

Guía del usuario

# Amazon Athena



# Amazon Athena: Guía del usuario

Copyright © 2024 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

¿Qué es Amazon Athena? .....	1
¿Cuándo debo utilizar Athena? .....	1
Amazon Athena .....	2
Amazon EMR .....	2
Amazon Redshift .....	3
Integraciones de los Servicio de AWS con Athena .....	4
Configuración .....	9
Registro en una Cuenta de AWS .....	10
Creación de un usuario con acceso administrativo .....	10
Conceder acceso programático .....	11
Asociación de políticas administradas para Athena .....	13
Acceso a Athena .....	14
Uso de Athena SQL .....	16
Comprensión de las tablas, las bases de datos y los catálogos de datos .....	17
Introducción .....	20
Requisitos previos .....	20
Paso 1: Crear una base de datos .....	20
Paso 2: crear una tabla .....	24
Paso 3: Consultar los datos .....	29
Guardar las consultas .....	32
Atajos de teclado y sugerencias de escritura anticipada .....	32
Conexión con otros orígenes de datos .....	33
Conexión con orígenes de datos .....	33
Integración con AWS Glue .....	34
Uso de un almacén de metadatos de Hive .....	54
Uso de consulta federada de Amazon Athena .....	90
Políticas de IAM para acceder a catálogos de datos .....	372
Administración de orígenes de datos .....	379
Uso de DataZone .....	381
Conexión a Amazon Athena con controladores ODBC y JDBC .....	383
Conexión a Athena con JDBC .....	384
Conexión a Athena con ODBC .....	435
Creación de bases de datos y tablas .....	583
Creación de bases de datos .....	584

Creación de tablas .....	587
Nombres de tablas, bases de datos y columnas .....	591
Palabras clave reservadas .....	594
Ubicación de las tablas en Simple Storage Service (Amazon S3) .....	596
Formatos de almacenamiento en columnas .....	599
Conversión a formatos de columnas .....	601
Particiones de datos .....	601
Proyección de particiones .....	609
Creación de una tabla a partir de los resultados de una consulta (CTAS) .....	634
Consideraciones y limitaciones de las consultas CTAS .....	635
Ejecución de consultas CTAS en la consola .....	638
Creación de particiones y asignación de buckets .....	640
Ejemplos de CTAS .....	645
Uso de CTAS e INSERT INTO en ETL .....	651
Cómo evitar el límite de 100 particiones .....	660
Referencia del SerDe .....	665
Uso de un SerDe .....	665
Formatos de datos y SerDes compatibles .....	666
Ejecución de consultas .....	718
Visualización de planes de consulta .....	720
Resultados de la consulta y consultas recientes .....	725
Reutilización de los resultados de las consultas .....	743
Visualización de estadísticas de consultas .....	749
Uso de vistas .....	755
Uso de consultas guardadas .....	771
Uso de consultas parametrizadas .....	774
Optimizador basado en costes .....	783
Consulta de S3 Express One Zone .....	789
Consulta de S3 Glacier .....	791
Gestión de las actualizaciones de los esquemas .....	793
Matrices de consulta .....	807
Consulta de datos geoespaciales .....	834
Consulta de JSON .....	860
Uso de ML con Athena .....	871
Consulta con UDF .....	874
Consultas entre regiones .....	886

Consulta de AWS Glue Data Catalog .....	887
Consulta de registros de Servicio de AWS .....	895
Consulta de registros de servidor web .....	977
Uso de las transacciones ACID .....	988
Consulta de las tablas de Delta Lake .....	989
Consulta de los conjuntos de datos de Hudi .....	994
Utilización de tablas de Iceberg .....	1005
Seguridad .....	1030
Protección de datos .....	1031
Administración de identidades y accesos .....	1046
Registro y monitorización .....	1117
Validación de cumplimiento .....	1123
Resiliencia .....	1124
Seguridad de la infraestructura .....	1125
Configuración y análisis de vulnerabilidades .....	1128
Uso de Athena con Lake Formation .....	1129
Administración de la carga de trabajo .....	1192
Uso de grupos de trabajo para controlar el acceso a las consultas y los costos .....	1192
Administración de la capacidad de procesamiento de consultas .....	1258
Ajuste del rendimiento .....	1276
Compatibilidad con la compresión .....	1300
Etiquetado de recursos .....	1310
Service Quotas .....	1326
Control de versiones del motor Athena .....	1329
Cambio de las versiones del motor Athena .....	1330
Referencia de la versión del motor Athena .....	1335
Referencia de SQL para Athena .....	1369
Tipos de datos en Athena .....	1370
Consultas, funciones y operadores de DML .....	1379
Instrucciones DDL .....	1441
Consideraciones y limitaciones .....	1499
Resolución de problemas .....	1500
CREATE TABLE AS SELECT (CTAS) .....	1501
Problemas de archivos de datos .....	1502
Tablas de Linux Foundation Delta Lake .....	1504
Consultas federadas .....	1504

Errores relacionados con JSON .....	1506
MSCK REPAIR TABLE .....	1507
Problemas de salida .....	1507
Problemas de Parquet .....	1508
Problemas de particiones .....	1509
Permisos .....	1512
Problemas con la sintaxis de las consultas .....	1513
Problemas con el tiempo de espera .....	1516
Problemas de limitación .....	1516
Vistas .....	1517
Grupos de trabajo .....	1517
Recursos adicionales de .....	1517
Catálogo de errores de Athena .....	1518
Ejemplos de código .....	1525
Constantes .....	1526
Crear un cliente para obtener acceso a Athena .....	1527
Iniciar la ejecución de una consulta .....	1527
Detener la ejecución de una consulta .....	1531
Generar una lista de ejecuciones de consultas .....	1533
Crear una consulta con nombre .....	1534
Eliminar una consulta con nombre .....	1536
Generar una lista de consultas con nombre .....	1538
Uso de Apache Spark .....	1540
Consideraciones y limitaciones .....	1540
Introducción .....	1542
Creación de un grupo de trabajo habilitado para Spark en Athena .....	1542
Apertura del explorador de cuadernos y cambio de grupo de trabajo .....	1547
Ejecución del cuaderno de ejemplo .....	1548
Edición de detalles de la sesión .....	1549
Visualización de detalles de sesión y cálculo .....	1551
Terminación de una sesión .....	1552
Creación de un cuaderno propio .....	1552
Apertura de un cuaderno creado anteriormente .....	1554
Uso de los cuadernos .....	1554
Sesiones y cálculos .....	1555
Uso del editor de cuadernos de Athena .....	1555

Comandos mágicos .....	1559
Administración de los archivos de cuaderno .....	1569
Uso de formatos de tabla que no son de Hive .....	1571
Compatibilidad con las bibliotecas Python .....	1576
Definiciones .....	1577
Administración del ciclo de vida .....	1577
Bibliotecas Python .....	1578
Importación de archivos y bibliotecas .....	1580
Adición de archivos JAR y configuración personalizada .....	1593
Uso de la consola de Athena .....	1593
Uso de la AWS CLI o la API de Athena .....	1594
Solución de problemas .....	1594
Formatos de datos y almacenamiento compatibles .....	1596
Supervisión de los cálculos de Apache Spark .....	1597
Lista de métricas y dimensiones de CloudWatch para cálculos de Apache Spark en Athena .....	1598
Habilitación de buckets de pagos por solicitante .....	1598
1. Habilite los pagos por solicitante en un bucket de Amazon S3 y agregue una política de bucket. ....	1599
2. Cree una política de IAM y adjúntela al rol de IAM. ....	1600
3. Agregue una propiedad de sesión de Athena para Spark. ....	1601
Habilitación del cifrado de Spark .....	1601
Consola de Athena .....	1602
AWS CLI .....	1603
API de Athena .....	1604
Acceso al catálogo entre cuentas .....	1604
1. Proporcionar acceso a los roles de consumidor en AWS Glue .....	1604
2. Configurar la cuenta de consumidor para el acceso .....	1605
3. Configurar una sesión y crear una consulta .....	1606
Recursos adicionales de .....	1607
Service Quotas .....	1608
API de cuadernos de Athena .....	1609
Problemas conocidos .....	1610
Excepción de argumento ilegal al crear una tabla .....	1610
Base de datos creada en una ubicación de grupo de trabajo .....	1611

Problemas con las tablas administradas por Hive en la base de datos de AWS Glue predeterminada .....	1611
Incompatibilidad de formatos de archivos CSV y JSON entre Athena para Spark y Athena SQL .....	1612
Solución de problemas .....	1613
Grupos de trabajo habilitados para Spark .....	1613
Uso de EXPLAIN de Spark .....	1616
Registro de eventos de aplicaciones .....	1619
Uso de CloudTrail para llamadas a la API de cuadernos .....	1623
Límite de tamaño para bloques de código .....	1630
Sesiones .....	1632
Tablas .....	1633
Cómo obtener asistencia .....	1635
Notas de la versión .....	1636
2024 .....	1636
26 de abril de 2024 .....	1636
24 de abril de 2024 .....	1636
16 de abril de 2024 .....	1637
10 de abril de 2024 .....	1637
8 de abril de 2024 .....	1638
15 de marzo de 2024 .....	1638
15 de febrero de 2024 .....	1638
31 de enero de 2024 .....	1639
2023 .....	1639
14 de diciembre de 2023 .....	1639
9 de diciembre de 2023 .....	1639
7 de diciembre de 2023 .....	1640
5 de diciembre de 2023 .....	1640
28 de noviembre de 2023 .....	1641
27 de noviembre de 2023 .....	1641
17 de noviembre de 2023 .....	1642
16 de noviembre de 2023 .....	1643
31 de octubre de 2023 .....	1644
25 de octubre de 2023 .....	1644
17 de octubre de 2023 .....	1644
26 de septiembre de 2023 .....	1644



---

23 de agosto de 2023 .....	1645
10 de agosto de 2023 .....	1645
31 de julio de 2023 .....	1645
27 de julio de 2023 .....	1646
24 de julio de 2023 .....	1646
20 de julio de 2023 .....	1646
13 de julio de 2023 .....	1647
3 de julio de 2023 .....	1648
30 de junio de 2023 .....	1648
29 de junio de 2023 .....	1648
28 de junio de 2023 .....	1649
12 de junio de 2023 .....	1649
8 de junio de 2023 .....	1650
2 de junio de 2023 .....	1650
25 de mayo de 2023 .....	1651
18 de mayo de 2023 .....	1652
15 de mayo de 2023 .....	1652
10 de mayo de 2023 .....	1653
8 de mayo de 2023 .....	1653
28 de abril de 2023 .....	1655
17 de abril de 2023 .....	1655
14 de abril de 2023 .....	1656
4 de abril de 2023 .....	1656
30 de marzo de 2023 .....	1657
28 de marzo de 2023 .....	1657
27 de marzo de 2023 .....	1658
17 de marzo de 2023 .....	1658
8 de marzo de 2023 .....	1659
15 de febrero de 2023 .....	1659
31 de enero de 2023 .....	1659
20 de enero de 2023 .....	1660
3 de enero de 2023 .....	1660
2022 .....	1661
14 de diciembre de 2022 .....	1661
2 de diciembre de 2022 .....	1661
30 de noviembre de 2022 .....	1662

18 de noviembre de 2022 .....	1662
17 de noviembre de 2022 .....	1662
14 de noviembre de 2022 .....	1663
11 de noviembre de 2022 .....	1664
8 de noviembre de 2022 .....	1665
13 de octubre de 2022 .....	1665
10 de octubre de 2022 .....	1666
23 de septiembre de 2022 .....	1666
13 de septiembre de 2022 .....	1667
31 de agosto de 2022 .....	1667
23 de agosto de 2022 .....	1667
3 de agosto de 2022 .....	1668
1 de agosto de 2022 .....	1668
21 de julio de 2022 .....	1669
11 de julio de 2022 .....	1670
8 de julio de 2022 .....	1670
6 de junio de 2022 .....	1670
25 de mayo de 2022 .....	1671
6 de mayo de 2022 .....	1671
22 de abril de 2022 .....	1672
21 de abril de 2022 .....	1672
13 de abril de 2022 .....	1673
30 de marzo de 2022 .....	1674
18 de marzo de 2022 .....	1674
2 de marzo de 2022 .....	1675
23 de febrero de 2022 .....	1675
15 de febrero de 2022 .....	1676
14 de febrero de 2022 .....	1676
9 de febrero de 2022 .....	1677
8 de febrero de 2022 .....	1677
28 de enero de 2022 .....	1677
13 de enero de 2022 .....	1678
2021 .....	1678
26 de noviembre de 2021 .....	1678
24 de noviembre de 2021 .....	1679
22 de noviembre de 2021 .....	1679

---

18 de noviembre de 2021 .....	1680
17 de noviembre de 2021 .....	1681
16 de noviembre de 2021 .....	1681
12 de noviembre de 2021 .....	1682
2 de noviembre de 2021 .....	1682
29 de octubre de 2021 .....	1682
4 de octubre de 2021 .....	1684
16 de septiembre de 2021 .....	1684
15 de septiembre de 2021 .....	1685
31 de agosto de 2021 .....	1686
12 de agosto de 2021 .....	1686
6 de agosto de 2021 .....	1686
5 de agosto de 2021 .....	1687
30 de julio de 2021 .....	1687
21 de julio de 2021 .....	1688
16 de julio de 2021 .....	1688
8 de julio de 2021 .....	1689
1 de julio de 2021 .....	1689
23 de junio de 2021 .....	1689
12 de mayo de 2021 .....	1690
10 de mayo de 2021 .....	1690
5 de mayo de 2021 .....	1690
30 de abril de 2021 .....	1691
29 de abril de 2021 .....	1691
26 de abril de 2021 .....	1691
21 de abril de 2021 .....	1691
5 de abril de 2021 .....	1692
30 de marzo de 2021 .....	1692
25 de marzo de 2021 .....	1693
5 de marzo de 2021 .....	1693
25 de febrero de 2021 .....	1693
2020 .....	1693
16 de diciembre de 2020 .....	1693
24 de noviembre de 2020 .....	1694
11 de noviembre de 2020 .....	1694
22 de octubre de 2020 .....	1697

---

29 de julio de 2020 .....	1697
9 de julio de 2020 .....	1697
1 de junio de 2020 .....	1698
21 de mayo de 2020 .....	1698
1 de abril de 2020 .....	1699
11 de marzo de 2020 .....	1699
6 de marzo de 2020 .....	1699
2019 .....	1699
26 de noviembre de 2019 .....	1699
12 de noviembre de 2019 .....	1704
8 de noviembre de 2019 .....	1704
8 de octubre de 2019 .....	1704
19 de septiembre de 2019 .....	1705
12 de septiembre de 2019 .....	1705
16 de agosto de 2019 .....	1705
9 de agosto de 2019 .....	1706
26 de junio de 2019 .....	1706
24 de mayo de 2019 .....	1706
05 de marzo de 2019 .....	1706
22 de febrero de 2019 .....	1707
18 de febrero de 2019 .....	1708
2018 .....	1710
20 de noviembre de 2018 .....	1710
15 de octubre de 2018 .....	1711
10 de octubre de 2018 .....	1712
6 de septiembre de 2018 .....	1712
23 de agosto de 2018 .....	1713
16 de agosto de 2018 .....	1714
7 de agosto de 2018 .....	1715
5 de junio de 2018 .....	1715
17 de mayo de 2018 .....	1716
19 de abril de 2018 .....	1717
6 de abril de 2018 .....	1717
15 de marzo de 2018 .....	1717
2 de febrero de 2018 .....	1718
19 de enero de 2018 .....	1718

---

2017 .....	1719
13 de noviembre de 2017 .....	1719
1 de noviembre de 2017 .....	1719
19 de octubre de 2017 .....	1719
3 de octubre de 2017 .....	1720
25 de septiembre de 2017 .....	1720
14 de agosto de 2017 .....	1720
4 de agosto de 2017 .....	1720
22 de junio de 2017 .....	1720
8 de junio de 2017 .....	1721
19 de mayo de 2017 .....	1721
4 de abril de 2017 .....	1722
24 de marzo de 2017 .....	1724
20 de febrero de 2017 .....	1725
Historial de documentos .....	1728
Glosario de AWS .....	1757

# ¿Qué es Amazon Athena?

Amazon Athena es un servicio de consultas interactivo que facilita el análisis de datos directamente en Amazon Simple Storage Service (Amazon S3) con [SQL](#) estándar. Con unas pocas acciones en la AWS Management Console, puede apuntar Athena a los datos almacenados en Amazon S3 y comenzar a utilizar SQL estándar para ejecutar consultas ad hoc y obtener resultados en cuestión de segundos.

Para obtener más información, consulte [Introducción](#).

Amazon Athena también facilita la ejecución interactiva de análisis de datos mediante Apache Spark sin tener que planificar, configurar ni administrar los recursos. Cuando ejecute aplicaciones de Apache Spark en Athena, envía el código de Spark para su procesamiento y recibe directamente los resultados. Utilice la experiencia simplificada de cuadernos de la consola de Amazon Athena para desarrollar aplicaciones de Apache Spark mediante Python o [API de cuadernos de Athena](#).

Para obtener más información, consulte [Introducción a Apache Spark en Amazon Athena](#).

Athena SQL y Apache Spark en Amazon Athena no requieren un servidor, por lo que no hay que configurar ni administrar ninguna infraestructura y solo pagará por las consultas que ejecute. Athena se escala automáticamente y ejecuta las consultas en paralelo, de modo que los resultados se obtienen rápidamente, incluso con conjuntos de datos de gran tamaño y consultas complejas.

## Temas

- [¿Cuándo debo utilizar Athena?](#)
- [Integraciones de los Servicio de AWS con Athena](#)
- [Configuración](#)
- [Acceso a Athena](#)

# ¿Cuándo debo utilizar Athena?

Los servicios de consultas como Amazon Athena, los almacenamientos de datos como Amazon Redshift y los sofisticados marcos de procesamiento de datos como Amazon EMR abordan diferentes necesidades y casos de uso. Las siguientes orientaciones pueden ayudar a elegir uno o varios servicios en función de sus requisitos.

## Amazon Athena

Athena lo ayuda a analizar datos no estructurados, semiestructurados y estructurados almacenados en Amazon S3. Algunos ejemplos son datos en CSV, JSON o con formatos de columnas, como Apache Parquet y Apache ORC. Puede utilizar Athena para ejecutar consultas ad hoc con ANSI SQL y sin necesidad de agregar los datos o cargarlos en Athena.

Athena se integra con Amazon QuickSight para facilitar la visualización de datos. Puede utilizar Athena para generar informes o para explorar datos con herramientas de inteligencia empresarial o clientes SQL conectados mediante un controlador ODBC o JDBC. Para obtener más información consulte [Qué es Amazon QuickSight](#) en la Guía del usuario de Amazon QuickSight y en [Conexión a Amazon Athena con controladores ODBC y JDBC](#).

Athena se integra con AWS Glue Data Catalog, que ofrece un almacén de metadatos persistente para los datos en Amazon S3. Esto le permite crear tablas y consultar datos en Athena partiendo de un almacén de metadatos central disponible en toda su cuenta de Amazon Web Services e integrado con ETL y las características de descubrimiento de datos de AWS Glue. Para obtener más información, consulte [Integración con AWS Glue](#) y [¿Qué es AWS Glue?](#) en la Guía para desarrolladores de AWS Glue.

Amazon Athena facilita la ejecución de consultas interactivas con datos directamente en Simple Storage Service (Amazon S3) sin tener que dar formato a los datos ni administrar la infraestructura. Por ejemplo, Athena es útil si desea ejecutar una consulta rápida en los registros web para solucionar un problema de rendimiento en su sitio. Con Athena, puede comenzar rápidamente: basta con definir una tabla para los datos y comenzar a realizar consultas mediante SQL estándar.

Conviene utilizar Amazon Athena si desea ejecutar consultas SQL ad hoc interactivas con datos de Simple Storage Service (Amazon S3), sin tener que administrar ninguna infraestructura o clústeres. Amazon Athena proporciona la forma más sencilla de ejecutar consultas ad hoc para datos en Simple Storage Service (Amazon S3) sin necesidad de configurar ni administrar ningún servidor.

Para obtener una lista de Servicios de AWS que utiliza o con los que se integra Athena, consulte [the section called “Integraciones de los Servicio de AWS con Athena”](#).

## Amazon EMR

Amazon EMR simplifica y hace que resulte rentable ejecutar marcos de procesamiento altamente distribuidos como Hadoop, Spark y Presto en comparación con las implementaciones en las instalaciones. Amazon EMR es flexible: puede ejecutar aplicaciones y código personalizados y definir

parámetros específicos de computación, memoria, almacenamiento y aplicaciones para optimizar sus requisitos de análisis.

Además de ejecutar consultas SQL, Amazon EMR puede ejecutar una amplia variedad de tareas de procesamiento de datos de escalado horizontal para aplicaciones como machine learning, análisis de gráficos, transformación de datos, datos de streaming y prácticamente cualquier cosa que pueda codificar. Conviene utilizar Amazon EMR si utiliza código personalizado para procesar y analizar conjuntos de datos extremadamente grandes con los marcos de procesamiento de macrodatos más recientes, tales como Spark, Hadoop, Presto o Hbase. Amazon EMR proporciona un control total sobre la configuración de los clústeres y el software instalado en ellos.

Puede utilizar Amazon Athena para consultar datos que procese mediante Amazon EMR. Amazon Athena admite muchos de los mismos formatos de datos que Amazon EMR. El catálogo de datos de Athena es compatible con metaalmacenes de Hive. Si utiliza EMR y ya tiene un metaalmacén de Hive, puede ejecutar las instrucciones DDL en Amazon Athena y consultar los datos inmediatamente sin que ello afecte a los trabajos de Amazon EMR.

## Amazon Redshift

Un almacenamiento de datos como Amazon Redshift es la mejor opción cuando se necesita reunir datos de muchos orígenes diferentes, como sistemas de inventario, sistemas financieros y sistemas de ventas minoristas, en un formato común y almacenarlos durante largos periodos de tiempo. Si desea crear informes empresariales sofisticados a partir de datos históricos, entonces un almacenamiento de datos como Amazon Redshift es la mejor opción. El motor de consultas de Amazon Redshift se ha optimizado para que funcione especialmente bien en la ejecución de consultas complejas que unen un gran número de tablas de bases de datos muy grandes. Cuando necesite ejecutar consultas sobre datos altamente estructurados con muchas combinaciones en muchas tablas muy grandes, elija Amazon Redshift.

Para obtener más información acerca de cuándo utilizar Athena, consulte los siguientes recursos:

- [Guía de decisiones para los servicios de análisis en AWS](#) en el Centro de recursos introductorios
- [Cuándo utilizar Athena en comparación con otros servicios de macrodatos](#) en las Preguntas frecuentes de Amazon Athena
- [Información general de Amazon Athena](#)
- [Características de Amazon Athena](#)
- [Preguntas frecuentes sobre Amazon Athena](#)



- [Publicaciones de blog de Amazon Athena](#)

## Integraciones de los Servicio de AWS con Athena

Puede utilizar Athena para consultar los datos de los Servicios de AWS que se enumeran en esta sección. Para ver las regiones compatibles con cada servicio, consulte [Regiones y puntos de conexión](#) en la Referencia general de Amazon Web Services.

### Servicios de AWS integrados con Athena

- [AWS CloudFormation](#)
- [Amazon CloudFront](#)
- [AWS CloudTrail](#)
- [Amazon DataZone](#)
- [Elastic Load Balancing](#)
- [Amazon EMR Studio](#)
- [AWS Glue Data Catalog](#)
- [AWS Identity and Access Management \(IAM\)](#)
- [Amazon QuickSight](#)
- [Inventario de Amazon S3](#)
- [AWS Step Functions](#)
- [Inventario de AWS Systems Manager](#)
- [Amazon Virtual Private Cloud](#)

Para obtener información sobre cada integración, consulte las siguientes secciones.

### AWS CloudFormation

#### Reserva de capacidad

Tema de referencia: [AWS::Athena::CapacityReservation](#) en la Guía del usuario de AWS CloudFormation

Especifica una reserva de capacidad con el nombre y el número de unidades de procesamiento de datos solicitadas proporcionados. Para obtener más información, consulte

[Administración de la capacidad de procesamiento de consultas](#) en la Guía del usuario de Amazon Athena y [CreateCapacityReservation](#) en la Referencia de la API de Amazon Athena.

## Catálogo de datos

Tema de referencia: [AWS::Athena::DataCatalog](#) en la Guía del usuario de AWS CloudFormation

Especifique un catálogo de datos de Athena que incluya nombre, descripción, tipo, parámetros y etiquetas. Para obtener más información, consulte [Comprensión de las tablas, las bases de datos y los catálogos de datos](#) en la Guía del usuario de Amazon Athena y [CreateDataCatalog](#) en la Referencia de la API de Amazon Athena.

## Consulta con nombre

Tema de referencia: [AWS::Athena::NamedQuery](#) en la Guía del usuario de AWS CloudFormation

Especifique consultas con nombre con AWS CloudFormation y ejecútelas en Athena. Las consultas con nombre le permiten asignar un nombre de consulta a una consulta y, a continuación, ejecutarla como una consulta guardada desde la consola de Athena. Para obtener más información, consulte [Uso de consultas guardadas](#) en la Guía del usuario de Amazon Athena y [CreateNamedQuery](#) en la Referencia de la API de Amazon Athena.

## Instrucciones preparadas

Tema de referencia: [AWS::Athena::PreparedStatement](#) en la Guía del usuario de AWS CloudFormation

Especifica una instrucción preparada para usarse con consultas SQL en Athena. Una instrucción preparada contiene marcadores de posición de parámetros cuyos valores se proporcionan en el momento de la ejecución. Para obtener más información, consulte [Uso de consultas parametrizadas](#) en la Guía del usuario de Amazon Athena y [CreatePreparedStatement](#) en la Referencia de la API de Amazon Athena.

## Grupo de trabajo

Tema de referencia: [AWS::Athena::WorkGroup](#) en la Guía del usuario de AWS CloudFormation

Especifique grupos de trabajo de Athena mediante AWS CloudFormation. Utilice los grupos de trabajo de Athena para aislar las consultas realizadas por usted o su grupo de otras consultas de la misma cuenta. Para obtener más información, consulte [Uso de grupos de trabajo para](#)

[controlar el acceso a las consultas y los costos](#) en la Guía del usuario de Amazon Athena y [CreateWorkGroup](#) en la Referencia de la API de Amazon Athena.

## Amazon CloudFront

Tema de referencia: [Consultas de registros de Amazon CloudFront](#)

Utilice Athena para consultar registros de Amazon CloudFront. Para obtener más información sobre el uso de CloudFront, consulte la [Guía para desarrolladores de Amazon CloudFront](#).

## AWS CloudTrail

Tema de referencia: [Consulta de registros de AWS CloudTrail](#)

El uso de Athena con los registros de CloudTrail supone un modo eficaz de mejorar el análisis de la actividad de un servicio de AWS. Por ejemplo, puede utilizar consultas para identificar tendencias y aislar la actividad por atributo, como el usuario o la dirección IP de fuente. Puede crear tablas para consultar los registros directamente desde la consola de CloudTrail y utilizar esas tablas para ejecutar consultas en Athena. Para obtener más información, consulte [Uso de la consola de CloudTrail para crear una tabla de Athena para registros de CloudTrail](#).

## Amazon DataZone

Tema de referencia: [Uso de Amazon DataZone en Athena](#)

Utilice [Amazon DataZone](#) para compartir, buscar y descubrir datos a escala más allá de los límites de la organización. DataZone simplifica su experiencia en servicios de análisis de AWS como Athena, AWS Glue y AWS Lake Formation. Si tiene grandes cantidades de datos en diferentes orígenes de datos, puede utilizar Amazon DataZone para crear agrupaciones de personas, datos y herramientas basadas en casos de uso empresarial.

En Athena, puede utilizar el editor de consultas para acceder a los entornos de DataZone y consultarlos. Para obtener más información, consulte [Uso de Amazon DataZone en Athena](#).

## Elastic Load Balancing

Tema de referencia: [Consulta de los registros de Application Load Balancer](#)

La consulta de los registros del Equilibrador de carga de aplicación le permite ver el origen del tráfico, la latencia y los bytes transferidos a instancias de Elastic Load Balancing y desde dichas instancias y las aplicaciones de backend. Para obtener más información, consulte [Consulta de los registros de Application Load Balancer](#).

Tema de referencia: [Consulta de los registros del Equilibrador de carga clásico](#)

Consulte los registros del Equilibrador de carga clásico para analizar y conocer los patrones del tráfico desde y hacia instancias de Elastic Load Balancing y aplicaciones de backend. Puede ver el origen del tráfico, su latencia y los bytes transferidos. Para obtener más información, consulte [Creación de la tabla para los registros de ELB](#).

## Amazon EMR Studio

Tema de referencia: [Uso del editor SQL de Amazon Athena en EMR Studio](#)

Puede usar Athena en un EMR Studio para desarrollar y ejecutar consultas interactivas. Esto le permite utilizar EMR Studio para análisis de SQL en Athena desde la misma interfaz de Amazon EMR que utiliza para sus cargas de trabajo de Spark, Scala y otras. Con la integración de Athena en EMR Studio, puede llevar a cabo las siguientes tareas:

- Realizar consultas SQL de Athena
- Visualización de los resultados de la consulta
- Visualizar el historial de consultas
- Visualizar las consultas guardadas
- Realizar consultas parametrizadas
- Ver bases de datos, tablas y vistas de un catálogo de datos

Las siguientes características de Athena no están disponibles en Amazon EMR Studio:

- Características administrativas, por ejemplo, crear o actualizar grupos de trabajo, orígenes de datos o reservas de capacidad de Athena
- Athena para Spark o portátiles Spark
- Integración de DataZone
- Step Functions

La integración de EMR Studio con Athena está disponible en todas las Regiones de AWS donde estén disponibles EMR Studio y Athena. Para obtener más información sobre el uso de Athena en EMR Studio, consulte [Uso del editor SQL de Amazon Athena en EMR Studio](#) en la Guía de administración de Amazon EMR.

## AWS Glue Data Catalog

Tema de referencia: [Integración con AWS Glue](#)

Athena se integra con AWS Glue Data Catalog, que ofrece un almacén de metadatos persistente para los datos en Amazon S3. Esto le permite crear tablas y consultar datos en Athena partiendo

de un almacén de metadatos central disponible en toda su cuenta de Amazon Web Services e integrado con ETL y las características de descubrimiento de datos de AWS Glue. Para obtener más información, consulte [Integración con AWS Glue](#) y [What Is AWS Glue](#) (¿Qué es ?) en la Guía para desarrolladores de AWS Glue.

## AWS Identity and Access Management (IAM)

Tema de referencia: [Acciones de Amazon Athena](#)

Puede utilizar acciones de la API de Athena en políticas de permisos de IAM. Para obtener más información, consulte [Acciones de Amazon Athena](#) y [Identity and Access Management en Athena](#).

## Amazon QuickSight

Tema de referencia: [Conexión a Amazon Athena con controladores ODBC y JDBC](#)

Athena se integra con Amazon QuickSight para facilitar la visualización de datos. Puede utilizar Athena para generar informes o para explorar datos con herramientas de inteligencia empresarial o clientes SQL conectados mediante un controlador ODBC o JDBC. Para obtener más información acerca de Amazon QuickSight, consulte [Qué es Amazon QuickSight](#) en la Guía del usuario de Amazon QuickSight. Para obtener información sobre el uso de controladores JDBC y ODBC con Athena, consulte [Conexión a Amazon Athena con controladores ODBC y JDBC](#).

## Inventario de Amazon S3

Tema de referencia: [Consultas de inventario con Athena](#) en la Guía del usuario de Amazon Simple Storage Service

Puede utilizar Amazon Athena para consultar el inventario de Amazon S3 mediante SQL estándar. Puede utilizar el inventario de Amazon S3 para auditar e informar sobre el estado de replicación y cifrado de los objetos para sus necesidades empresariales, de conformidad y legales. Para obtener más información, consulte [Inventario de Amazon S3](#) en la Guía del usuario de Amazon Simple Storage Service.

## AWS Step Functions

Tema de referencia: [Llamar a Athena con Step Functions](#) en la Guía para desarrolladores de AWS Step Functions

Llame a Athena con AWS Step Functions. AWS Step Functions puede controlar determinados Servicios de AWS directamente con [Amazon States Language](#). Puede utilizar Step Functions con Athena para iniciar y detener la ejecución de consultas, obtener resultados de consultas,

ejecutar consultas de datos ad hoc o programadas y recuperar resultados de lagos de datos en Amazon S3. El rol de Step Functions debe tener permisos para usar Athena. Para obtener más información, consulte la [Guía para desarrolladores de AWS Step Functions](#).

Video: Orquestación de consultas de Amazon Athena mediante AWS Step Functions

En el siguiente video se muestra cómo utilizar Amazon Athena y AWS Step Functions para ejecutar una consulta Athena programada regularmente y generar un informe correspondiente.

### [Orquestación de las consultas de Amazon Athena mediante AWS Step Functions](#)

Para obtener un ejemplo que utiliza Step Functions y Amazon EventBridge para orquestar AWS Glue DataBrew, Athena y Amazon QuickSight, consulte [Orquestación de un trabajo AWS Glue DataBrew y una consulta de Amazon Athena con AWS Step Functions](#) en el blog de big data de AWS.

## Inventario de AWS Systems Manager

Tema de referencia: [Consulta de datos de inventario de varias regiones y cuentas](#) en la Guía del usuario de AWS Systems Manager

El inventario de AWS Systems Manager se integra con Amazon Athena para ayudarle a consultar los datos de inventario de varias cuentas y Regiones de AWS. Para más información, consulte la [Guía del usuario de AWS Systems Manager](#).

## Amazon Virtual Private Cloud

Tema de referencia: [Consulta de los registros de flujo de Amazon VPC](#)

Los registros de flujo de Amazon Virtual Private Cloud capturan información acerca del tráfico IP entrante y saliente por las interfaces de red de una VPC. Consulte los registros en Athena para investigar los patrones de tráfico de red e identificar las amenazas y los riesgos en la red de Amazon VPC. Para obtener más información sobre Amazon VPC, consulte la [Guía del usuario de Amazon VPC](#).

# Configuración

Si ya se registró en Amazon Web Services, puede comenzar a utilizar Amazon Athena de inmediato. Si aún no está registrado en AWS o necesita ayuda para comenzar, asegúrese de realizar las tareas siguientes.

## Registro en una Cuenta de AWS

Si no dispone de una Cuenta de AWS, siga estos pasos para crear una.

### Procedimiento para registrarse en Cuenta de AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga las instrucciones que se le indiquen.

Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

Al registrarse en una Cuenta de AWS, se crea un Usuario raíz de la cuenta de AWS. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar [tareas que requieren acceso de usuario raíz](#).

AWS le enviará un email de confirmación cuando complete el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en <https://aws.amazon.com/> y seleccionando Mi cuenta.

## Creación de un usuario con acceso administrativo

Después de registrarse para obtener una Cuenta de AWS, proteja su Usuario raíz de la cuenta de AWS, habilite AWS IAM Identity Center y cree un usuario administrativo para no utilizar el usuario raíz en las tareas cotidianas.

### Protección de Usuario raíz de la cuenta de AWS

1. Inicie sesión en [AWS Management Console](#) como propietario de la cuenta; para ello, elija Usuario raíz e introduzca el correo electrónico de su Cuenta de AWS. En la siguiente página, escriba su contraseña.

Para obtener ayuda para iniciar sesión con el usuario raíz, consulte [Signing in as the root user](#) en la Guía del usuario de AWS Sign-In.

2. Active la autenticación multifactor (MFA) para el usuario raíz.

Para obtener instrucciones, consulte [Habilitación de un dispositivo MFA virtual para su usuario raíz de la Cuenta de AWS \(consola\)](#) en la Guía del usuario de IAM.

## Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

Consulte las instrucciones en [Activar AWS IAM Identity Center](#) en la Guía del usuario de AWS IAM Identity Center.

2. En el IAM Identity Center, conceda acceso administrativo a un usuario.

Para ver un tutorial sobre cómo utilizar Directorio de IAM Identity Center como origen de identidad, consulte [Configuración del acceso de los usuarios con el Directorio de IAM Identity Center predeterminado](#) en la Guía del usuario de AWS IAM Identity Center.

## Inicio de sesión como usuario con acceso administrativo

- Para iniciar sesión con el usuario de IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

Para obtener ayuda para iniciar sesión con un usuario del IAM Identity Center, consulte [Inicio de sesión en el portal de acceso de AWS](#) en la Guía del usuario de AWS Sign-In.

## Concesión de acceso a usuarios adicionales

1. En el IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

Para conocer las instrucciones, consulte [Creación de un conjunto de permisos](#) en la Guía del usuario de AWS IAM Identity Center.

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

Para conocer las instrucciones, consulte [Cómo agregar grupos](#) en la Guía del usuario de AWS IAM Identity Center.

## Conceder acceso programático

Los usuarios necesitan acceso programático si desean interactuar con AWS fuera de la AWS Management Console. La forma de conceder el acceso programático depende del tipo de usuario que acceda a AWS.



Para conceder acceso programático a los usuarios, seleccione una de las siguientes opciones.

¿Qué usuario necesita acceso programático?	Para	Mediante
Identidad del personal  (Usuarios administrados en el IAM Identity Center)	Utilice credenciales temporales para firmar las solicitudes programáticas a la AWS CLI, los AWS SDK y las API de AWS.	Siga las instrucciones de la interfaz que desea utilizar: <ul style="list-style-type: none"> <li>• Para utilizar la AWS CLI, consulte <a href="#">Configuring the AWS CLI to use AWS IAM Identity Center</a> en la Guía del usuario de AWS Command Line Interface.</li> <li>• Para usar AWS SDK, las herramientas y las API de AWS, consulte <a href="#">IAM Identity Center authentication</a> en la Guía de referencia del SDK y las herramientas de AWS.</li> </ul>
IAM	Utilice credenciales temporales para firmar las solicitudes programáticas a la AWS CLI, los AWS SDK y las API de AWS.	Siguiendo las instrucciones de <a href="#">Uso de credenciales temporales con recursos de AWS</a> de la Guía del usuario de IAM.
IAM	(No recomendado) Utilizar credenciales a largo plazo para firmar las solicitudes programáticas a la AWS CLI, los AWS SDK o las API de AWS.	Siga las instrucciones de la interfaz que desea utilizar: <ul style="list-style-type: none"> <li>• Para la AWS CLI, consulte <a href="#">Autenticación mediante credenciales de usuario de IAM</a> en la Guía del usuario de AWS Command Line Interface.</li> <li>• Para ver los AWS SDK y las herramientas, consulte</li> </ul>

¿Qué usuario necesita acceso programático?	Para	Mediante
		<p><a href="#">Autenticar mediante credenciales a largo plazo</a> en la Guía de referencia de AWS SDK y herramientas.</p> <ul style="list-style-type: none"> <li>• Para las API de AWS, consulte <a href="#">Administración de claves de acceso para usuarios de IAM</a> en la Guía del usuario de IAM.</li> </ul>

## Asociación de políticas administradas para Athena

Las políticas administradas por Athena otorgan permisos para usar las características de Athena. Puede adjuntar estas políticas administradas a uno o más roles de IAM que los usuarios pueden asumir para utilizar Athena.

Un [rol de IAM](#) es una identidad de IAM que puede crear en su cuenta y que tiene permisos específicos. Un rol de IAM es similar a un usuario de IAM en que se trata de una identidad de AWS con políticas de permisos que determinan lo que la identidad puede hacer y lo que no en AWS. No obstante, en lugar de asociarse exclusivamente a una persona, la intención es que cualquier usuario pueda asumir un rol que necesite. Además, un rol no tiene asociadas credenciales a largo plazo estándar, como una contraseña o claves de acceso. En su lugar, cuando se asume un rol, este proporciona credenciales de seguridad temporales para la sesión de rol.

Para obtener más información acerca de la creación y el uso de roles, consulte [roles de IAM](#) y [Creación de roles de IAM](#) en la Guía del usuario de IAM.

Para crear un rol que otorgue acceso a Athena, debe adjuntar las políticas administradas por Athena al rol. Existen dos políticas administradas para Athena: `AmazonAthenaFullAccess` y `AWSQuicksightAthenaAccess`. Estas políticas conceden permisos a Athena para consultar Amazon S3 y escribir los resultados de las consultas en un bucket aparte en su nombre. Consulte [Políticas administradas de AWS para Amazon Athena](#) para ver el contenido de estas políticas de Athena.

Para conocer los pasos para adjuntar las políticas administradas por Athena a un rol, siga lo indicado en [Adición de permisos de identidad de IAM \(consola\)](#) en la Guía del usuario de IAM y agregue las políticas administradas AmazonAthenaFullAccess y AWSQuicksightAthenaAccess al rol que creó.

### Note

Es posible que necesite permisos adicionales para el acceso al conjunto de datos subyacente en Amazon S3. Si no es el propietario de la cuenta o su acceso a un bucket está restringido de algún otro modo, póngase en contacto con el propietario del bucket para que le conceda acceso mediante una política de bucket basada en recursos, o póngase en contacto con el administrador de cuentas para que le conceda acceso mediante una política basada roles. Para obtener más información, consulte [Acceso a Amazon S3](#). Si el conjunto de datos o los resultados de la consulta de Athena están cifrados, es posible que necesite permisos adicionales. Para obtener más información, consulte [Cifrado en reposo](#).

## Acceso a Athena

Puede acceder a Athena usando la AWS Management Console, una conexión JDBC u ODBC, la API de Athena, la CLI de Athena, el SDK de AWS o AWS Tools for Windows PowerShell.

- Para empezar a utilizar Athena SQL con la consola, consulte [Introducción](#).
- Para empezar a crear cuadernos compatibles con Jupyter y aplicaciones Apache Spark que utilicen Python, consulte [Uso de Apache Spark en Amazon Athena](#).
- Para obtener información sobre cómo utilizar los controladores JDBC u ODBC, consulte [Conexión a Amazon Athena con JDBC](#) y [Conexión a Amazon Athena con ODBC](#).
- Para utilizar la API de Athena, consulte la [Referencia de API de Amazon Athena](#).
- Para utilizar la CLI, [instale la AWS CLI](#) y, a continuación, escriba `aws athena help` en la línea de comandos para ver los comandos disponibles. Para obtener más información sobre los comandos disponibles, consulte la [referencia de línea de comandos de Amazon Athena](#).
- Para utilizar AWS SDK for Java 2.x, consulte la sección de Athena de la [Referencia de la API de AWS SDK for Java 2.x](#), los [Ejemplos de Athena Java V2](#) en GitHub.com y la [Guía para desarrolladores de AWS SDK for Java 2.x](#).

- Para utilizar AWS SDK for .NET, consulte el espacio de nombres de Amazon.Athena en la [Referencia de la API de AWS SDK for .NET](#), los [Ejemplos de Athena .NET](#) en GitHub.com y la [Guía para desarrolladores de AWS SDK for .NET](#).
- Para utilizar AWS Tools for Windows PowerShell, consulte la referencia de cmdlet de [AWS Tools for PowerShell - Amazon Athena](#), la página del portal de [AWS Tools for PowerShell](#) y la [Guía del usuario de AWS Tools for Windows PowerShell](#).
- Para obtener información sobre los puntos de conexión de servicio de Athena a los que puede conectarse mediante programación, consulte los [puntos de conexión y cuotas de Amazon Athena](#) en la [Referencia general de Amazon Web Services](#).

# Uso de Athena SQL

Puede utilizar Athena SQL para consultar sus datos de Amazon S3 in situ mediante el [AWS Glue Data Catalog](#), [un metaalmacén de Hive externo](#) o [consultas federadas](#) mediante una variedad de [conectores prediseñados](#) a otros orígenes de datos.

También puede:

- Conectarse a herramientas de inteligencia empresarial y otras aplicaciones mediante los [controladores JDBC y ODBC de Athena](#).
- Consultar los [registros de los servicios de AWS](#).
- Consultar las [tablas de Apache Iceberg](#), incluidas las consultas de viajes en el tiempo, y [los conjuntos de datos de Apache Hudi](#).
- Consultar los [datos geoespaciales](#).
- Realizar consultas mediante la [inferencia de machine learning](#) de Amazon SageMaker.
- Realizar consultas mediante sus propias [funciones definidas por el usuario](#).
- Acelerar el procesamiento de consultas de tablas particionadas y automatizar la administración de particiones mediante la [proyección de particiones](#).

## Temas

- [Comprensión de las tablas, las bases de datos y los catálogos de datos](#)
- [Introducción](#)
- [Conexión con orígenes de datos](#)
- [Conexión a Amazon Athena con controladores ODBC y JDBC](#)
- [Creación de bases de datos y tablas](#)
- [Creación de una tabla a partir de los resultados de una consulta \(CTAS\)](#)
- [Referencia del SerDe](#)
- [Ejecución de consultas SQL mediante Amazon Athena](#)
- [Uso de las transacciones ACID de Athena](#)
- [Seguridad de Amazon Athena](#)
- [Administración de la carga de trabajo](#)
- [Control de versiones del motor Athena](#)

- [Referencia de SQL para Athena](#)
- [Solución de problemas en Athena](#)
- [Ejemplos de código](#)

## Comprensión de las tablas, las bases de datos y los catálogos de datos

En Athena, los catálogos, las bases de datos y las tablas son contenedores en los que se incluyen las definiciones de los metadatos que definen un esquema para los datos de orígenes subyacentes.

Athena utiliza los siguientes términos para referirse a las jerarquías de los objetos de datos:

- Origen de datos: un grupo de bases de datos.
- Base de datos: un grupo de tablas.
- Tabla: datos organizados como un grupo de filas o columnas.

En ocasiones, también se hace referencia a estos objetos con nombres alternativos pero equivalentes, como los siguientes:

- Un origen de datos es lo que a veces se denomina catálogo.
- Una base de datos es lo que a veces se denomina esquema.

### Note

Esta terminología puede variar en los orígenes de datos federados que utilice con Athena. Para obtener más información, consulte [Calificadores de nombres de tablas federadas y Athena](#).

En la siguiente consulta de ejemplo en la consola de Athena, se utiliza el origen de datos `awsdatacatalog`, la base de datos `default` y la tabla `some_table`.

The screenshot shows the Amazon Athena console interface. On the left, the 'Data' panel is visible, showing the 'Data source' set to 'AwsDataCatalog' and the 'Database' set to 'default'. Below this, the 'Tables and views' section lists two tables: 'some\_table' and 'some\_table2'. The main editor area shows a query: `SELECT * FROM "awsdatacatalog"."default"."some_table" limit 10;`. The query has been executed successfully, as indicated by the 'Completed' status. The results are displayed in a table with 5 rows and 4 columns: #, id, data, and category.

#	id	data	category
1	1	a	A
2	3	d	d1
3	4	e	e1
4	4	f	f1
5	2	b	b1

Para cada conjunto de datos debe existir una tabla en Athena. Los metadatos de la tabla indican a Athena dónde se encuentran los datos en Amazon S3 y especifican su estructura; por ejemplo los nombres de columna, los tipos de datos y el nombre de la tabla. Las bases de datos son una agrupación lógica de tablas y contienen únicamente información de metadatos y esquema para un conjunto de datos.

Para cada conjunto de datos que desee consultar, Athena debe tener una tabla subyacente que usará para obtener y devolver los resultados de las consultas. Por lo tanto, para poder consultar los datos es necesario haber registrado previamente una tabla en Athena. El registro se produce al crear las tablas de forma automática o manual.

Puede crear una tabla de forma automática mediante un rastreador de AWS Glue. Para obtener más información sobre AWS Glue y los rastreadores, consulte [Integración con AWS Glue](#). Cuando AWS Glue crea una tabla, la registra en su propio catálogo de datos de AWS Glue. Athena utiliza el

catálogo de datos de AWS Glue para almacenar y recuperar estos metadatos, y los utiliza cuando se ejecutan consultas para analizar el conjunto de datos subyacente.

Independientemente de cómo se creen las tablas, su proceso de creación registra el conjunto de datos con Athena. Este registro se produce en AWS Glue Data Catalog y le permite a Athena ejecutar consultas sobre los datos. En el editor de consultas de Athena, se hace referencia a este catálogo (u origen de datos) con la etiqueta `AwsDataCatalog`.

Después de crear una tabla, puede utilizar las instrucciones de [SQL SELECT](#) para consultarla, incluida la obtención de [ubicaciones de archivo específicas para los datos de origen](#). Los resultados de la consulta se almacenan en [la ubicación de resultados de consulta que especifique](#) en Amazon S3.

El acceso al catálogo de datos de AWS Glue es posible a través de su cuenta de Amazon Web Services. Otros Servicios de AWS pueden compartir el catálogo de datos de AWS Glue, por lo que podrá ver bases de datos y tablas creadas en toda la organización mediante Athena y viceversa.

- Para crear una tabla manualmente:
  - Utilice la consola de Athena para ejecutar el Asistente de creación de tablas.
  - Utilice la consola de Athena para escribir instrucciones DDL de Hive en el Editor de consultas.
  - Utilice la API o la CLI de Athena para ejecutar una cadena con una consulta SQL que contenga instrucciones DDL.
  - Utilice el controlador JDBC u ODBC de Athena.

Al crear tablas y bases de datos manualmente, Athena utiliza instrucciones de lenguaje de definición de datos (DDL) de HiveQL como `CREATE TABLE`, `CREATE DATABASE` y `DROP TABLE`, internamente, para crear tablas y bases de datos en el AWS Glue Data Catalog.

Para comenzar, puede utilizar un tutorial en la consola de Athena o seguir una guía paso a paso en la documentación de Athena.

- Para utilizar el tutorial en la consola de Athena, elija el icono de información en la parte superior derecha de la consola y, a continuación, seleccione la pestaña Tutorial.
- Para ver un tutorial paso a paso sobre cómo crear una tabla y escribir consultas en el editor de consultas de Athena, consulte [Introducción](#).



# Introducción

Este tutorial le mostrará cómo utilizar Amazon Athena para consultar datos. Creará una tabla basada en datos de muestra almacenados en Amazon Simple Storage Service, la consultará y verificará los resultados de la consulta.

El tutorial utiliza recursos activos, por lo que se le cobrará por las consultas que ejecute. No se le cobrará por los datos de muestra en la ubicación que utiliza este tutorial, pero si carga sus propios archivos de datos a Amazon S3, sí se aplican cargos.

## Requisitos previos

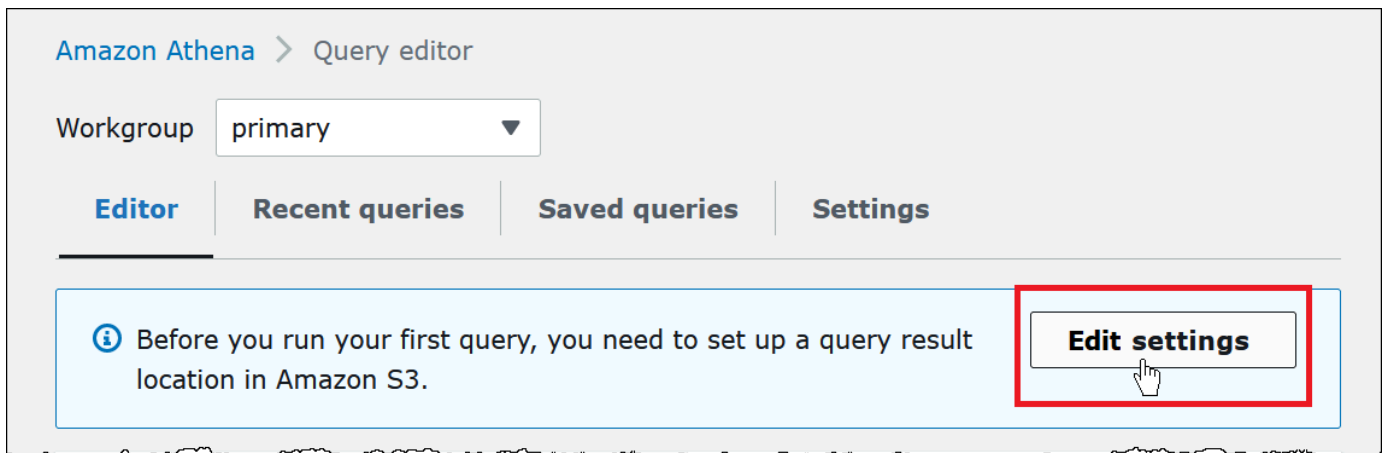
- Si aún no lo ha hecho, [regístrese para conseguir una Cuenta de AWS](#).
- Con la misma cuenta y Región de AWS (por ejemplo, Oeste de EE. UU. [Oregón]) que utiliza para Athena, siga los pasos para [crear un bucket en Amazon S3](#) a fin de retener los resultados de las consultas de Athena. Configuraré este bucket para que sea la ubicación de salida de consultas.

## Paso 1: Crear una base de datos

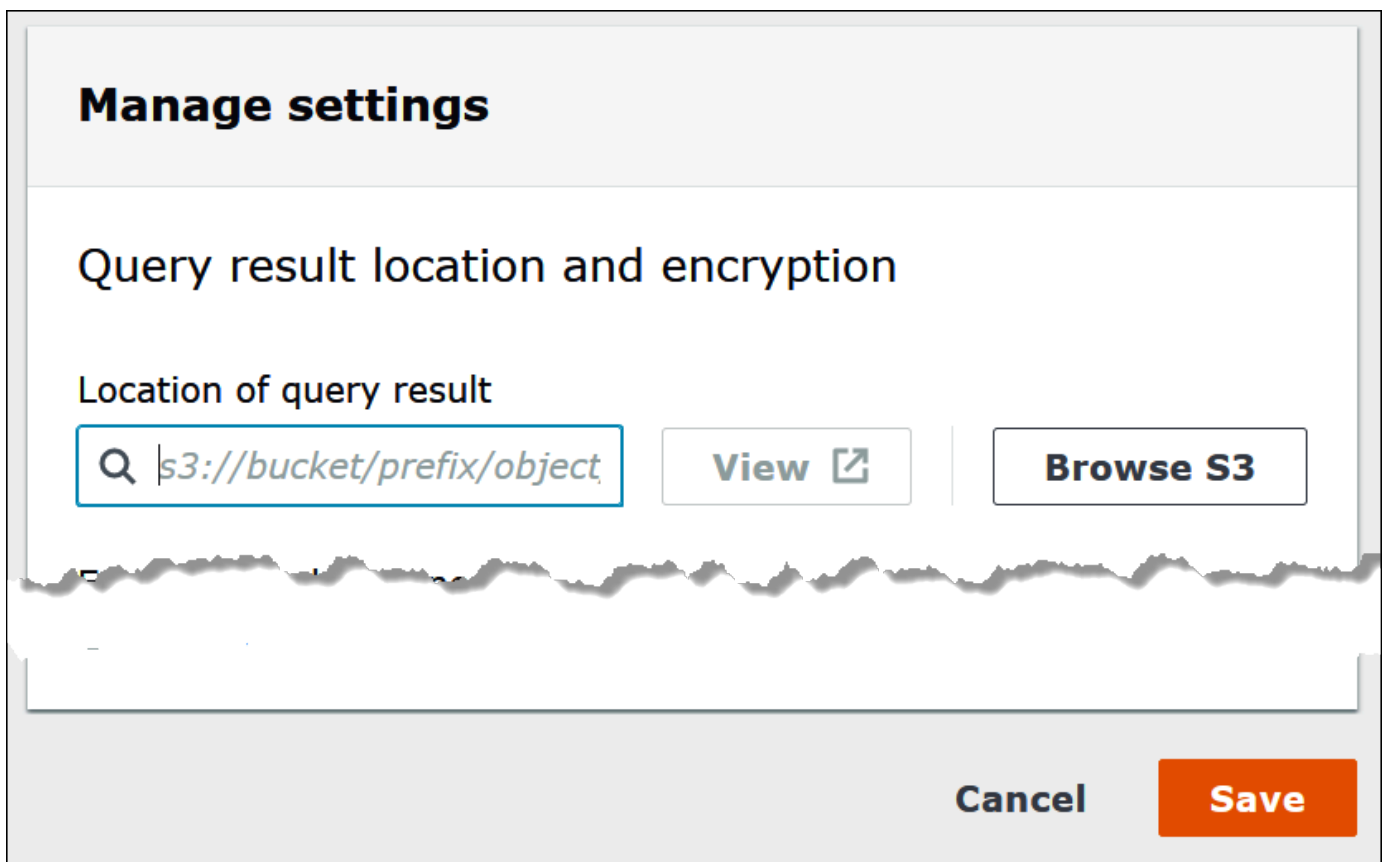
Primero debe crear una base de datos en Athena.

Para crear una base de datos de Athena

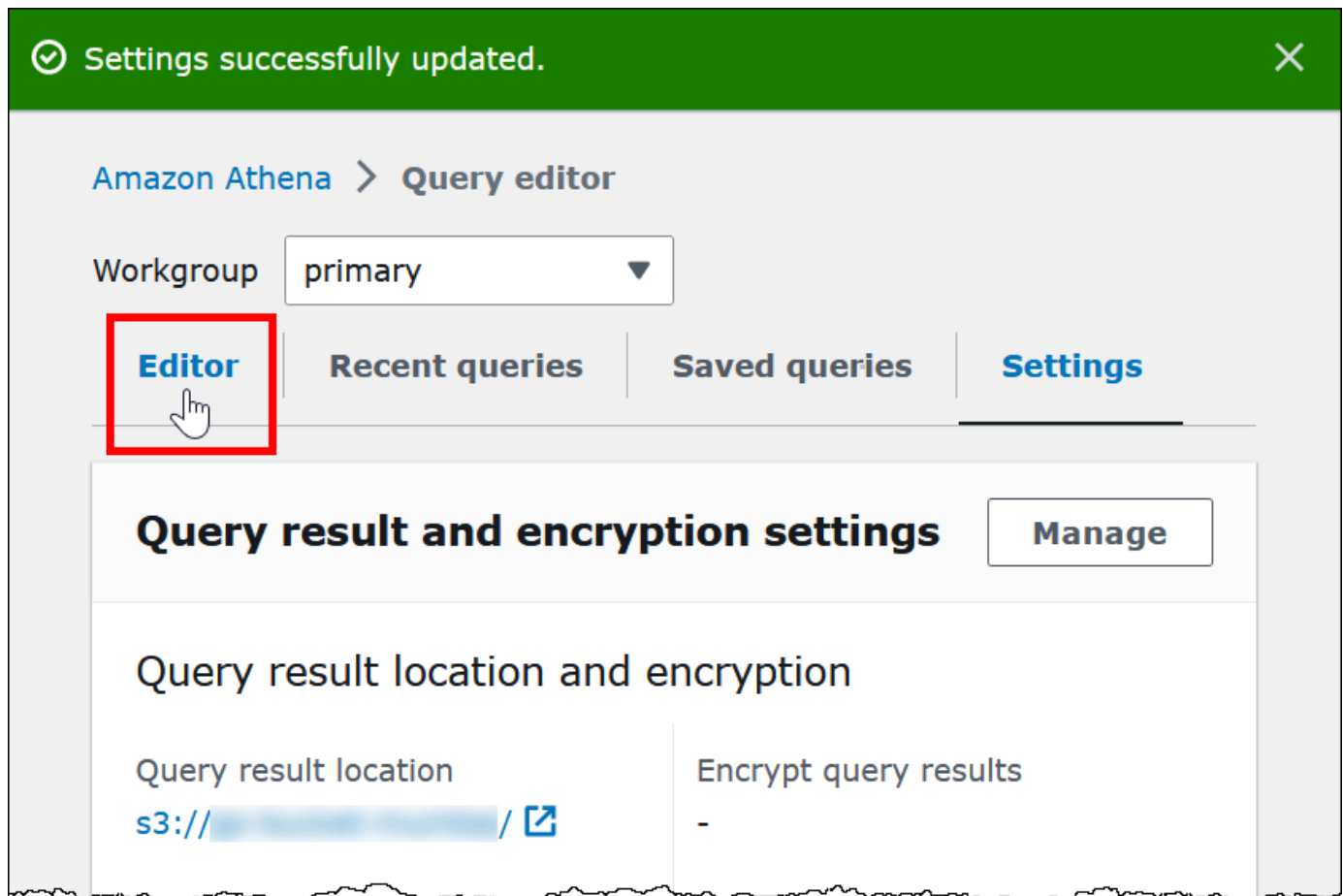
1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si es la primera vez que visita la consola de Athena en su Región de AWS actual, elija Explore the query editor (Explorar el editor de consultas) para abrir el editor de consultas. De lo contrario, Athena abre la consulta en el editor de consultas.
3. Elija Edit Settings (Editar configuración) para configurar una ubicación de resultados de la consulta en Amazon S3.



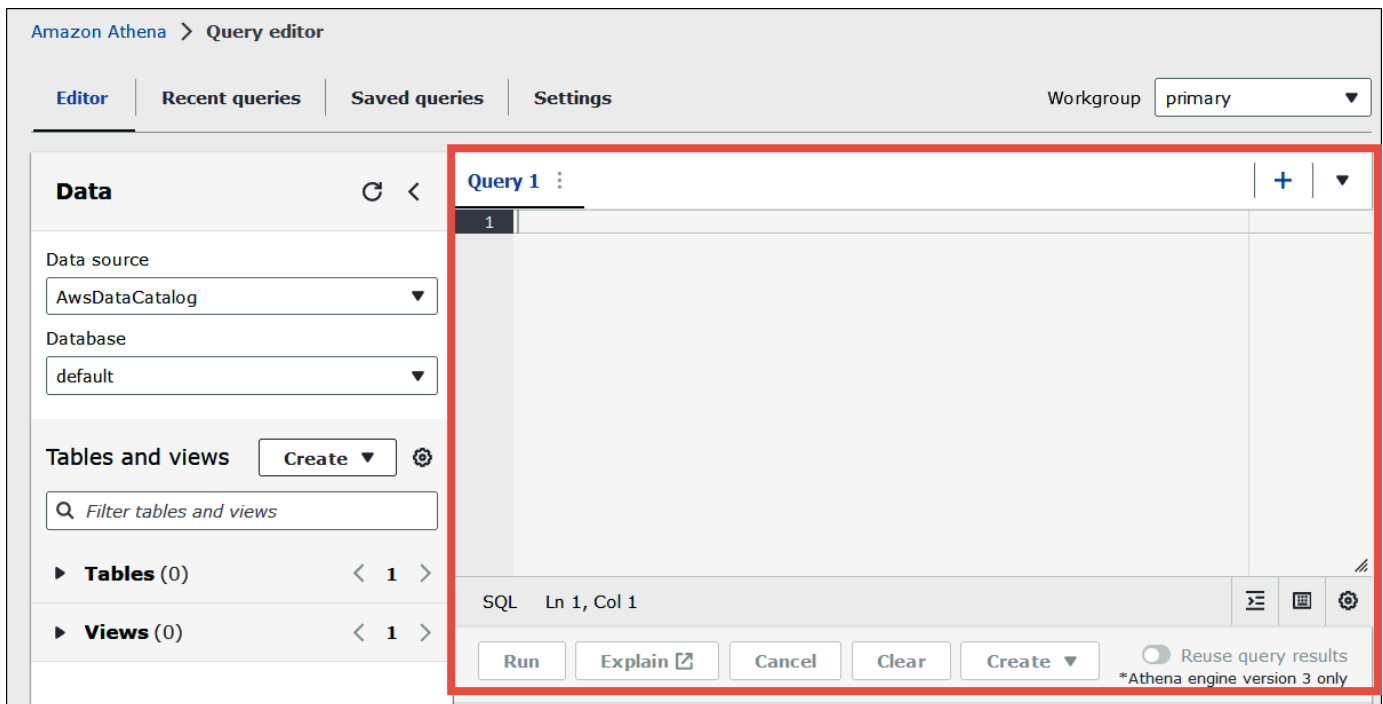
4. En Manage settings (Administrar configuración), realice una de las siguientes operaciones:
- En el cuadro Location of query result (Ubicación del resultado de la consulta), ingrese la ruta de acceso al bucket que creó en Amazon S3 para los resultados de la consulta. Prefije la ruta con `s3://`.
  - Elija Browse S3 (Navegar S3), elija el bucket de Amazon S3 que creó para su región actual y, a continuación, elija Choose (Elegir).



5. Elija Guardar.
6. Elija Editor para cambiar al editor de consultas.



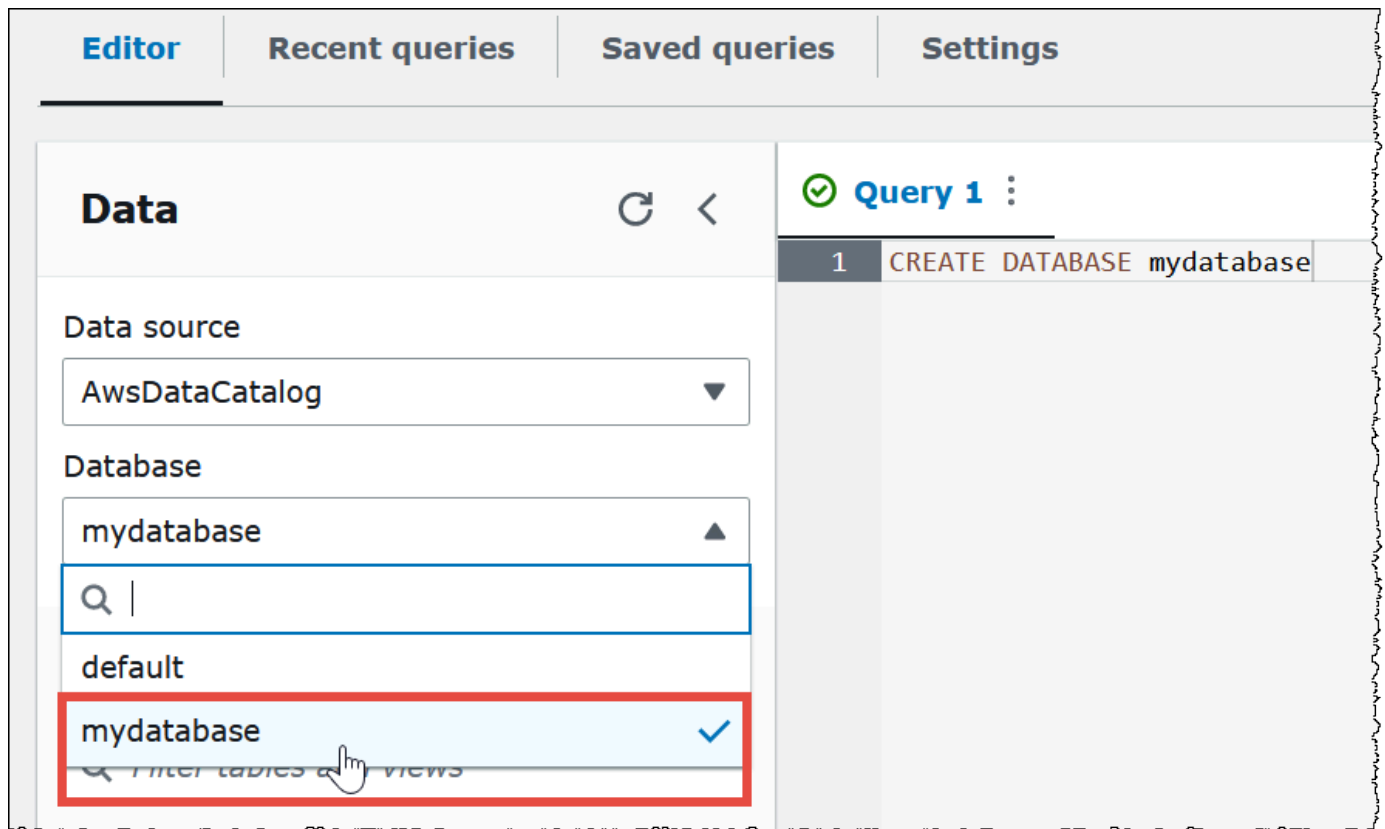
7. A la derecha del panel de navegación, puede utilizar el editor de consultas de Athena para ingresar y ejecutar consultas e instrucciones.



- Para crear una base de datos denominada `mydatabase`, escriba la siguiente instrucción `CREATE DATABASE`.

```
CREATE DATABASE mydatabase
```

- Elija `Run` (Ejecutar) o pulse **Ctrl+ENTER**.
- En la lista `Database` (Base de datos) de la izquierda, elija `mydatabase` para convertirla en su base de datos actual.



## Paso 2: crear una tabla

Ahora que tiene una base de datos, puede crear una tabla de Athena para ella. La tabla que cree se basará en los datos de registro de muestra de Amazon CloudFront en la ubicación `s3://athena-examples-myregion/cloudfront/plaintext/`, donde *myregion* es su Región de AWS actual.

Los datos de registro de muestra están en formato de valores separados por tabulaciones (TSV, tab-separated values), lo que significa que se utiliza un carácter de tabulación como delimitador para separar los campos. Los datos son similares al siguiente ejemplo. Por motivos de legibilidad, las pestañas del extracto se han convertido en espacios y el campo final se ha acortado.

```
2014-07-05 20:00:09 DFW3 4260 10.0.0.15 GET eabcd12345678.cloudfront.net /test-  
image-1.jpeg 200 - Mozilla/5.0[...]  
2014-07-05 20:00:09 DFW3 4252 10.0.0.15 GET eabcd12345678.cloudfront.net /test-  
image-2.jpeg 200 - Mozilla/5.0[...]  
2014-07-05 20:00:10 AMS1 4261 10.0.0.15 GET eabcd12345678.cloudfront.net /test-  
image-3.jpeg 200 - Mozilla/5.0[...]
```

Para permitir que Athena lea estos datos, puede ejecutar una instrucción directa `CREATE EXTERNAL TABLE` como la siguiente. La instrucción que crea la tabla define las columnas que se mapean a los datos. Además, especifica cómo se delimitan los datos y determina la ubicación de Amazon S3 que contiene los datos de muestra. Tenga en cuenta que, dado que Athena espera escanear todos los archivos de una carpeta, la cláusula `LOCATION` especifica una ubicación de carpeta de Amazon S3, no un archivo específico.

No utilice este ejemplo todavía, ya que tiene una limitación importante que se explicará en breve.

```
CREATE EXTERNAL TABLE IF NOT EXISTS cloudfront_logs (  
  `Date` DATE,  
  Time STRING,  
  Location STRING,  
  Bytes INT,  
  RequestIP STRING,  
  Method STRING,  
  Host STRING,  
  Uri STRING,  
  Status INT,  
  Referrer STRING,  
  ClientInfo STRING  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t'  
LINES TERMINATED BY '\n'  
LOCATION 's3://athena-examples-my-region/cloudfront/plaintext/';
```

En el ejemplo, se crea una tabla llamada `cloudfront_logs` y se especifica un nombre y un tipo de datos para cada campo. Estos campos se convierten en las columnas de la tabla. Dado que `date` es una [palabra reservada](#), se aplica escape con acentos graves (```). `ROW FORMAT DELIMITED` significa que Athena utilizará una biblioteca predeterminada llamada [LazySimpleSerDe](#) para realizar el trabajo real de analizar los datos. En el ejemplo también se especifica que los campos están separados por tabulaciones (`FIELDS TERMINATED BY '\t'`) y que cada registro en el archivo termina en un carácter de nueva línea (`LINES TERMINATED BY '\n'`). Por último, la cláusula `LOCATION` especifica la ruta de acceso en Amazon S3 donde se encuentran los datos reales que se van a leer.

Si tiene sus propios datos separados por tabulaciones o comas, puede utilizar una instrucción `CREATE TABLE` como en el ejemplo anterior, siempre y cuando los campos no contengan información anidada. Sin embargo, si tiene una columna como `ClientInfo`, que contiene información anidada y utiliza un delimitador diferente, necesitará un enfoque diferente.

## Extracción de datos del campo ClientInfo

Con respecto a los datos de muestra, aquí hay un ejemplo completo del campo final ClientInfo:

```
Mozilla/5.0%20(Android;%20U;%20Windows%20NT%205.1;%20en-US;%20rv:1.9.0.9)%20Gecko/2009040821%20IE/3.0.9
```

Como puede ver, este es un campo de múltiples valores. Debido a que la instrucción CREATE TABLE del ejemplo anterior especifica tabulaciones como delimitadores de campo, los componentes separados dentro del campo ClientInfo no se pueden dividir en columnas separadas. Por lo tanto, se requiere una nueva instrucción CREATE TABLE.

Puede utilizar una [expresión regular](#) (regex) que contenga grupos de regex para crear columnas a partir de los valores dentro del campo ClientInfo. Los grupos de expresiones regulares que especifique se convertirán en columnas de tabla independientes. Para usar una expresión regular en la instrucción CREATE TABLE, utilice una sintaxis como la siguiente. Esta sintaxis indica a Athena que utilice la biblioteca [SerDe de Regex](#) y la expresión regular que especifique.

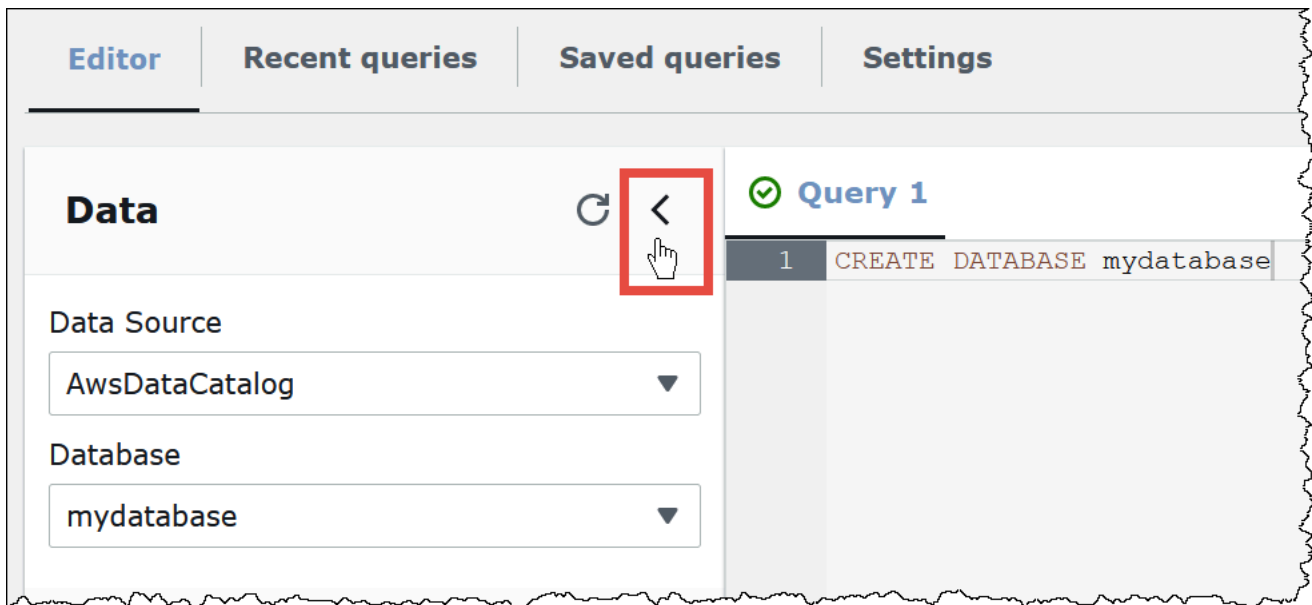
```
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'  
WITH SERDEPROPERTIES ("input.regex" = "regular_expression")
```

Las expresiones regulares pueden resultar útiles para crear tablas a partir de datos CSV o TSV complejos, pero pueden ser difíciles de escribir y mantener. Afortunadamente, existen otras bibliotecas que puede utilizar para formatos como JSON, Parquet y ORC. Para obtener más información, consulte [Formatos de datos y SerDes compatibles](#).

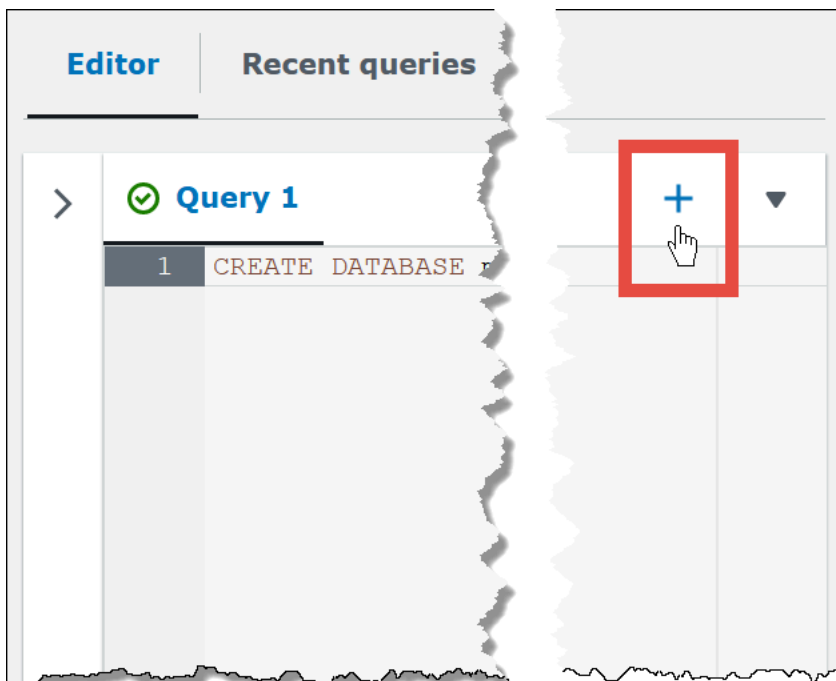
Ahora ya puede crear la tabla en el editor de consultas de Athena. La instrucción CREATE TABLE y expresiones regulares se le proporcionan.

Para crear una tabla en Athena

1. En el panel de navegación, en Database (Base de datos), asegúrese de que mydatabase esté seleccionado.
2. Para obtener más espacio en el editor de consultas, puede elegir el icono de flecha para contraer el panel de navegación.

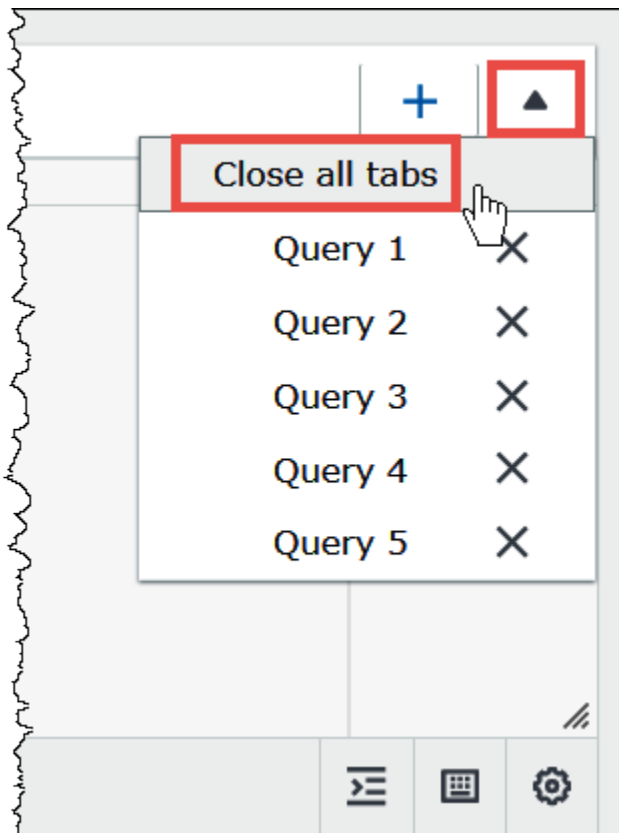


3. Para crear una pestaña para una nueva consulta, elija el signo más (+) en el editor de consultas. Puede tener hasta diez pestañas de consulta abiertas a la vez.



4. Para cerrar una o varias pestañas de consultas, elija la flecha situada junto al signo más. Para cerrar todas las pestañas a la vez, elija la flecha y, a continuación, elija Close all tabs (Cerrar todas las pestañas).

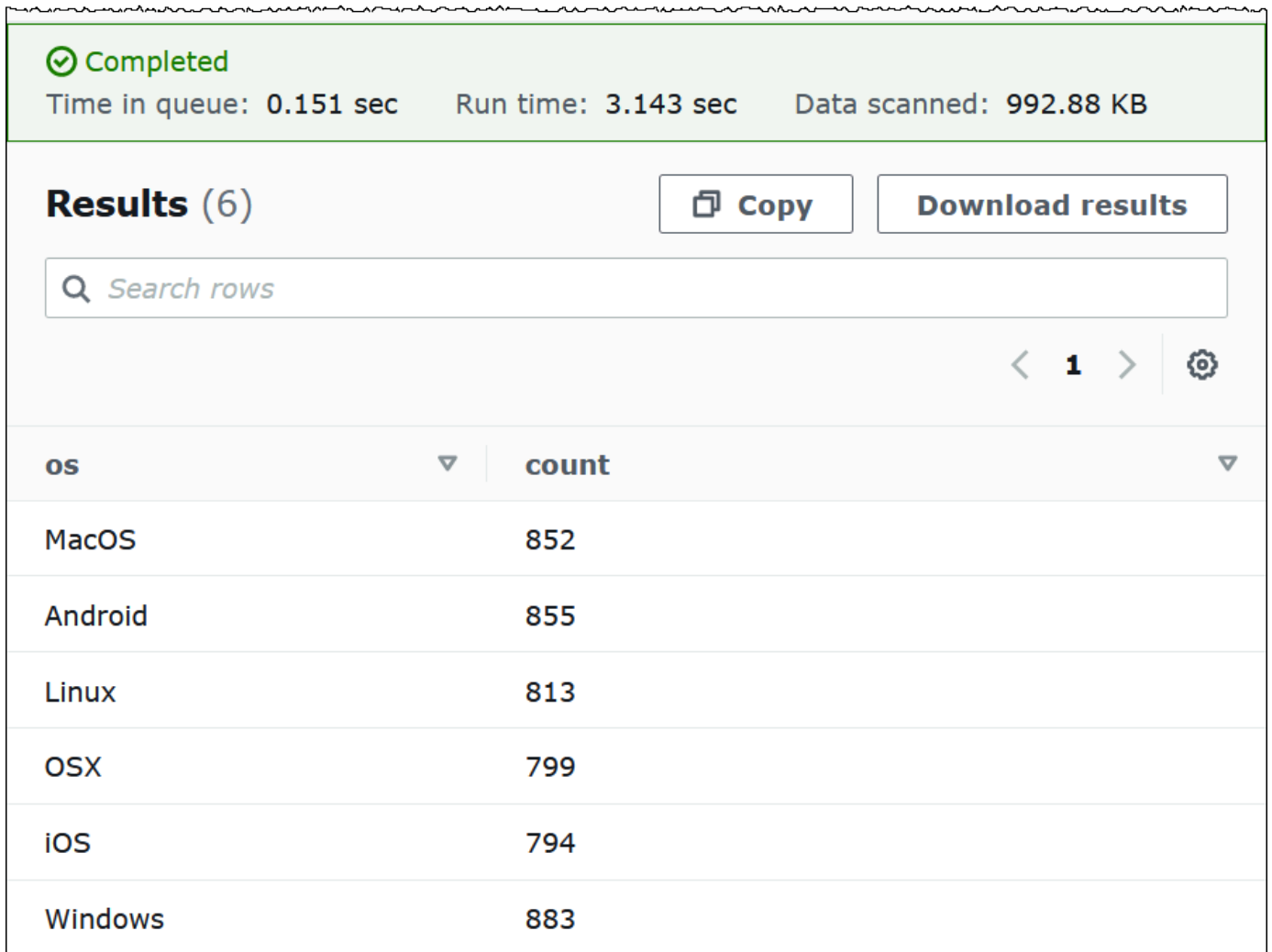




5. En el panel de consultas, escriba la siguiente instrucción CREATE EXTERNAL TABLE. La expresión regular desglosa la información del sistema operativo, el navegador y la versión del navegador del campo ClientInfo en los datos del registro.

```
CREATE EXTERNAL TABLE IF NOT EXISTS cloudfront_logs (  
  `Date` DATE,  
  Time STRING,  
  Location STRING,  
  Bytes INT,  
  RequestIP STRING,  
  Method STRING,  
  Host STRING,  
  Uri STRING,  
  Status INT,  
  Referrer STRING,  
  os STRING,  
  Browser STRING,  
  BrowserVersion STRING  
)  
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'  
WITH SERDEPROPERTIES (
```





Completed  
Time in queue: 0.151 sec    Run time: 3.143 sec    Data scanned: 992.88 KB

**Results (6)**    [Copy](#)    [Download results](#)

Search rows

< 1 > ⚙️

os	count
MacOS	852
Android	855
Linux	813
OSX	799
iOS	794
Windows	883

- Para guardar los resultados de una consulta en un archivo .csv, elija **Download results** (Descargar resultados).



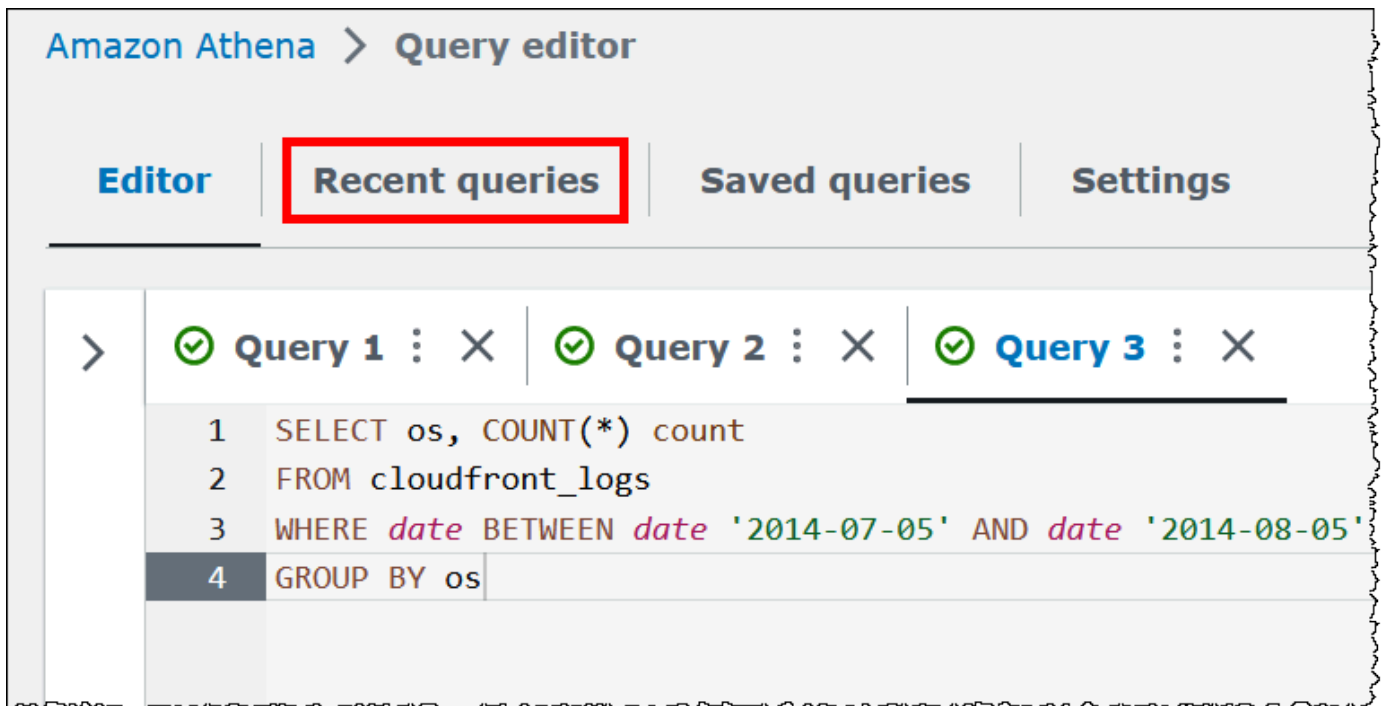
**Results (6)**    [Copy](#)    **Download results**

Search rows

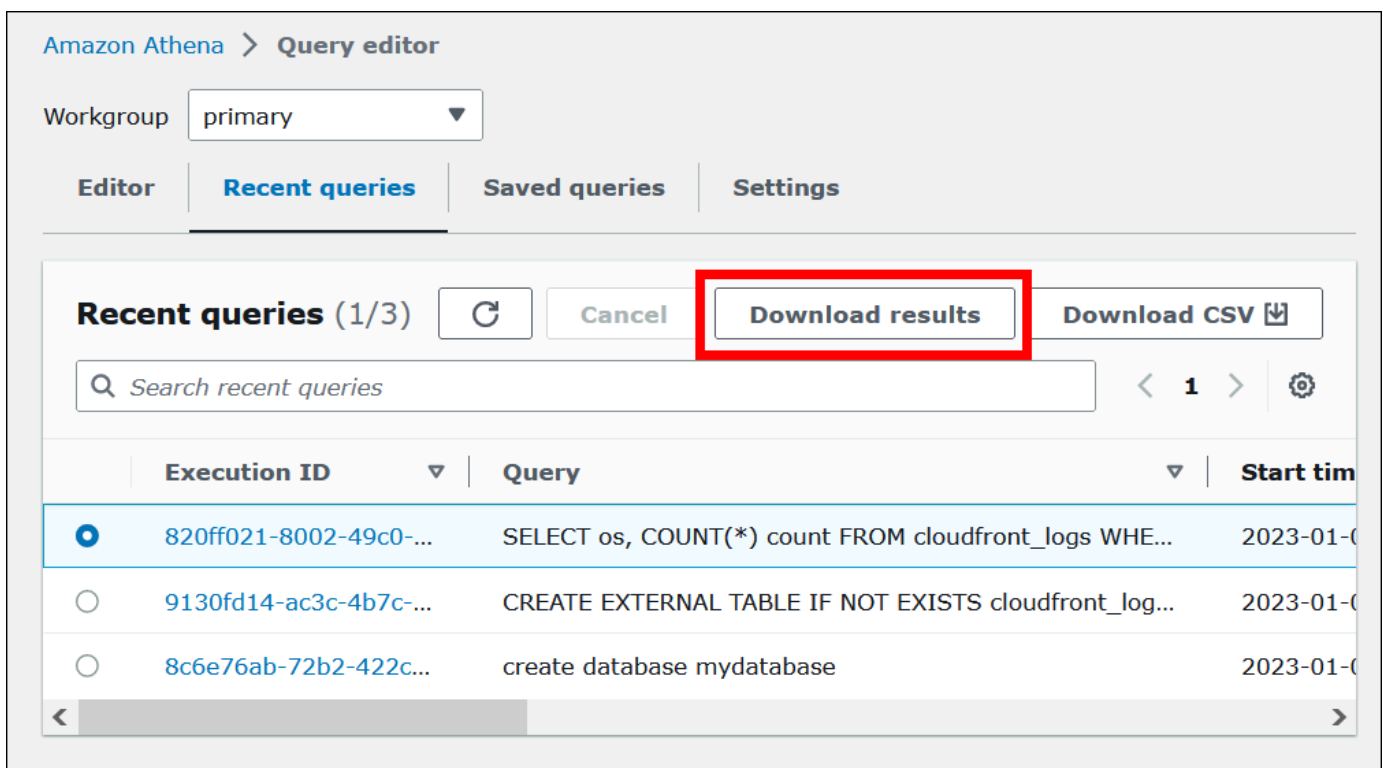
< 1 > ⚙️

os	count
----	-------

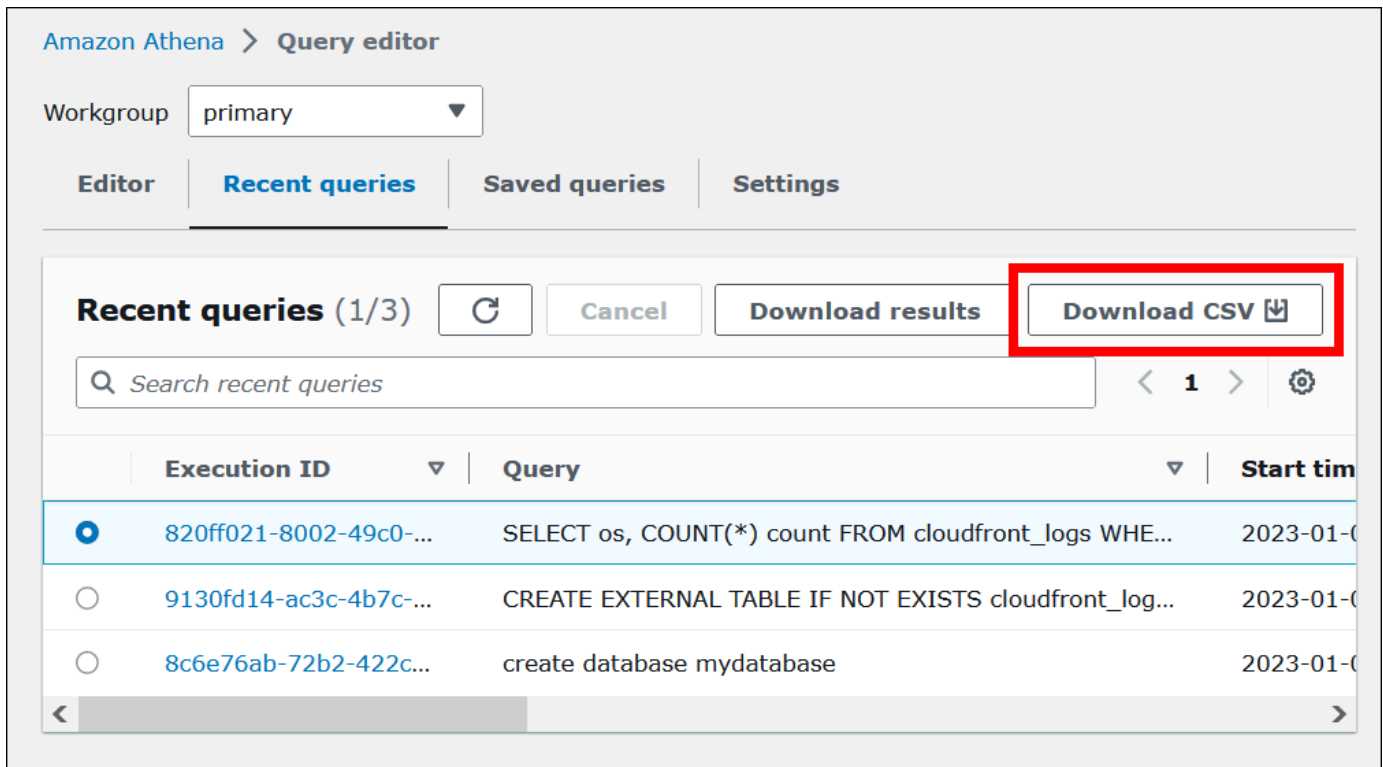
- Para ver o ejecutar consultas anteriores, elija la pestaña **Recent queries** (Consultas recientes).



5. Para descargar los resultados de una consulta anterior desde la pestaña Recent queries (Consultas recientes), seleccione la consulta y, a continuación, elija Download results (Descargar resultados). Las consultas se retienen durante 45 días.



- Para descargar una o más cadenas de consultas SQL recientes como un archivo CSV, elija Download CSV (Descargar CSV).



Para obtener más información, consulte [Trabajo con resultados de las consultas, consultas recientes y archivos de salida](#).

## Guardar las consultas

Puede guardar las consultas que cree o edite en el editor de consultas con un nombre. Athena almacena estas consultas en la pestaña Saved queries (Consultas guardadas). Puede utilizar la pestaña Saved queries (Consultas guardadas) para recuperar, ejecutar, cambiarles el nombre o eliminar las consultas guardadas. Para obtener más información, consulte [Uso de consultas guardadas](#).

## Atajos de teclado y sugerencias de escritura anticipada

El editor de consultas de Athena ofrece numerosos atajos de teclado para realizar acciones como ejecutar una consulta, formatear una consulta y efectuar operaciones de línea y buscar y reemplazar. Para obtener más información y una lista completa de atajos, consulte [Improve productivity by using keyboard shortcuts in Amazon Athena query editor](#) (Mejorar la productividad mediante el uso de atajos de teclado en el editor de consultas de Amazon Athena) en el blog sobre macrodatos de AWS.

El editor de consultas de Athena admite sugerencias de código de escritura anticipada para una experiencia de creación de consultas más rápida. Para que pueda escribir consultas SQL con mayor precisión y eficiencia, ofrece las siguientes características:

- A medida que escribe, aparecen sugerencias en tiempo real para palabras clave, variables locales, fragmentos y elementos del catálogo.
- Al escribir el nombre de una base de datos o de una tabla seguido de un punto, el editor muestra de forma oportuna una lista de tablas o columnas entre las que puede elegir.
- Al pasar el ratón por encima de una sugerencia de fragmento, aparece una sinopsis que muestra un breve resumen de la sintaxis y del uso del fragmento.
- Para mejorar la legibilidad del código, también se actualizaron las palabras clave y sus reglas de resaltado a fin de adaptarlas a la sintaxis más reciente de Trino y Hive.

Esta característica está habilitada de forma predeterminada. Para habilitar o deshabilitar la característica, utilice las preferencias del editor de código (icono con forma de engranaje) en la parte inferior derecha de la ventana del editor de consultas.

## Conexión con otros orígenes de datos

En este tutorial se utilizó un origen de datos de Amazon S3 en formato CSV. Para obtener información sobre el uso de Athena con AWS Glue, consulte [Uso de AWS Glue para conectarse a orígenes de datos en Amazon S3](#). También puede conectar Athena a distintos orígenes de datos mediante controladores ODBC y JDBC, metaalmacenes externos de Hive y conectores de origen de datos de Athena. Para obtener más información, consulte [Conexión con orígenes de datos](#).

## Conexión con orígenes de datos

Puede utilizar Amazon Athena para consultar datos almacenados en diferentes ubicaciones y formatos en un conjunto de datos. Este conjunto de datos puede estar en formato CSV, JSON, Avro, Parquet u cualquier otro.

Las tablas y bases de datos con las que trabaja Athena para ejecutar consultas se basan en metadatos. Los metadatos son datos sobre los datos subyacentes del conjunto de datos. La forma en que los metadatos describen su conjunto de datos se denomina esquema. Por ejemplo, un nombre de tabla, los nombres de columna de la tabla y el tipo de datos de cada columna son esquemas, guardados como metadatos, que describen un conjunto de datos subyacente. En

Athena, denominamos al sistema para organizar metadatos catálogo de datos o metaalmacén. La combinación de un conjunto de datos y el catálogo de datos que lo describe se denomina “origen de datos”.

La relación de los metadatos con un conjunto de datos subyacente depende del tipo de origen de datos con el que trabaje. Los orígenes de datos relacionales como MySQL, PostgreSQL y SQL Server integran estrechamente los metadatos con el conjunto de datos. En estos sistemas, los metadatos se escriben con mayor frecuencia cuando se escriben los datos. Otros orígenes de datos, como los creados con [Hive](#), le permiten definir metadatos sobre la marcha cuando se lee el conjunto de datos. El conjunto de datos puede estar en una variedad de formatos, por ejemplo, CSV, JSON, Parquet o Avro.

Athena admite AWS Glue Data Catalog de forma nativa. El AWS Glue Data Catalog es un catálogo de datos creado sobre otros conjuntos de datos y orígenes de datos como Amazon S3, Amazon Redshift y Amazon DynamoDB. También puede conectar Athena a otros orígenes de datos mediante una variedad de conectores.

## Temas

- [Integración con AWS Glue](#)
- [Uso del conector de datos de Athena para metaalmacén externo de Hive](#)
- [Uso de consulta federada de Amazon Athena](#)
- [Políticas de IAM para acceder a catálogos de datos](#)
- [Administración de orígenes de datos](#)
- [Uso de Amazon DataZone en Athena](#)

## Integración con AWS Glue

[AWS Glue](#) es un Servicio de AWS de extracción, transformación y carga (ETL, extract, transform, and load) completamente administrado. Una de sus capacidades clave es analizar y categorizar datos. Puede utilizar los rastreadores de AWS Glue para inferir de forma automática el esquema de bases de datos y tablas de sus datos en Amazon S3 y almacenar los metadatos asociados en el AWS Glue Data Catalog.

Athena utiliza el AWS Glue Data Catalog para almacenar y recuperar metadatos de tabla para los datos de Amazon S3 en su cuenta de Amazon Web Services. Los metadatos de la tabla permiten al motor de consultas de Athena saber cómo buscar, leer y procesar los datos que desea consultar.

Para crear un esquema de bases de datos y tablas en el AWS Glue Data Catalog, puede ejecutar un rastreador de AWS Glue desde Athena en un origen de datos, o puede ejecutar consultas de lenguaje de definición de datos (DDL) directamente en el Editor de consultas de Athena. A continuación, mediante el esquema de bases de datos y tablas que creó, puede utilizar consultas de manipulación de datos (DML) en Athena para consultar los datos.

Ahora puede registrar el AWS Glue Data Catalog desde una cuenta que no sea la suya. Una vez configurados los permisos de IAM necesarios para AWS Glue, puede utilizar Athena para ejecutar consultas entre cuentas. Para obtener más información, consulte [Acceso entre cuentas a los catálogos de datos de AWS Glue](#).

Para obtener más información sobre AWS Glue Data Catalog, consulte [Catálogo de datos y rastreadores en AWS Glue](#) en la Guía para desarrolladores de AWS Glue.

AWS Glue está sujeto a cargos por separado. Para obtener más información, consulte [Precios de AWS Glue](#).

## Temas

- [Uso de AWS Glue para conectarse a orígenes de datos en Amazon S3](#)
- [Registro de un AWS Glue Data Catalog de otra cuenta](#)
- [Prácticas recomendadas para utilizar Athena con AWS Glue](#)
- [Uso de la AWS CLI para recrear una base de datos de AWS Glue y sus tablas](#)

## Uso de AWS Glue para conectarse a orígenes de datos en Amazon S3

Athena puede conectarse a los datos almacenados en Amazon S3 mediante el AWS Glue Data Catalog para almacenar metadatos, como nombres de tablas y columnas. Una vez realizada la conexión, las bases de datos, las tablas y las vistas aparecen en el Editor de consultas de Athena.

Para definir la información de esquema para que AWS Glue la use, puede crear un rastreador de AWS Glue para recuperar la información automáticamente, o puede agregar manualmente una tabla e ingresar la información de esquema.

### Creación de un rastreador de AWS Glue

Para crear un rastreador, abra la consola de Athena y, a continuación, utilice la consola de AWS Glue de forma integrada. Al crear el rastreador, especifica una ubicación de datos en Amazon S3 para rastrearla.



Para crear un rastreador en AWS Glue desde la consola de Athena

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. En el editor de consultas, junto a Tablas y vistas, elija Crear y, a continuación, Rastreador de AWS Glue.
3. En la página Add crawler (Agregar rastreador) de AWS Glue, siga los pasos para crear un rastreador. Para obtener más información, consulte [Uso de rastreadores de AWS Glue](#) en esta guía y [Rellenar AWS Glue Data Catalog](#) en la Guía para desarrolladores de AWS Glue.

#### Note

Athena no reconoce los [patrones de exclusión](#) que especifica en un rastreador de AWS Glue. Por ejemplo, si tiene un bucket de Amazon S3 que contiene tanto `.csv` como `.json` y se excluyen los archivos `.json` desde el rastreador, Athena consulta ambos grupos de archivos. Para evitar esto, coloque los archivos que desea excluir en una ubicación diferente.

Adición de una tabla mediante un formulario

En el procedimiento siguiente se muestra cómo utilizar la consola de Athena para agregar una tabla mediante el formulario Create Table From S3 bucket data (Crear tabla a partir de datos de buckets de S3).

Para agregar una tabla e ingresar información del esquema mediante un formulario

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. En el editor de consultas, junto a Tables and views (Tablas y vistas), elija Create (Crear) y, a continuación, S3 bucket data (Datos del bucket de S3).
3. En el formulario Create Table From S3 bucket data (Crear tabla a partir de datos de buckets de S3), en Table name (Nombre de la tabla), ingrese un nombre para la tabla.
4. En Database configuration (Configuración de la base de datos), elija una base de datos existente o cree una nueva.
5. En Location of Input Data Set (Ubicación del conjunto de datos de entrada), especifique la ruta de acceso en Amazon S3 para la carpeta que contiene el conjunto de datos que desea procesar. No incluya ningún nombre de archivo en la ruta. Athena analiza todos los archivos de la carpeta que especifique. Si sus datos ya están particionados (por ejemplo,

s3://DOC-EXAMPLE-BUCKET/logs/year=2004/month=12/day=11/), ingrese únicamente la ruta base (por ejemplo, s3://DOC-EXAMPLE-BUCKET/logs/).

6. En Data Format (Formato de datos), elija una de las siguientes opciones:

- Para Table type (Tipo de tabla), elija Apache Hive, Apache Iceberg o Delta Lake. Athena usa el tipo de tabla Apache Hive como tipo de tabla predeterminado. Para obtener información acerca de la consulta de tablas de Apache Iceberg en Athena, consulte [Uso de tablas de Apache Iceberg](#). Para obtener información acerca del uso de tablas de Delta Lake en Athena, consulte [Consulta de tablas de Linux Foundation Delta Lake](#).
- En File format (Formato de archivo), elija el formato de archivo o registro en el que se encuentran los datos.
  - Para la opción Text File with Custom Delimiters (Archivo de texto con delimitadores personalizados), especifique un terminador de campo (es decir, un delimitador de columna). Si lo desea, puede especificar un Collection terminator (Terminador de colección) que marque el final de un tipo de matriz o un Collection terminator (Terminador de colección) que marque el final de un tipo de datos de mapa.
- SerDe library (Biblioteca SerDe): una biblioteca SerDe (serializador-deserializador) analiza un formato de datos determinado para que Athena pueda crearle una tabla. Para la mayoría de los formatos, se elige una biblioteca SerDe predeterminada. Para los siguientes formatos, elija una biblioteca según sus requisitos:
  - Apache Web Logs (Registros web de Apache): elija la biblioteca RegexSerDe o GrokSerDe. Para RegexSerDe, proporcione una expresión regular en el cuadro Regex definition (Definición de regex). Para GrokSerDe, proporcione una serie de expresiones regulares con nombre para la propiedad SerDe input . format. Las expresiones regulares nombradas son más fáciles de leer y mantener que las expresiones regulares. Para obtener más información, consulte [Consulta de registros de Apache almacenados en Amazon S3](#).
  - CSV: elija LazySimpleSerDe si sus datos separados por comas no contienen valores entre comillas dobles o si utilizan el formato java.sql.Timestamp. Elija OpenCSVSerDe si los datos incluyen comillas o utilizan el formato numérico de UNIX de TIMESTAMP (por ejemplo, 1564610311). Para obtener más información, consulte [LazySimpleSerDe para CSV, TSV y archivos con delimitación personalizada](#) y [OpenCSVSerDe para el procesamiento de CSV](#).
  - JSON: elija la biblioteca SerDe JSON de OpenX o Hive. Ambos formatos esperan que cada documento JSON esté en una sola línea de texto y que los campos no estén separados por caracteres de nueva línea. El SerDe de OpenX ofrece algunas propiedades adicionales. Para obtener más información sobre estas propiedades, consulte [El SerDe JSON de](#)

[OpenX](#). Para obtener más información acerca de SerDe de Hive, consulte [El SerDe JSON de Hive](#).

Para obtener más información sobre el uso de bibliotecas SerDe en Athena, consulte [Formatos de datos y SerDes compatibles](#).

7. En SerDe properties (Propiedades de SerDe), agregue, edite o elimine propiedades y valores de acuerdo con la biblioteca de SerDe que esté utilizando y sus requisitos.
  - Para agregar una propiedad de SerDe, elija Add SerDe property (Agregar propiedad de SerDe).
  - En el campo Name (Nombre), ingrese el nombre de la propiedad.
  - En el campo Value (Valor), ingrese un valor para la propiedad.
  - Para eliminar una propiedad de SerDe, elija Remove (Eliminar).
8. En Table properties (Propiedades de la tabla), elija o edite las propiedades de la tabla según sus requisitos.
  - En Write compression (Compresión de escritura), elija una opción de compresión. La disponibilidad de la opción de compresión de escritura y de las opciones de compresión disponibles depende del formato de los datos. Para obtener más información, consulte [Compatibilidad con la compresión de Athena](#).
  - En Encryption (Cifrado), seleccione Encrypted data set (Conjunto de datos cifrados) si los datos subyacentes están cifrados en Amazon S3. Esta opción establece la propiedad de la tabla `has_encrypted_data` en True (Verdadero) en la instrucción CREATE TABLE.
9. En Column details (Detalles de la columna), ingrese los nombres y los tipos de datos de las columnas que desee agregar a la tabla.
  - Para agregar más columnas de una a la vez, elija Add a column (Agregar una columna).
  - Para agregar más columnas rápidamente, elija Bulk add columns (Agregar columnas en bloque). En el cuadro de texto, ingrese una lista de columnas separadas por comas con el formato `column_name data_type, column_name data_type, [...]` y, a continuación, elija Add (Agregar).
10. (Opcional) En Partition details (Detalles de la partición), agregue uno o varios nombres y tipos de datos de columnas. La partición mantiene los datos relacionados juntos en función de los valores de las columnas y puede ayudar a reducir la cantidad de datos escaneados por consulta. Para obtener información sobre las particiones, consulte [Particiones de datos en Athena](#).

11. (Opcional) En Bucketing (Agrupación en buckets), puede especificar una o más columnas que tengan filas que desee agrupar y, a continuación, colocarlas en varios buckets. Esto le permite consultar solo el bucket que desea leer cuando se especifica el valor de las columnas agrupadas en buckets.
  - En Buckets, seleccione una o más columnas que tengan un gran número de valores únicos (por ejemplo, una clave principal) y que se usen con frecuencia para filtrar los datos de las consultas.
  - En Number of buckets (Número de buckets), ingrese un número que permita que los archivos tengan un tamaño óptimo. Para obtener información, consulte [Top 10 Performance Tuning Tips for Amazon Athena](#) (Los 10 principales consejos de ajuste de rendimiento de Amazon Athena) en el Blog de macrodatos de AWS.
  - Para especificar las columnas agrupadas, la instrucción CREATE TABLE utilizará la siguiente sintaxis:

```
CLUSTERED BY (bucketed_columns) INTO number_of_buckets BUCKETS
```

#### Note

La opción Bucketing (Agrupación en buckets) no está disponible para los tipos de tablas de Iceberg.

12. El recuadro Preview table query (Vista previa de consulta de tablas) muestra la instrucción CREATE TABLE generada por la información ingresada en el formulario. La instrucción de vista previa no se puede editar directamente. Para cambiar la instrucción, modifique los campos del formulario o  [Cree la instrucción directamente](#) en el editor de consultas en lugar de utilizar el formulario.
13. Elija Create table (Crear tabla) para ejecutar la instrucción generada en el editor de consultas y crear la tabla.

## Registro de un AWS Glue Data Catalog de otra cuenta

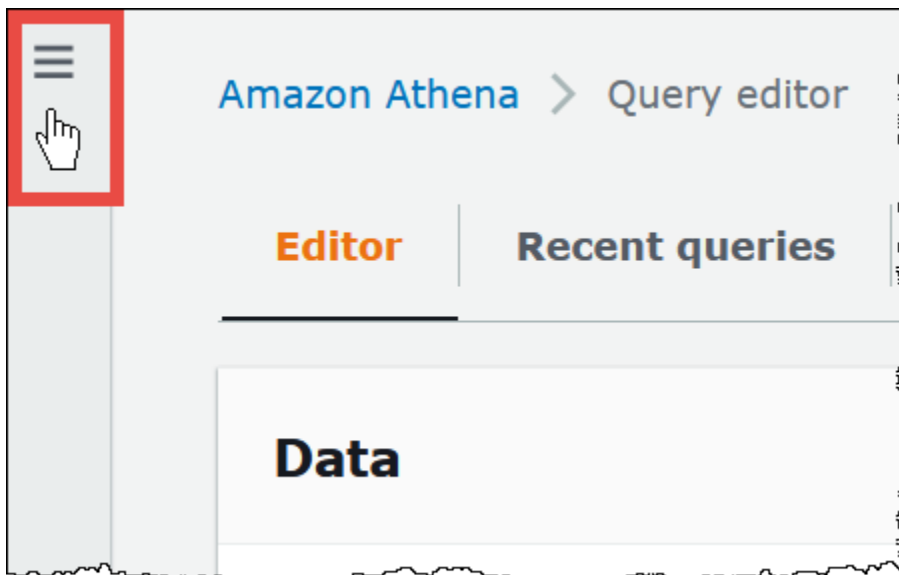
Puede utilizar la función de catálogo de AWS Glue de cuentas cruzadas de Athena para registrar un catálogo AWS Glue desde una cuenta que no sea la suya. Después de configurar los permisos de IAM necesarios para AWS Glue y registrar el catálogo como recurso de DataCatalog de Athena, puede utilizar Athena para ejecutar consultas entre cuentas. Para obtener información sobre la

configuración de los permisos necesarios, consulte [Acceso entre cuentas a los catálogos de datos de AWS Glue](#).

El procedimiento siguiente muestra cómo utilizar la consola de Athena para configurar un AWS Glue Data Catalog en una cuenta de Amazon Web Services distinta de la suya como origen de datos.

Para registrar un AWS Glue Data Catalog de otra cuenta

1. Siga los pasos de [Acceso entre cuentas a los catálogos de datos de AWS Glue](#) para asegurarse de que tiene permisos para consultar el catálogo de datos en la otra cuenta.
2. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
3. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.



4. Elija Edit Data Source (Editar origen de datos).
5. En la parte superior derecha de la consola, seleccione Create data source (Crear origen de datos).
6. En la página Choose a data source (Elegir un origen de datos), para Data Sources (Orígenes de datos), elija S3 -AWS Glue Data Catalog y, a continuación, elija Next (Siguiente).
7. En la página Introducir detalles del origen de datos, en la sección AWS Glue Data Catalog, para Elegir un AWS Glue Data Catalog, elija AWS Glue Data Catalog en otra cuenta.
8. En Dataset details (Detalles del origen de datos), ingrese la siguiente información:
  - Data source name (Nombre del origen de datos): ingrese el nombre que desea utilizar en las consultas SQL para hacer referencia al catálogo de datos de la otra cuenta.

- Descripción: (opcional) ingrese una descripción del catálogo de datos en la otra cuenta.
  - ID del catálogo: ingrese el ID de cuenta de Amazon Web Services de 12 dígitos de la cuenta a la que pertenece el catálogo de datos. El ID de cuenta de Amazon Web Services es el ID del catálogo.
9. (Opcional) En Tags (Etiquetas), ingrese pares clave-valor que quiera asociar con el origen de datos. Para obtener más información acerca de las etiquetas, consulte [Etiquetado de recursos de Athena](#).
  10. Elija Siguiente.
  11. En la página Review and create (Revisar y crear), revise la información que ha proporcionado y, a continuación, elija Create data source (Crear un origen de datos). En la página Data source details (Detalles de origen de datos) se enumeran las bases de datos y etiquetas del catálogo de datos que ha registrado.
  12. Elija Edit Data Source (Editar origen de datos). El catálogo de datos que ha registrado se muestra en la columna Data source name (Nombre de origen de datos).
  13. Para ver o editar información sobre el nuevo catálogo de datos, elija el catálogo y, a continuación, elija Actions (Acciones) y Edit (Editar).
  14. Para eliminar el nuevo catálogo de datos, elija el catálogo y, a continuación, elija Actions (Acciones) y Delete (Eliminar).

Para obtener más información, consulte [Query cross-account AWS Glue Data Catalogs using Amazon Athena](#) en el blog sobre macrodatos de AWS.

## Prácticas recomendadas para utilizar Athena con AWS Glue

Al utilizar Athena con el AWS Glue Data Catalog, puede usar AWS Glue para crear las bases de datos y las tablas (esquemas) que consultará en Athena, o bien puede utilizar Athena para crear un esquema y entonces utilizarlo en AWS Glue u otros servicios relacionados. Este tema contiene consideraciones y prácticas recomendadas para utilizar ambos métodos.

Desde el punto de vista tecnológico, Athena utiliza Trino para procesar instrucciones DML y Hive para procesar instrucciones DDL que se usan en la creación y modificación de esquemas. Con estas tecnologías, es preciso respetar un par de convenciones para que Athena y AWS Glue funcionen bien conjuntamente.

En este tema

- [Nombres de base de datos, tablas y columnas](#)

- [Uso de rastreadores AWS Glue](#)
  - [Programación de un rastreador para mantener AWS Glue Data Catalog y Simple Storage Service \(Amazon S3\) sincronizados](#)
  - [Uso de varios orígenes de datos con rastreadores](#)
  - [Sincronización de un esquema de partición para evitar "HIVE\\_PARTITION\\_SCHEMA\\_MISMATCH"](#)
  - [Actualización de los metadatos de una tabla](#)
- [Trabajo con archivos CSV](#)
  - [Datos de archivos CSV entre comillas](#)
  - [Archivos CSV con encabezados](#)
- [Indexación y filtrado de particiones de AWS Glue](#)
- [Uso de datos geoespaciales](#)
- [Uso de AWS Glue para realizar trabajos de ETL con Athena](#)
  - [Creación de tablas con Athena para los trabajos de ETL de AWS Glue](#)
  - [Uso de trabajos de ETL para optimizar el rendimiento de las consultas](#)
  - [Conversión de los tipos de datos SMALLINT y TINYINT en INT cuando realiza conversiones a ORC](#)
  - [Automatización de trabajos de ETL de AWS Glue](#)

## Nombres de base de datos, tablas y columnas

Cuando cree un esquema en AWS Glue para realizar consultas en Athena, tenga en cuenta lo siguiente:

- Los caracteres aceptables para los nombres de base de datos, los nombres de tablas y los nombres de columnas en AWS Glue deben ser una cadena de UTF-8. La cadena debe tener 1 byte de largo, como mínimo, y 255 bytes de largo, como máximo. Los caracteres que se pueden usar incluyen espacios y se definen mediante el siguiente patrón de cadena de una sola línea:

```
[\\u0020-\\uD7FF\\uE000-\\uFFFF\\uD800\\uDC00-\\uDBFF\\uDFFF\\t]*
```

- Actualmente, el patrón de expresiones regulares AWS Glue permite añadir espacios iniciales al principio de los nombres. Dado que estos espacios iniciales pueden ser difíciles de detectar y pueden provocar problemas de usabilidad tras su creación, evite crear nombres de objetos que tengan espacios iniciales.

- Si usa una plantilla [AWS::Glue::Database](#) de AWS CloudFormation para crear una base de datos de AWS Glue y no especifica ningún nombre de la base de datos, AWS Glue genera automáticamente un nombre de la base de datos con el formato *nombre\_recurso-cadena\_aleatoria* que no es compatible con Athena.
- Puede utilizar el administrador de catálogo de AWS Glue para cambiar el nombre de las columnas, pero no los nombres de las tablas ni los nombres de bases de datos. Para evitar esta limitación, debe usar una definición de la base de datos antigua para crear una base de datos con el nombre nuevo. A continuación, utilice las definiciones de las tablas de la base de datos antigua para volver a crear las tablas en la base de datos nueva. Para ello, puede utilizar la AWS CLI o el SDK de AWS Glue. Para ver los pasos, consulte [Uso de la AWS CLI para recrear una base de datos de AWS Glue y sus tablas](#).

Para obtener más información sobre las bases de datos y las tablas de AWS Glue, consulte [Bases de datos](#) y [Tablas](#) en la Guía para desarrolladores de AWS Glue.

## Uso de rastreadores AWS Glue

Los rastreadores de AWS Glue son útiles para detectar el esquema de conjuntos de datos y registrarlos como tablas en el catálogo de datos de AWS Glue. Los rastreadores revisan sus datos y establecen cuál es el esquema. Además, los rastreadores puede detectar y registrar particiones. Para obtener más información, consulte [Definición de rastreadores](#) en la Guía para desarrolladores de AWS Glue. Las tablas de datos que se han rastreado correctamente se pueden consultar desde Athena.

### Note

Athena no reconoce los [patrones de exclusión](#) que especifica en un rastreador de AWS Glue. Por ejemplo, si tiene un bucket de Amazon S3 que contiene tanto `.csv` como `.json` y se excluyen los archivos `.json` desde el rastreador, Athena consulta ambos grupos de archivos. Para evitar esto, coloque los archivos que desea excluir en una ubicación diferente.

## Programación de un rastreador para mantener AWS Glue Data Catalog y Simple Storage Service (Amazon S3) sincronizados

Los rastreadores de AWS Glue se pueden configurar para ejecutarse siguiendo una programación o bajo demanda. Para obtener más información, consulte [Programaciones basadas en el tiempo para trabajos y rastreadores](#) en la Guía para desarrolladores de AWS Glue.



Si los datos para una tabla con particiones llegan a una hora establecida, puede configurar un rastreador de AWS Glue para que se ejecute a una hora programada y detecte y actualice las particiones de la tabla. De esta manera, puede eliminar la necesidad de ejecutar un comando `MSCK REPAIR` potencialmente largo y costoso, o ejecutar manualmente un comando `ALTER TABLE ADD PARTITION`. Para obtener más información, consulte [Particiones de tablas](#) en la Guía para desarrolladores de AWS Glue.

### Uso de varios orígenes de datos con rastreadores

Cuando un rastreador de AWS Glue analiza Amazon S3 y detecta varios directorios, utiliza una heurística para determinar dónde se encuentra la raíz de una tabla en la estructura de directorios y qué directorios son particiones de tabla. En algunos casos en que el esquema detectado en dos o más directorios es similar, el rastreador puede tratarlos como si fueran particiones en vez de tablas diferentes. Una forma de ayudar al rastreador a detectar tablas individuales consiste en añadir el directorio raíz de cada tabla como almacén de datos para el rastreador.

Las siguientes particiones en Amazon S3 son un ejemplo:

```
s3://DOC-EXAMPLE-BUCKET/folder1/table1/partition1/file.txt
s3://DOC-EXAMPLE-BUCKET/folder1/table1/partition2/file.txt
s3://DOC-EXAMPLE-BUCKET/folder1/table1/partition3/file.txt
s3://DOC-EXAMPLE-BUCKET/folder1/table2/partition4/file.txt
s3://DOC-EXAMPLE-BUCKET/folder1/table2/partition5/file.txt
```

Si los esquemas de `table1` y `table2` son similares, y se ha establecido un único origen de datos para `s3://DOC-EXAMPLE-BUCKET/folder1/` en AWS Glue, el rastreador puede crear una única tabla con dos columnas de partición: una columna de partición que contenga `table1` y `table2`, y otra columna que contenga de `partition1` a `partition5`.

Para que el rastreador de AWS Glue cree dos tablas diferentes configure el rastreador con dos orígenes de datos, `s3://DOC-EXAMPLE-BUCKET/folder1/table1/` y `s3://DOC-EXAMPLE-BUCKET/folder1/table2`, tal y como se muestra en el siguiente procedimiento.

Para agregar otro almacén de datos de S3 a un rastreador existente en AWS Glue

1. Inicie sesión en la AWS Management Console y abra la consola de AWS Glue en <https://console.aws.amazon.com/glue/>.
2. En el panel de navegación, elija Crawlers (Rastreadores).
3. Elija el enlace a su rastreador y, a continuación, elija Edit (Editar).

4. Para Step 2: Choose data sources and classifiers (Paso 2: Elegir orígenes de datos y clasificadores), elija Edit (Editar).
5. En Data sources (Orígenes de datos), elija Add a data source (Agregar un origen de datos).
6. En el cuadro de diálogo Add data source (Agregar origen de datos), en S3 path (Ruta de S3), elija Browse (Examinar).
7. Elija el bucket que desee actualizar y, a continuación, elija Choose (Elegir).

El origen de datos que ha agregado aparece en la lista Data sources (Orígenes de datos).

8. Elija Siguiente.
9. En la página Configurar ajustes de seguridad, cree o elija un rol de IAM para el rastreador y, a continuación, elija Siguiente.
10. Asegúrese de que la ruta de S3 termina en una barra diagonal y, a continuación, seleccione Add an S3 data source (Agregar un origen de datos de S3).
11. En la página Set output and scheduling (Definir la salida y la programación), en Output configuration (Configuración de salida), elija la base de datos de destino.
12. Elija Siguiente.
13. En la página Review and update (Revisar y actualizar), revise las elecciones que ha realizado. Para editar un paso, seleccione Edit (Editar).
14. Elija Actualizar.

Sincronización de un esquema de partición para evitar "HIVE\_PARTITION\_SCHEMA\_MISMATCH"

Por cada tabla de AWS Glue Data Catalog que tenga columnas de partición, el esquema se almacena en el nivel de tabla y para cada partición individual de la tabla. El esquema de las particiones lo rellena un rastreador de AWS Glue basándose en la muestra de datos que lee en la partición. Para obtener más información, consulte [Uso de varios orígenes de datos con rastreadores](#).

Cuando Athena ejecuta una consulta, valida el esquema de la tabla y el esquema de las particiones necesarias para la consulta. En la validación se comparan los tipos de datos de la columna en orden y se confirma que estos tipos de datos coinciden en el caso de las columnas que se solapan. De esta forma, se evita que se produzcan operaciones imprevistas como añadir o eliminar columnas del medio de una tabla. Si Athena detecta que el esquema de una partición es diferente del esquema de la tabla, es posible que Athena no pueda procesar la consulta y genere el error HIVE\_PARTITION\_SCHEMA\_MISMATCH.

Existen varias maneras de resolver este problema. En primer lugar, si los datos se añadieron por error, puede eliminar los archivos de datos que generan la diferencia en el esquema, anular la partición y volver a rastrear los datos. En segundo lugar, puede anular la partición individual y, a continuación, ejecutar `MSCK REPAIR` en Athena para volver a crear la partición utilizando el esquema de la tabla. Esta segunda opción solo funciona si está convencido de que el esquema aplicado continuará leyendo los datos correctamente.

### Actualización de los metadatos de una tabla

Después de un rastreo, el rastreador de AWS Glue asigna automáticamente determinados metadatos de tabla para hacerlos compatibles con otras tecnologías externas como Apache Hive, Presto y Spark. Ocasionalmente, el rastreador puede asignar incorrectamente las propiedades de los metadatos. Corrija manualmente las propiedades en AWS Glue antes de consultar la tabla con Athena. Para obtener más información, consulte [Visualización y edición de los detalles de una tabla](#) en la Guía para desarrolladores de AWS Glue.

AWS Glue puede asignar erróneamente metadatos cuando un archivo CSV tiene todos los campos de datos entre comillas, lo que hace que se interprete incorrectamente la propiedad `serializationLib`. Para obtener más información, consulte [Datos de archivos CSV entre comillas](#).

### Trabajo con archivos CSV

Los archivos CSV ponen ocasionalmente los valores de datos destinados a cada columna entre comillas y es posible que los archivos CSV contengan valores de encabezado que no formen parte de los datos que se van a analizar. Cuando utilice AWS Glue para crear esquemas a partir de estos archivos, siga las instrucciones de esta sección.

#### Datos de archivos CSV entre comillas

Puede tener un archivo CSV que contenga campos de datos entre comillas dobles como el siguiente ejemplo:

```
"John","Doe","123-555-1231","John said \"hello\""  
"Jane","Doe","123-555-9876","Jane said \"hello\""
```

Para ejecutar una consulta en Athena en una tabla creada a partir de un archivo CSV que contiene valores entre comillas, debe modificar las propiedades de la tabla en AWS Glue para utilizar el `OpenCSVSerDe`. Para obtener más información acerca de `SerDe` de `OpenCSV`, consulte [OpenCSVSerDe para el procesamiento de CSV](#).

## Para editar propiedades de tabla en la consola de AWS Glue

1. En el panel de navegación de la consola de AWS Glue, elija Tables.
2. Elija el enlace para la tabla que quiere editar y, a continuación, elija Actions (Acciones), Edit table details (Editar detalles de la tabla).
3. En la página Editar tabla, realice los siguientes cambios:
  - En Serialization lib (Biblioteca de serialización), ingrese `org.apache.hadoop.hive.serde2.OpenCSVSerde`.
  - Para Patrones de Serde, ingrese los siguientes valores para las claves `escapeChar`, `quoteChar` y `separatorChar`:
    - Para `escapeChar`, ingrese una barra diagonal inversa (`\`).
    - Para `quoteChar`, ingrese una comilla doble (`"`).
    - Para `separatorChar`, ingrese una coma (`,`).
4. Seleccione Guardar.

Para obtener más información, consulte [Visualización y edición de los detalles de una tabla](#) en la Guía para desarrolladores de AWS Glue.

## Actualización de propiedades de tablas de AWS Glue mediante programación

Puede utilizar la operación de la API [UpdateTable](#) de AWS Glue o el comando de la CLI [update-table](#) para modificar el bloque `SerDeInfo` en la definición de tabla, como en el siguiente ejemplo JSON.

```
"SerDeInfo": {
  "name": "",
  "serializationLib": "org.apache.hadoop.hive.serde2.OpenCSVSerde",
  "parameters": {
    "separatorChar": ",",
    "quoteChar": "\""
    "escapeChar": "\\"
  }
},
```

## Archivos CSV con encabezados

Cuando define una tabla en Athena con una instrucción `CREATE TABLE`, puede utilizar la propiedad de tabla `skip.header.line.count` para ignorar los encabezados en los datos CSV, como en el siguiente ejemplo.

```
...  
STORED AS TEXTFILE  
LOCATION 's3://DOC-EXAMPLE-BUCKET/csvdata_folder/';  
TBLPROPERTIES ("skip.header.line.count"="1")
```

También puede eliminar los encabezados de CSV de antemano para que la información de encabezado no se incluya en los resultados de la consulta de Athena. Una forma de hacerlo es utilizar trabajos de AWS Glue, que se encargan de las tareas de extracción, transformación y carga (ETL). Puede escribir scripts en AWS Glue en un lenguaje que es una extensión del dialecto PySpark Python. Para obtener más información, consulte la sección sobre [creación de trabajos en GlueAWS](#) en la Guía para desarrolladores de AWS Glue.

En el siguiente ejemplo se muestra una función de un script de AWS Glue que escribe un marco dinámico utilizando `from_options` y establece la opción de formato `writeHeader` como `"false"`, lo que elimina la información de encabezado:

```
glueContext.write_dynamic_frame.from_options(frame = applymapping1, connection_type =  
"s3", connection_options = {"path": "s3://DOC-EXAMPLE-BUCKET/MYTABLEDATA/"}, format =  
"csv", format_options = {"writeHeader": False}, transformation_ctx = "datasink2")
```

## Indexación y filtrado de particiones de AWS Glue

Cuando Athena consulta las tablas divididas, recupera y filtra las particiones de tabla disponibles en el subconjunto relevante para la consulta. A medida que se agregan nuevos datos y particiones, se necesita más tiempo para procesar las particiones y el tiempo de ejecución de consulta puede aumentar. Si tiene una tabla con un gran número de particiones que crece con el tiempo, considere la posibilidad de utilizar indexación y filtrado de particiones de AWS Glue. La indexación de particiones permite a Athena optimizar el procesamiento de particiones y mejorar el rendimiento de las consultas en tablas altamente particionadas. La configuración del filtrado de particiones en las propiedades de una tabla es un proceso de dos pasos:

1. Creación de un índice de particiones en AWS Glue.
2. Habilitación del filtrado de particiones para la tabla.

## Creación de un índice de particiones

A fin de conocer los pasos para crear un índice de particiones en AWS Glue, consulte [Trabajar con índices de partición](#) en la Guía para desarrolladores de AWS Glue. Para conocer las limitaciones de los índices de particiones en AWS Glue, consulte [Acerca de los índices de particiones](#) de esa página.

## Habilitación del filtrado de particiones

A fin de habilitar el filtrado de particiones para la tabla, debe configurar una nueva propiedad de tabla en AWS Glue. Para conocer los pasos sobre cómo configurar las propiedades de la tabla en AWS Glue, consulte la página [Configuración de proyección de particiones](#). Cuando edita los detalles de la tabla en AWS Glue, agregue el siguiente valor de clave a la sección Table properties (Propiedades de la tabla):

- En Key (Clave), agregue `partition_filtering.enabled`.
- En Value (Valor), agregue `true`.

Para desactivar el filtrado de particiones en esta tabla en cualquier momento, configure el valor `partition_filtering.enabled` como `false`.

Después de completar los pasos anteriores, puede volver a la consola de Athena para consultar los datos.

Para obtener más información sobre la creación de índices y filtros de particiones, consulte el artículo [Mejora del rendimiento de consultas de Amazon Athena con índices de particiones de AWS Glue Data Catalog](#) en el Blog de macrodatos de AWS.

## Uso de datos geoespaciales

AWS Glue no admite de forma nativa Well-known Text (WKT), Well-Known Binary (WKB) u otros tipos de datos PostGIS. El clasificador AWS Glue analiza los datos geoespaciales y los clasifica utilizando los tipos de datos admitidos para el formato, como `varchar` para CSV. Al igual que ocurre con otras tablas de AWS Glue, es posible que tenga que actualizar las propiedades de las tablas creadas a partir de datos geoespaciales para permitir a Athena analizar estos tipos de datos tal y como están. Para obtener más información, consulte [Uso de rastreadores AWS Glue](#) y [Trabajo con archivos CSV](#). Es posible que Athena no pueda analizar algunos tipos de datos geoespaciales en las tablas de AWS Glue tal y como están. Para obtener más información acerca de cómo trabajar con datos geoespaciales en Athena, consulte [Consulta de datos geoespaciales](#).

## Uso de AWS Glue para realizar trabajos de ETL con Athena

Los trabajos de AWS Glue realizan operaciones de ETL. Un trabajo de AWS Glue ejecuta un script que extrae datos de las fuentes, los transforma y los carga en los destinos. Para obtener más información, consulte la sección sobre [creación de trabajos en GlueAWS](#) en la Guía para desarrolladores de AWS Glue.

### Creación de tablas con Athena para los trabajos de ETL de AWS Glue

Es necesario agregar a las tablas que cree en Athena una propiedad denominada `classification`, que identifica el formato de los datos. Esto permite a AWS Glue utilizar las tablas para trabajos de ETL. Los valores de clasificación pueden ser `avro`, `csv`, `json`, `orc`, `parquet` o `xml`. A continuación, se muestra un ejemplo de instrucción `CREATE TABLE` en Athena:

```
CREATE EXTERNAL TABLE sampleTable (  
  column1 INT,  
  column2 INT  
) STORED AS PARQUET  
TBLPROPERTIES (  
  'classification'='parquet')
```

Si no se añadió la propiedad al crear la tabla, puede hacerse ahora por medio de la consola de AWS Glue.

Para agregar la propiedad de clasificación con la consola de AWS Glue

1. Inicie sesión en la AWS Management Console y abra la consola de AWS Glue en <https://console.aws.amazon.com/glue/>.
2. En el panel de navegación de la consola, elija Tables (Tablas).
3. Elija el enlace para la tabla que quiere editar y, a continuación, elija Actions (Acciones), Edit table details (Editar detalles de la tabla).
4. Desplácese hacia abajo hasta la sección Table properties (Propiedades de la tabla).
5. Elija Añadir.
6. En Clave, escriba **classification**.
7. En Value (Valor), especifique un tipo de datos (por ejemplo, **json**).
8. Seleccione Guardar.

En la sección Table details (Detalles de la tabla), el tipo de datos que ha especificado aparecerá en el campo Classification (Clasificación) de la tabla.

Para obtener más información, consulte [Uso de tablas](#) en la Guía para desarrolladores de AWS Glue.

## Uso de trabajos de ETL para optimizar el rendimiento de las consultas

Los trabajos de AWS Glue son útiles para transformar los datos a un formato que optimice el rendimiento de las consultas en Athena. Los formatos de datos tienen un gran impacto en el rendimiento y el costo de las consultas en Athena.

Recomendamos el uso de los formatos Parquet y ORC. AWS Glue admite la escritura en estos dos formatos de datos, lo que puede facilitar y agilizar la tarea de transformar los datos en un formato óptimo para Athena. Para obtener más información sobre estos formatos y otras formas de mejorar el rendimiento, consulte los [10 mejores consejos para ajustar el rendimiento de Amazon Athena](#).

## Conversión de los tipos de datos SMALLINT y TINYINT en INT cuando realiza conversiones a ORC

Para reducir la probabilidad de que Athena no pueda leer los tipos de datos SMALLINT y TINYINT generados por un trabajo de ETL de AWS Glue, convierta SMALLINT y TINYINT a INT al utilizar el asistente o escribir un script para un trabajo de ETL.

## Automatización de trabajos de ETL de AWS Glue

Puede configurar los trabajos de ETL de AWS Glue para que se ejecuten automáticamente basándose en desencadenadores. Esta característica es ideal cuando se insertan datos del exterior de AWS en un bucket de Amazon S3, en un formato ineficiente para realizar consultas en Athena. Para obtener más información, consulte [Desencadenar trabajos de AWS Glue](#) en la Guía para desarrolladores de AWS Glue.

## Uso de la AWS CLI para recrear una base de datos de AWS Glue y sus tablas

No es posible cambiar el nombre de una base de datos de AWS Glue directamente, pero puede copiar su definición, modificarla y utilizarla para volver a crear la base de datos con un nombre diferente. Del mismo modo, puede copiar las definiciones de las tablas en la base de datos antigua, modificarlas y utilizarlas para volver a crear las tablas en la base de datos nueva.

### Note

Con el método presentado no se copian las particiones de las tablas.



En el siguiente procedimiento para Windows se supone que la AWS CLI está configurada para la salida JSON. Para cambiar el formato de salida predeterminado en la AWS CLI, ejecute `aws configure`.

Para copiar una base de datos de AWS Glue mediante la AWS CLI

1. En el símbolo del sistema, ejecute el siguiente comando de la AWS CLI para recuperar la definición de la base de datos de AWS Glue que desea copiar.

```
aws glue get-database --name database_name
```

Para obtener más información acerca del comando `get-database`, consulte [get-database](#).

2. Guarde la salida JSON en un archivo con el nombre de la nueva base de datos (por ejemplo, *new\_database\_name.json*) en su escritorio.
3. Abra el archivo *new\_database\_name.json* en un editor de texto.
4. Siga los siguientes pasos en el archivo JSON:
  - a. Elimina la entrada { "Database": exterior y el corchete de cierre correspondiente } al final del archivo.
  - b. Cambie la entrada Name por el nuevo nombre de la base de datos.
  - c. Quite el campo CatalogId.
5. Guarde el archivo.
6. En el símbolo del sistema, ejecute el siguiente comando de la AWS CLI para usar el archivo de definición de base de datos modificado a fin de crear la base de datos con el nombre nuevo.

```
aws glue create-database --database-input "file://~/Desktop\new_database_name.json"
```

Para obtener más información acerca del comando `create-database`, consulte [create-database](#). Para obtener información acerca de cómo cargar los parámetros de la AWS CLI desde un archivo, consulte [Carga de los parámetros de la AWS CLI desde un archivo](#) en la Guía del usuario de la AWS Command Line Interface.

7. Para comprobar que la base de datos nueva se ha creado en AWS Glue, ejecute el siguiente comando:

```
aws glue get-database --name new_database_name
```

Ahora está listo para obtener la definición de una tabla que desea copiar en la base de datos nueva, modificar la definición y usar la definición modificada para volver a crear la tabla en la base de datos nueva. Con este procedimiento no se cambia el nombre de la tabla.

Para copiar una tabla de AWS Glue mediante la AWS CLI

1. En el símbolo del sistema, ejecute el siguiente comando de la AWS CLI.

```
aws glue get-table --database-name database_name --name table_name
```

Para obtener más información acerca del comando `get-table`, consulte [get-table](#).

2. Guarde la salida JSON en un archivo con el nombre de la tabla (por ejemplo, *table\_name*.json) en el escritorio de Windows.
3. Abra el archivo en un editor de texto.
4. En el archivo JSON, elimine la entrada exterior `{"Table":` y el corchete de cierre correspondiente `}` al final del archivo.
5. En el archivo JSON, elimine las siguientes entradas y sus valores:
  - `DatabaseName`: esta entrada no es necesaria porque el comando de la CLI `create-table` usa el parámetro `--database-name`.
  - `CreateTime`
  - `UpdateTime`
  - `CreatedBy`
  - `IsRegisteredWithLakeFormation`
  - `CatalogId`
  - `VersionId`
6. Guarde el archivo de definición de la tabla.
7. En el símbolo del sistema, ejecute el siguiente comando de la AWS CLI para volver a crear la tabla en la base de datos nueva:

```
aws glue create-table --database-name new_database_name --table-input "file://~/Desktop\table_name.json"
```

Para obtener más información acerca del comando `create-table`, consulte [create-table](#).

La tabla aparece ahora en la base de datos nueva en AWS Glue y se puede consultar desde allí.

8. Repita los pasos para copiar cada tabla adicional a la base de datos nueva en AWS Glue.

## Uso del conector de datos de Athena para metaalmacén externo de Hive

Puede utilizar el conector de datos de Amazon Athena para metaalmacén externo de Hive para consultar conjuntos de datos en Amazon S3 que utilicen un metaalmacén de Apache Hive. No es necesaria la migración de los metadatos a AWS Glue Data Catalog. En la consola de administración de Athena, configure una función de Lambda para comunicarse con el metaalmacén de Hive en su VPC privada y, a continuación, conéctela al metaalmacén. La conexión desde Lambda a su metaalmacén de Hive está asegurada por un canal de Amazon VPC privado y no utiliza Internet público. Puede proporcionar su propio código de función de Lambda, o puede usar la implementación predeterminada del conector de datos de Athena para metaalmacén externo de Hive.

### Temas

- [Información general de las características](#)
- [Flujo de trabajo](#)
- [Consideraciones y limitaciones](#)
- [Conexión de Athena al almacén de metadatos de Apache Hive](#)
- [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos de Hive](#)
- [Conexión de Athena a un almacén de metadatos de Hive mediante un rol de ejecución de IAM existente](#)
- [Configuración de Athena para utilizar un conector de almacén de metadatos de Hive implementado](#)
- [Uso de un nombre de origen de datos predeterminado en consultas de almacenes de metadatos externos de Hive](#)
- [Trabajo con vistas de Hive](#)
- [Uso de la AWS CLI con almacenes de matadatos de Hive](#)
- [Implementación de referencia](#)

### Información general de las características

Con el conector de datos de Athena para metaalmacén externo de Hive, puede realizar las siguientes tareas:

- Utilice la consola de Athena para registrar catálogos personalizados y ejecutar consultas con ellos.
- Defina funciones de Lambda para diferentes metaalmacenes externos de Hive y únalas en consultas de Athena.
- Utilice el AWS Glue Data Catalog y sus metaalmacenes externos de Hive en la misma consulta de Athena.
- Especifique un catálogo en el contexto de ejecución de la consulta como el catálogo predeterminado actual. Esto elimina el requisito de prefijo de nombres de catálogo a nombres de base de datos en las consultas. En lugar de usar la sintaxis *catalog.database.table*, puede usar *database.table*.
- Utilice una variedad de herramientas para ejecutar consultas que hagan referencia a metaalmacenes externos de Hive. Puede utilizar la consola de Athena, la AWS CLI, el SDK de AWS, las API de Athena y los controladores JDBC y ODBC de Athena actualizados. Los controladores actualizados son compatibles con catálogos personalizados.

## Compatibilidad con API

Athena Data Connector para metaalmacenes externos de Hive incluye soporte para operaciones de API de registro de catálogos y operaciones de API de metadatos.

- Registro de catálogos: registre catálogos personalizados para metaalmacenes externos de Hive y [orígenes de datos federados](#).
- Metadatos: utilice las API de metadatos para proporcionar información de bases de datos y tablas para AWS Glue y cualquier catálogo que registre con Athena.
- Cliente Athena JAVA SDK: utilice las API de registro de catálogos, las API de metadatos y la compatibilidad con catálogos de la operación `StartQueryExecution` en el cliente Athena Java SDK actualizado.

## Implementación de referencia

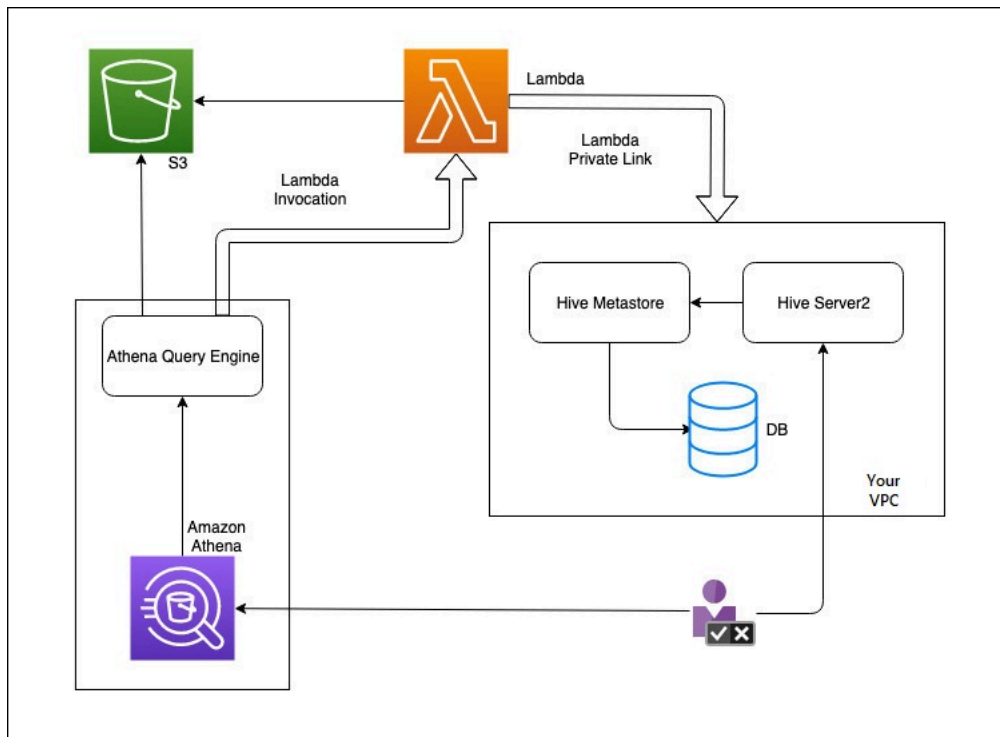
Athena proporciona una implementación de referencia para la función Lambda que se conecta a metaalmacenes externos de Hive. La implementación de referencia se proporciona en GitHub como un proyecto de código abierto en el [almacén de metadatos de Athena Hive](#).

La implementación de referencia está disponible como las dos aplicaciones de AWS SAM siguientes en el AWS Serverless Application Repository (SAR). Puede utilizar cualquiera de estas aplicaciones en el SAR para crear sus propias funciones de Lambda.

- **AthenaHiveMetastoreFunction**: archivo .jar de la función Uber Lambda. Un “uber” JAR (también conocido como fat JAR o JAR con dependencias) es un archivo .jar que contiene un programa Java y sus dependencias en un solo archivo.
- **AthenaHiveMetastoreFunctionWithLayer**: capa de Lambda y archivo .jar de función delgada de Lambda.

## Flujo de trabajo

En el siguiente diagrama, se muestra cómo interactúa Athena con su metaalmacén externo de Hive.



En este flujo de trabajo, el metaalmacén de Hive conectado a la base de datos está dentro de su VPC. Utilice Hive Server2 para administrar su metaalmacén de Hive mediante la CLI de Hive.

El flujo de trabajo para utilizar metaalmacenes externos de Hive de Athena incluye los siguientes pasos.

1. Cree una función Lambda que conecte Athena al metaalmacén de Hive que está dentro de su VPC.
2. Usted registra un nombre de catálogo único para su metaalmacén de Hive y un nombre de función correspondiente en su cuenta.

3. Cuando ejecuta una consulta DML o DDL de Athena que utiliza el nombre del catálogo, el motor de consulta de Athena llama al nombre de la función Lambda que asoció con el nombre del catálogo.
4. Con AWS PrivateLink, la función Lambda se comunica con el metaalmacén externo de Hive en su VPC y recibe respuestas a solicitudes de metadatos. Athena utiliza los metadatos del metaalmacén externo de Hive al igual que usa los metadatos del AWS Glue Data Catalog predeterminado.

## Consideraciones y limitaciones

Cuando utilice el conector de datos de Athena para metaalmacén externo de Hive, tenga en cuenta los siguientes puntos:

- Puede utilizar CTAS para crear una tabla en un metaalmacén externo de Hive.
- Puede utilizar INSERT INTO para insertar datos en un metaalmacén externo de Hive.
- El soporte de DDL para el metaalmacén externo de Hive está limitado a las siguientes instrucciones.
  - ALTER DATABASE SET DBPROPERTIES
  - ALTER TABLE ADD COLUMNS
  - ALTER TABLE ADD PARTITION
  - ALTER TABLE DROP PARTITION
  - ALTER TABLE RENAME PARTITION
  - ALTER TABLE REPLACE COLUMNS
  - ALTER TABLE SET LOCATION
  - ALTER TABLE SET TBLPROPERTIES
  - CREATE DATABASE
  - CREATE TABLE
  - CREATE TABLE AS
  - DESCRIBE TABLE
  - DROP DATABASE
  - DROP TABLE
  - SHOW COLUMNS
  - **SHOW CREATE TABLE**

- SHOW PARTITIONS
- MOSTRAR ESQUEMAS
- SHOW TABLES
- SHOW TBLPROPERTIES
- El número máximo de catálogos registrados que puede tener es de 1000.
- No se admite la autenticación Kerberos para el metaalmacén de Hive.
- Para utilizar el controlador JDBC con un metaalmacén externo de Hive o las [consultas federadas](#), incluya `MetadataRetrievalMethod=ProxyAPI` en la cadena de conexión JDBC. Para obtener información acerca del controlador JDBC, consulte [Conexión a Amazon Athena con JDBC](#).
- Las columnas ocultas de Hive `$path`, `$bucket`, `$file_size`, `$file_modified_time`, `$partition` y `$row_id` no se pueden utilizar para un filtrado de control de acceso detallado.
- Las tablas ocultas del sistema de Hive como `example_table$partitions` o `example_table$properties` no son compatibles con el control de acceso detallado.

## Permisos

Los conectores de datos prediseñados y personalizados pueden requerir acceso a los siguientes recursos para funcionar correctamente. Compruebe la información del conector que utiliza para asegurarse de que ha configurado correctamente la VPC. Para obtener información sobre los permisos de IAM necesarios para ejecutar consultas y crear un conector de origen de datos en Athena, consulte [Permitir el acceso a un conector de datos de Athena para un almacén de metadatos externo de Hive](#) y [Permitir a la función de Lambda el acceso a los almacenes de metadatos externos de Hive](#).

- Amazon S3: además de escribir los resultados de la consulta en la ubicación de resultados de la consulta de Athena en Amazon S3, los conectores de datos también escriben en un bucket de desbordamiento en Amazon S3. Se requiere conectividad y permisos para esta ubicación de Amazon S3. Para obtener más información, consulte [Ubicación de desbordamiento en Simple Storage Service \(Amazon S3\)](#) más adelante en este tema.
- Athena: se requiere acceso para verificar el estado de la consulta y evitar el sobreescaneo.
- AWS Glue: se requiere acceso si el conector utiliza AWS Glue para metadatos complementarios o principales.
- AWS Key Management Service

- Políticas: el metaalmacén de Hive, la Athena Query Federation y las UDF requieren políticas además de [Política administrada de AWS: AmazonAthenaFullAccess](#). Para obtener más información, consulte [Identity and Access Management en Athena](#).

## Ubicación de desbordamiento en Simple Storage Service (Amazon S3)

Debido al [límite](#) en los tamaños de respuesta de la función Lambda, las respuestas mayores que el umbral se desbordan en una ubicación de Amazon S3 que especifique al crear la función de Lambda. Athena lee estas respuestas de Amazon S3 directamente.

### Note

Athena no elimina los archivos de respuesta en Amazon S3. Se recomienda configurar una política de retención para eliminar automáticamente los archivos de respuesta.

## Conexión de Athena al almacén de metadatos de Apache Hive

Para conectar Athena al metaalmacén de Apache Hive, debe crear y configurar una función Lambda. Para una implementación básica, puede realizar todos los pasos necesarios comenzando desde la consola de administración de Athena.

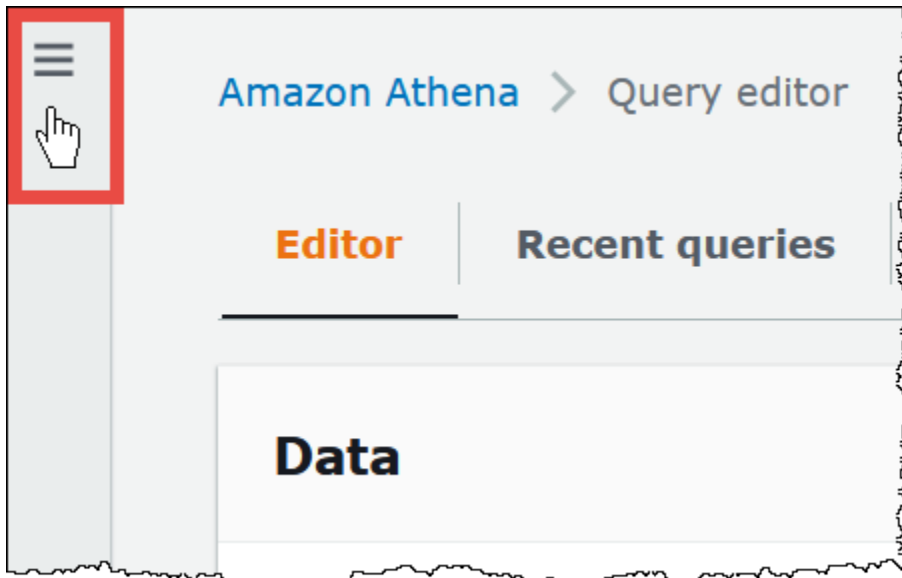
### Note

El siguiente procedimiento requiere tener permiso para crear un rol de IAM personalizado para la función Lambda. Si no tiene permiso para crear un rol personalizado, puede utilizar la [implementación de referencia](#) de Athena para crear una función Lambda por separado y, a continuación, utilizar la consola AWS Lambda para elegir un rol de IAM existente para la función. Para obtener más información, consulte [Conexión de Athena a un almacén de metadatos de Hive mediante un rol de ejecución de IAM existente](#).

Para conectar Athena a un metaalmacén de Hive

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.






3. Elija Edit Data Source (Editar origen de datos).
4. En la parte superior derecha de la consola, elija Create data source (Crear origen de datos).
5. En la página Choose data sources (Elegir orígenes de datos), en Data sources (Orígenes de datos), elija S3 - Apache Hive metastore (S3: metastore de Apache Hive).
6. Elija Siguiente.
7. En la sección Data source details (Detalles de origen de datos), en Data source name (Nombre de origen de datos), ingrese el nombre que quiera utilizar en las instrucciones SQL cuando consulte el origen de datos desde Athena. El nombre puede tener hasta 127 caracteres y debe ser único dentro de su cuenta. No se puede cambiar después de crearlo. Los caracteres válidos son a-z, A-Z, 0-9, \_ (guion bajo), @ (arroba) y - (guion). Los nombres awsdatalog, hive, jmx y system están reservados por Athena y no se pueden utilizar para nombres de orígenes de datos.
8. En Lambda function (Función Lambda), elija Create Lambda function (Crear una función Lambda) y, luego, elija Create a new Lambda function in AWS Lambda (Crear una nueva función Lambda en ).

La página AthenaHiveMetastoreFunction se abre en la consola de AWS Lambda. La página incluye información detallada sobre el conector.

Lambda > Functions > Create function > Review, configure and deploy

# AthenaHiveMetastoreFunction — version 1.0.1

Review, configure and deploy

 Copy as SAM Resource

## Application details

Author	Source code URL	Description	Report a vulnerability
default author	<a href="https://github.com/aws-labs/aws-athena-hive-metastore">https://github.com/aws-labs/aws-athena-hive-metastore</a>	An Athena Lambda function to interact with Hive Metastore	If you believe this application poses a security risk

## Readme file

Amazon Athena  
Hive Metastore  
Lambda Function

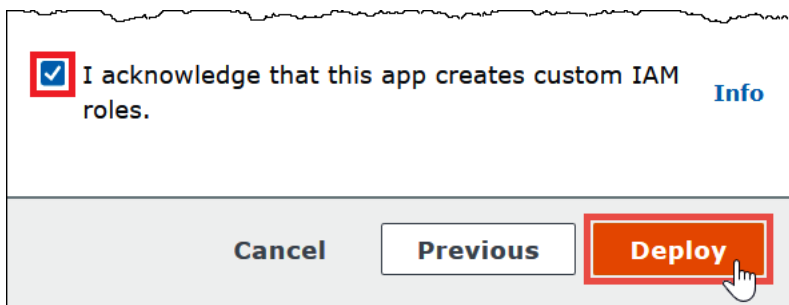
## Application settings

Application name  
The stack name of this application created via AWS CloudFormation

AthenaHiveMetastoreFunction

- En Application settings (Configuración de aplicación), ingrese los parámetros de la función Lambda.
  - LambdaFuncName: Proporcione un nombre para la función. Por ejemplo, myHiveMetastore.
  - SpillLocation: especifique una ubicación de Amazon S3 en esta cuenta para contener los metadatos de desbordamiento si el tamaño de la respuesta de la función de Lambda supera los 4 MB.

- HMSUri: ingrese el URI de su host del metaalmacén de Hive que utiliza el protocolo Thrift en el puerto 9083. Utilice la sintaxis `thrift://<host_name>:9083`.
  - LambdaMemory: especifique un valor comprendido entre 128 y 3008 MB. A la función Lambda se le asignan ciclos de CPU proporcionales a la cantidad de memoria que configure. El valor predeterminado es 1024.
  - LambdaTimeout: especifique el tiempo máximo permitido de ejecución de invocación Lambda en segundos de 1 a 900 (900 segundos es 15 minutos). El valor predeterminado es 300 segundos (5 minutos).
  - VPCSecurityGroupIds: ingrese una lista separada por comas de ID de grupo de seguridad de la VPC para el metaalmacén de Hive.
  - VPCSubnetIds: ingrese una lista separada por comas de ID de subred de la VPC para el metaalmacén de Hive.
10. Seleccione **I acknowledge that this app creates custom IAM roles** (Confirmando que esta aplicación puede crear roles de IAM personalizados) y, a continuación, elija **Deploy** (Implementar).



Cuando se completa la implementación, la función aparece en la lista de aplicaciones de Lambda. Ahora que se ha implementado la función del metaalmacén de Hive en la cuenta, puede configurar Athena para que la use.

11. Vuelva a la página **Enter data source details** (Ingresar detalles de orígenes de datos) de la consola de Athena.
12. En la sección **Lambda function** (Función Lambda), elija el icono de actualizar situado junto al cuadro de búsqueda de la función Lambda. Actualizar la lista de funciones disponibles hace que la función recién creada aparezca en la lista.
13. Elija el nombre de la función que acaba de crear en la consola de Lambda. Se muestra el ARN de la función Lambda.
14. (Opcional) En **Tags** (Etiquetas), agregue pares clave-valor que asociar con este origen de datos. Para obtener más información acerca de las etiquetas, consulte [Etiquetado de recursos de Athena](#).

15. Elija Siguiente.
16. En la página Review and create (Revisar y crear), revise los detalles del origen de datos y, a continuación, elija Create data source (Crear origen de datos).
17. La sección Data source details (Detalles del origen de datos) de la página de la fuente de datos muestra información sobre el nuevo conector.

Ahora puede usar el Data source name (Nombre de origen de datos) que especificó para referenciar el metastore de Hive en las consultas SQL en Athena. En las consultas SQL, utilice la sintaxis de ejemplo siguiente, reemplazando `hms-catalog-1` por el nombre de catálogo especificado anteriormente.

```
SELECT * FROM hms-catalog-1.CustomerData.customers
```

18. Para obtener información sobre la visualización, edición o eliminación de los orígenes de datos creados, consulte [Administración de orígenes de datos](#).

## Uso de AWS Serverless Application Repository para implementar un conector de origen de datos de Hive

Para implementar un conector de origen de datos de Athena para Hive, puede utilizar [AWS Serverless Application Repository](#) en lugar de empezar con la consola de Athena. Utilice AWS Serverless Application Repository para encontrar el conector que quiere utilizar, proporcione los parámetros que requiere el conector y, a continuación, implemente el conector en su cuenta. Luego, después de implementar el conector, utilice la consola de Athena para poner el origen de datos a disposición de Athena.

Para usar el AWS Serverless Application Repository para implementar un conector de origen de datos para Hive en su cuenta

1. Inicie sesión en la AWS Management Console y abra el repositorio de aplicaciones sin servidor.
2. En el panel de navegación, elija Aplicaciones disponibles.
3. Seleccione la opción Show apps that create custom IAM roles or resource policies (Mostrar aplicaciones que crean roles de IAM personalizados o políticas de recursos).
4. En el cuadro de búsqueda, escriba **Hive**. Los conectores que aparecen incluyen los dos siguientes:
  - AthenaHiveMetastoreFunction: archivo `.jar` de la función Uber Lambda.

- AthenaHiveMetastoreFunctionWithLayer: capa de Lambda y archivo .jar de función delgada de Lambda.

Las dos aplicaciones tienen la misma funcionalidad y difieren solo en su implementación. Puede utilizar cualquiera de ellas para crear una función Lambda que conecte Athena al metaalmacén de Hive.

5. Elija el nombre del conector que desea usar. En este tutorial se utiliza AthenaHiveMetastoreFunction.

The screenshot shows the Serverless Application Repository interface. On the left, there is a sidebar with the title "Serverless Application Repository" and two sections: "Available applications" (highlighted in orange) and "Published applications" (in blue). The main content area is titled "Available applications" and is split into "Public applications (1)" and "Private applications". A search bar contains the text "AthenaHiveMetastoreFunction". Below the search bar, there is a checked checkbox for "Show apps that create custom IAM roles or resource policies" and a "Sort by" dropdown menu set to "Best Match". At the bottom right of the search area, there are navigation arrows and the number "1". The application card for "AthenaHiveMetastoreFunction" is displayed, featuring a warning icon and the text "Creates custom IAM roles or resource policies". Below this, it says "An Athena Lambda function to interact with Hive Metastore". A blue button labeled "athena-hive-metastore" is visible. At the bottom of the card, it shows "default author" on the left and "5 deployments" on the right.

6. En Application settings (Configuración de aplicación), ingrese los parámetros de la función Lambda.

- LambdaFuncName: Proporcione un nombre para la función. Por ejemplo, myHiveMetastore.

- **SpillLocation:** especifique una ubicación de Amazon S3 en esta cuenta para contener los metadatos de desbordamiento si el tamaño de la respuesta de la función de Lambda supera los 4 MB.
  - **HMSUri:** ingrese el URI de su host del metaalmacén de Hive que utiliza el protocolo Thrift en el puerto 9083. Utilice la sintaxis `thrift://<host_name>:9083`.
  - **LambdaMemory:** especifique un valor comprendido entre 128 y 3008 MB. A la función Lambda se le asignan ciclos de CPU proporcionales a la cantidad de memoria que configure. El valor predeterminado es 1024.
  - **LambdaTimeout:** especifique el tiempo máximo permitido de ejecución de invocación Lambda en segundos de 1 a 900 (900 segundos es 15 minutos). El valor predeterminado es 300 segundos (5 minutos).
  - **VPCSecurityGroupIds:** ingrese una lista separada por comas de ID de grupo de seguridad de la VPC para el metaalmacén de Hive.
  - **VPCSubnetIds:** ingrese una lista separada por comas de ID de subred de la VPC para el metaalmacén de Hive.
7. En la parte inferior derecha de la página **Application details** (Detalles de la aplicación), seleccione **I acknowledge that this app creates custom IAM roles** (Confirmando que esta aplicación puede crear roles de IAM personalizados) y, a continuación, elija **Deploy** (Implementar).

En este punto, puede configurar Athena para que utilice la función Lambda para conectarse al metaalmacén de Hive. Para ver los pasos, consulte [Configuración de Athena para utilizar un conector de almacén de metadatos de Hive implementado](#).

## Conexión de Athena a un almacén de metadatos de Hive mediante un rol de ejecución de IAM existente

Para conectar el metaalmacén externo de Hive a Athena con una función Lambda que utiliza un rol de IAM existente, puede utilizar la implementación de referencia de Athena del conector Athena para metaalmacén externo de Hive.

Los tres pasos principales son los siguientes:

1. **Clonar y crear:** clone la implementación de referencia de Athena y cree el archivo JAR que contiene el código de función de Lambda.
2. **Consola AWS Lambda:** en la consola AWS Lambda, cree una función Lambda, asígnele un rol de ejecución de IAM existente y cargue el código de función que generó.

3. [Consola de Amazon Athena](#): en la consola de Amazon Athena, cree un nombre de origen de datos que pueda utilizar para hacer referencia al metaalmacén externo de Hive en las consultas de Athena.

Si ya tiene permisos para crear un rol de IAM personalizado, puede utilizar un flujo de trabajo más simple que use la consola de Athena y el AWS Serverless Application Repository para crear y configurar una función Lambda. Para obtener más información, consulte [Conexión de Athena al almacén de metadatos de Apache Hive](#).

#### Requisitos previos

- Git debe estar instalado en el sistema.
- Debe tener instalado [Apache Maven](#).
- Tiene un rol de ejecución de IAM que puede asignar a la función Lambda. Para obtener más información, consulte [Permitir a la función de Lambda el acceso a los almacenes de metadatos externos de Hive](#).

#### Clonar y crear la función de Lambda

El código de función para la implementación de referencia de Athena es un proyecto Maven ubicado en GitHub en [awslabs/aws-athena-hive-metastore](#). Para obtener información detallada sobre el proyecto, consulte el archivo README correspondiente en GitHub o el tema [Implementación de referencia](#) en esta documentación.

Para clonar y crear el código de la función Lambda

1. Ingrese el siguiente comando para clonar la implementación de referencia de Athena:

```
git clone https://github.com/awslabs/aws-athena-hive-metastore
```

2. Ejecute el siguiente comando para crear el archivo `.jar` para la función Lambda:

```
mvn clean install
```

Una vez creado el proyecto correctamente, se crea el siguiente archivo `.jar` en la carpeta de destino del proyecto:

```
hms-lambda-func-1.0-SNAPSHOT-withdep.jar
```

En la siguiente sección, utilizará la consola AWS Lambda para cargar este archivo en la cuenta de Amazon Web Services.

## Crear y configurar la función de Lambda en la consola de AWS Lambda

En esta sección, utilizará la consola de AWS Lambda para crear una función que utilice un rol de ejecución de IAM existente. Después de configurar una VPC para la función, cargue el código de función y configure las variables de entorno para la función.

### Crear la función de Lambda

En este paso, creará una función en la consola de AWS Lambda que utiliza un rol de IAM existente.

Para crear una función Lambda que utiliza un rol de IAM existente

1. Inicie sesión en la AWS Management Console y abra la consola AWS Lambda en <https://console.aws.amazon.com/lambda/>.
2. Seleccione Funciones en el panel de navegación.
3. Elija Crear función.
4. Elija Crear desde cero.
5. En Function name (Nombre de función), ingrese el nombre de la función Lambda (por ejemplo, **EHMSBasedLambda**).
6. En Runtime (Tiempo de ejecución), elija Java 8.
7. En Permissions (Permisos), expanda Change default execution role (Cambiar rol de ejecución predeterminado).
8. En Execution role (Rol de ejecución), elija Use an existing role (Usar un rol existente).
9. En Existing role (Rol existente), elija el rol de ejecución de IAM que utilizará la función Lambda para Athena (en este ejemplo, se utiliza un rol llamado AthenaLambdaExecutionRole).
10. Amplíe Advanced settings (Configuración avanzada).
11. Seleccione Enable Network (Habilitar red).
12. En VPC, elija la VPC a la que tendrá acceso la función.
13. En Subnets (Subredes), elija las subredes de la VPC que utilizará Lambda.
14. En Security groups (Grupos de seguridad), elija los grupos de seguridad de VPC que utilizará Lambda.



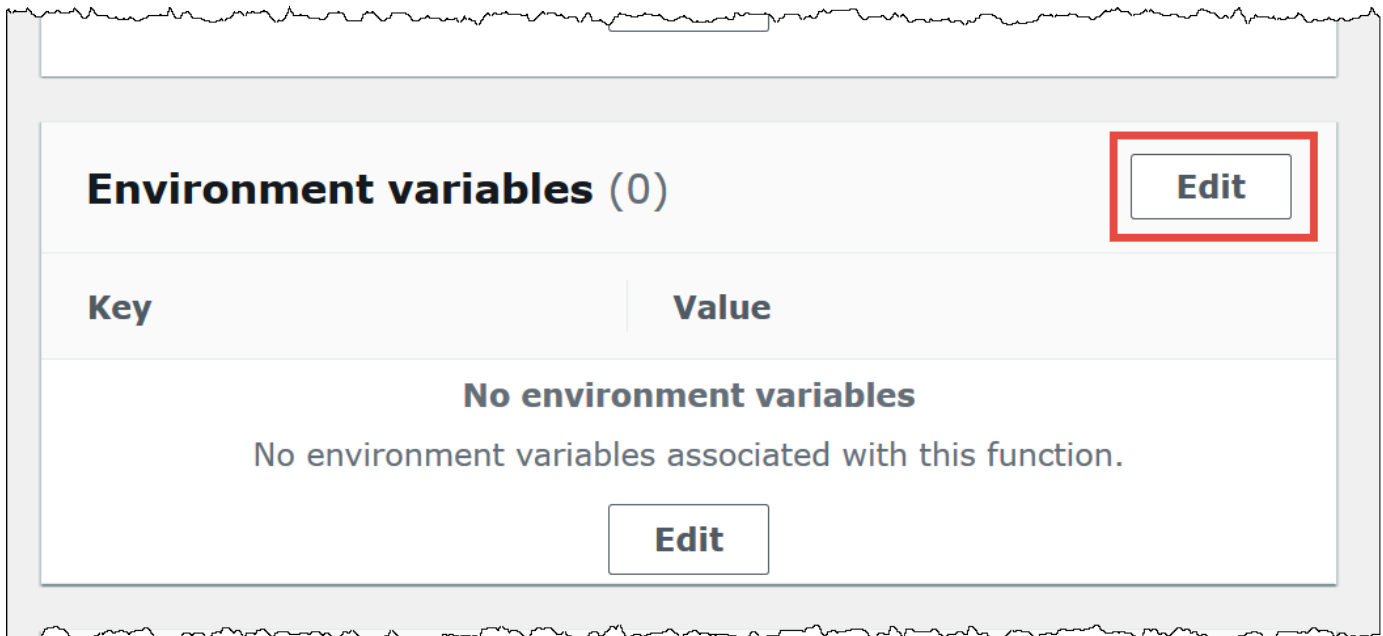
15. Elija Crear función. La consola de AWS Lambda abre la página de configuración de la función y comienza a crear la función.

### Cargue el código y configure la función de Lambda

Cuando la consola le informe de que se ha creado la función correctamente, estará listo para cargar el código de función y configurar sus variables de entorno.

Para cargar el código de la función Lambda y configurar sus variables de entorno

1. En la consola de Lambda, asegúrese de estar en la pestaña Code (Código) de la función que especificó.
2. En Code source (Código fuente), elija Upload from (Cargar desde) y, a continuación, .zip or .jar file (Archivo .zip o .jar).
3. Cargue el archivo `hms-lambda-func-1.0-SNAPSHOT-withdep.jar` que generó anteriormente.
4. En la página de la función de Lambda, seleccione la opción Configuración Pestaña.
5. En el panel de la izquierda, elija Environment variables (Variables de entorno).
6. En la sección Variables de entorno, elija Editar.



7. En la página Edit environment variables (Editar variables de entorno), use la opción Add environment variable (Agregar variable de entorno) para agregar los siguientes valores y claves de variable de entorno:

- HMS\_URIS: utilice la siguiente sintaxis para ingresar el URI de su host del metaalmacén de Hive que utiliza el protocolo Thrift en el puerto 9083.


```
thrift://<host_name>:9083
```

- SPILL\_LOCATION: especifique una ubicación de Amazon S3 en la cuenta de Amazon Web Services para contener los metadatos de desbordamiento si el tamaño de la respuesta de la función de Lambda supera los 4 MB.

Lambda > Functions > EHMSBasedLambda > Edit environment variables

## Edit environment variables

### Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#) 

Key	Value	
<input type="text" value="HMS_URIS"/>	<input type="text"/>	<input type="button" value="Remove"/>
<input type="text" value="SPILL_LOCATION"/>	<input type="text"/>	<input type="button" value="Remove"/>

► Encryption configuration

8. Elija Guardar.

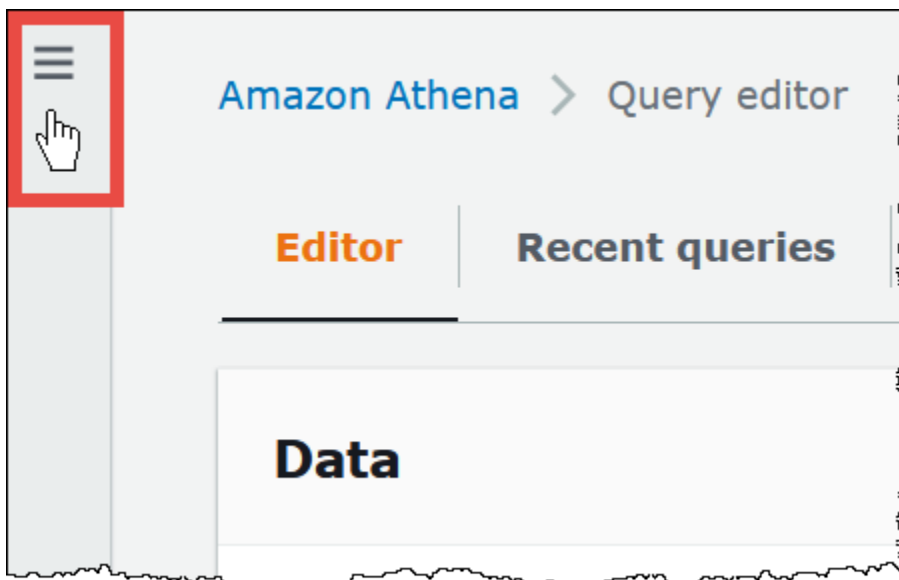
En este punto, lo tiene todo listo para configurar Athena para que utilice la función de Lambda para conectarse al metaalmacén de Hive. Para ver los pasos, consulte [Configuración de Athena para utilizar un conector de almacén de metadatos de Hive implementado](#).

## Configuración de Athena para utilizar un conector de almacén de metadatos de Hive implementado

Después de haber implementado un conector de origen de datos de Lambda como AthenaHiveMetastoreFunction en su cuenta, puede configurar Athena para utilizarlo. Para ello, cree un nombre de origen de datos que haga referencia al metaalmacén externo de Hive para utilizarlo en consultas de Athena.

Para conectar Athena al metaalmacén de Hive mediante una función de Lambda existente

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.



3. Elija Edit Data Source (Editar origen de datos).
4. En la página Data sources (Orígenes de datos), elija Create data source (Crear origen de datos).
5. En la página Choose data sources (Elegir orígenes de datos), en Data sources (Orígenes de datos), elija S3 - Apache Hive metastore (S3: metastore de Apache Hive).
6. Elija Siguiente.

7. En la sección Data source details (Detalles del origen de datos), en Data source name (Nombre del origen de datos), ingrese el nombre que quiera utilizar en las instrucciones SQL cuando consulte el origen de datos desde Athena (por ejemplo, MyHiveMetastore). El nombre puede tener hasta 127 caracteres y debe ser único dentro de su cuenta. No se puede cambiar después crearlo. Los caracteres válidos son a-z, A-Z, 0-9, \_ (guion bajo), @ (arroba) y - (guion). Los nombres awsdatabacatalog, hive, jmx y system están reservados por Athena y no se pueden utilizar para nombres de orígenes de datos.
8. En la sección Connection details (Detalles de la conexión), utilice el cuadro Select or enter a Lambda function (Seleccione o introduzca una función de Lambda) para elegir el nombre de la función que acaba de crear. Se muestra el ARN de la función Lambda.
9. (Opcional) En Tags (Etiquetas), agregue pares clave-valor que asociar con este origen de datos. Para obtener más información acerca de las etiquetas, consulte [Etiquetado de recursos de Athena](#).
10. Elija Siguiente.
11. En la página Review and create (Revisar y crear), revise los detalles del origen de datos y, a continuación, elija Create data source (Crear origen de datos).
12. La sección Data source details (Detalles del origen de datos) de la página de la fuente de datos muestra información sobre el nuevo conector.

Ahora puede usar el Data source name (Nombre de origen de datos) que especificó para referenciar el metastore de Hive en las consultas SQL en Athena.

En las consultas SQL, utilice la sintaxis de ejemplo siguiente, reemplazando ehms-catalog por el origen de datos especificado anteriormente.

```
SELECT * FROM ehms-catalog.CustomerData.customers
```

13. Para ver, editar o eliminar los orígenes de datos creados, consulte [Administración de orígenes de datos](#).

## Uso de un nombre de origen de datos predeterminado en consultas de almacenes de metadatos externos de Hive

Cuando ejecuta consultas DML y DDL en metaalmacenes externos de Hive, puede simplificar la sintaxis de la consulta omitiendo el nombre del catálogo si ese nombre está seleccionado en el editor de consultas. Se aplican determinadas restricciones a esta funcionalidad.

## instrucciones DML

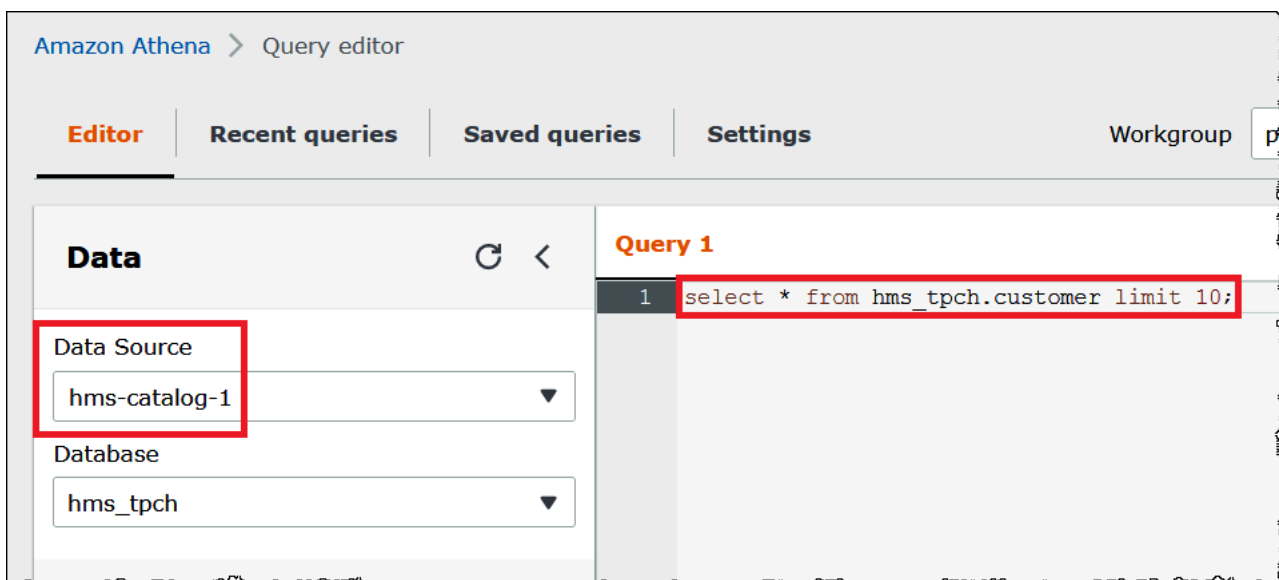
Para ejecutar consultas con catálogos registrados

1. Puede colocar el nombre del origen de datos antes de la base de datos mediante la sintaxis `[[data_source_name].database_name].table_name`, como en el ejemplo siguiente.

```
select * from "hms-catalog-1".hms_tpch.customer limit 10;
```

2. Cuando el origen de datos que desea utilizar ya está seleccionado en el editor de consultas, puede omitir el nombre de la consulta, como en el ejemplo siguiente.

```
select * from hms_tpch.customer limit 10:
```



3. Cuando utiliza varios orígenes de datos en una consulta, solo puede omitir el nombre del origen de datos predeterminado y debe especificar el nombre completo de cualquier origen de datos no predeterminado.

Por ejemplo, supongamos que seleccionó `AwsDataCatalog` como origen de datos predeterminado en el editor de consultas. La instrucción `FROM` del fragmento de consulta siguiente cumple totalmente los dos primeros nombres de origen de datos, pero omite el nombre del tercero porque se encuentra en el catálogo de datos AWS Glue.

```
...
FROM ehms01.hms_tpch.customer,
     "hms-catalog-1".hms_tpch.orders,
```

hms\_tpch.lineitem

...

## Instrucciones DDL

Las siguientes instrucciones DDL de Athena admiten prefijos de nombre de catálogo. Los prefijos de nombre de catálogo en otras instrucciones DDL provocan errores de sintaxis.

```
SHOW TABLES [IN [catalog_name.]database_name] ['regular_expression']

SHOW TBLPROPERTIES [[catalog_name.]database_name.]table_name [('property_name')]

SHOW COLUMNS IN [[catalog_name.]database_name.]table_name

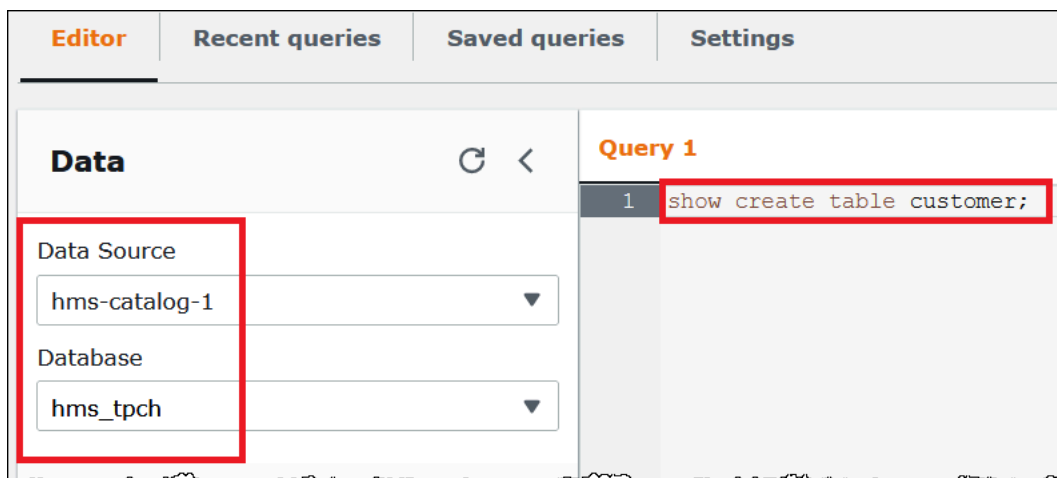
SHOW PARTITIONS [[catalog_name.]database_name.]table_name

SHOW CREATE TABLE [[catalog_name.][database_name.]table_name

DESCRIBE [EXTENDED | FORMATTED] [[catalog_name.][database_name.]table_name [PARTITION
partition_spec] [col_name ( [.field_name] | [.'$elem$'] | [.'$key$'] | [.'$value$'] )]
```

Al igual que con las instrucciones DML, puede omitir los prefijos de origen de datos y base de datos de la consulta cuando el origen de datos y la base de datos están seleccionados en el editor de consultas.

En el ejemplo siguiente, el origen de datos hms-catalog-1 y la base de datos hms\_tpch están seleccionados en el editor de consultas. La instrucción `show create table customer` se ejecuta correctamente aunque se omiten el prefijo hms-catalog-1 y el nombre de la base de datos hms\_tpch en la consulta.



## Especificación de un origen de datos predeterminado en una cadena de conexión JDBC

Cuando utiliza el controlador JDBC de Athena para conectar Athena a un almacén de metadatos externo de Hive, puede utilizar el parámetro `Catalog` para especificar el nombre del origen de datos predeterminado en la cadena de conexión en un editor SQL, como [SQL Workbench](#).

### Note

Para descargar los controladores JDBC de Athena más recientes, consulte [Uso de Athena con el controlador JDBC](#).

La siguiente cadena de conexión especifica el origen de datos predeterminado *hms-catalog-name*.

```
jdbc:awsathena://AwsRegion=us-east-1;S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/lambda/results/;Workgroup=AmazonAthenaPreviewFunctionality;Catalog=hms-catalog-name;
```

En la imagen siguiente se muestra un ejemplo de URL de conexión JDBC según se ha configurado en SQL Workbench.

## Trabajo con vistas de Hive

Puede utilizar Athena para consultar las vistas existentes en los almacenes de metadatos de Apache Hive externos. Athena traduce las vistas sobre la marcha en tiempo de ejecución sin cambiar la vista original ni almacenar la traducción.

Suponga, por ejemplo, que tiene una vista de Hive como la siguiente que utiliza una sintaxis no admitida en Athena como `LATERAL VIEW explode()`:

```
CREATE VIEW team_view AS
SELECT team, score
FROM matches
LATERAL VIEW explode(scores) m AS score
```

Athena traduce la cadena de consulta de vista de Hive en una instrucción como la siguiente que Athena puede ejecutar:



```
SELECT team, score
FROM matches
CROSS JOIN UNNEST(scores) AS m (score)
```

Para obtener información sobre cómo conectar un almacén de metadatos de Hive externo a Athena, consulte [Uso del conector de datos de Athena para metaalmacén externo de Hive](#).

## Consideraciones y limitaciones

Cuando consulte las vistas de Hive desde Athena, tenga en cuenta los siguientes puntos:

- Athena no admite la creación de vistas de Hive. Puede crear vistas de Hive en su almacén de metadatos de Hive externo, que luego puede consultar desde Athena.
- Athena no admite las UDF personalizadas para las vistas de Hive.
- Debido a un problema conocido en la consola de Athena, las vistas de Hive aparecen debajo de la lista de tablas en lugar de la lista de vistas.
- Aunque el proceso de traducción es automático, algunas funciones de Hive no son compatibles con las vistas de Hive ni requieren un manejo especial. Para más información, consulte la sección a continuación.

## Limitaciones de compatibilidad de las funciones de Hive

En esta sección, se destacan las funciones de Hive que Athena no admite para las vistas de Hive o que requieren un tratamiento especial. Actualmente, dado que Athena admite principalmente funciones de Hive 2.2.0, las funciones que están disponibles solo en versiones superiores (como Hive 4.0.0) no están disponibles. Para obtener una lista completa de las funciones de Hive, consulte el [Hive language manual UDF](#) (UDF manual del lenguaje Hive).

## Funciones de agregación

### Funciones de agregación que requieren un manejo especial

La siguiente función de agregación para las vistas de Hive requiere un manejo especial.

- Avg: en lugar de `avg(INT i)`, utilice `avg(CAST(i AS DOUBLE))`.

### Funciones de agregación no admitidas

Las siguientes funciones de agregación de Hive no se admiten en Athena para las vistas de Hive.

```
covar_pop
histogram_numeric
ntile
percentile
percentile_approx
```

Las funciones de regresión como `regr_count`, `regr_r2` y `regr_sxx` no se admiten en las vistas de Athena para Hive.

### Funciones de fecha no admitidas

Las siguientes funciones de fecha de Hive no se admiten en Athena para las vistas de Hive.

```
date_format(date/timestamp/string ts, string fmt)
day(string date)
dayofmonth(date)
extract(field FROM source)
hour(string date)
minute(string date)
month(string date)
quarter(date/timestamp/string)
second(string date)
weekofyear(string date)
year(string date)
```

### Funciones de enmascaramiento no admitidas

Las funciones de enmascaramiento de Hive, como `mask()` y `mask_first_n()`, no se admiten en Athena para las vistas de Hive.

### Funciones varias

#### Funciones varias que requieren un manejo especial

Las siguientes funciones varias para las vistas de Hive requieren un manejo especial.

- `md5`: Athena admite `md5(binary)`, pero no `md5(varchar)`.
- `Explode`: Athena admite `explode` cuando se utiliza en la siguiente sintaxis:

```
LATERAL VIEW [OUTER] EXPLode(<argument>)
```

- **Posexplode:** Athena admite `posexplode` cuando se utiliza en la siguiente sintaxis:

```
LATERAL VIEW [OUTER] POSEXPLODE(<argument>)
```

En la salida (`pos`, `val`), Athena trata la columna `pos` como `BIGINT`. Debido a esto, es posible que necesite convertir la columna `pos` en `BIGINT` para evitar una vista obsoleta. En el siguiente ejemplo, se ilustra esta técnica.

```
SELECT CAST(c AS BIGINT) AS c_bigint, d  
FROM table LATERAL VIEW POSEXPLODE(<argument>) t AS c, d
```

## Funciones varias no admitidas

Las siguientes funciones de Hive no se admiten en Athena para las vistas de Hive.

```
aes_decrypt  
aes_encrypt  
current_database  
current_user  
inline  
java_method  
logged_in_user  
reflect  
sha/sha1/sha2  
stack  
version
```

## Operadores

### Operadores que requieren un manejo especial

Los siguientes operadores para las vistas de Hive requieren un manejo especial.

- **Operador mod (%):** debido a que el tipo `DOUBLE` se convierte implícitamente a `DECIMAL(x, y)`, la siguiente sintaxis puede provocar el mensaje de error: La vista es obsoleta

```
a_double % 1.0 AS column
```

Para solucionar este problema, utilice `CAST`, como en el siguiente ejemplo.

```
CAST(a_double % 1.0 as DOUBLE) AS column
```

- Operador de división (/): en Hive, `int` dividido por `int` produce un `double`. En Athena, la misma operación produce un `int` truncado.

## Operadores no admitidos

Athena no admite los siguientes operadores para vistas de Hive.

`~A`: bit a bit NOT

`A ^ b`: bit a bit XOR

`A & b`: bit a bit AND

`A | b`: bit a bit OR

`A <=> b`: devuelve el mismo resultado que los operadores iguales a (`=`) para operandos no nulos. Devuelve `TRUE` si ambos son `NULL`, `FALSE` si uno de ellos es `NULL`.

## Funciones de cadena

### Funciones de cadena que requieren un manejo especial

Las siguientes funciones de cadena de Hive para las vistas de Hive requieren un manejo especial.

- `chr(bigint|double a)`: Hive permite argumentos negativos; Athena no lo hace.
- `instr(string str, string substr)`: debido a que la asignación de Athena para la función `instr` devuelve `BIGINT` en lugar de `INT`, utilice la siguiente sintaxis:

```
CAST(instr(string str, string substr) as INT)
```

Sin este paso, la vista se considerará obsoleta.

- `length(string a)`: debido a que la asignación de Athena para la función `length` devuelve `BIGINT` en lugar de `INT`, utilice la siguiente sintaxis para que no se considere la vista obsoleta:

```
CAST(length(string str) as INT)
```

## Funciones de cadena no admitidas

Las siguientes funciones de cadena de Hive no se admiten en Athena para las vistas de Hive.

```
ascii(string str)
character_length(string str)
decode(binary bin, string charset)
encode(string src, string charset)
elt(N int, str1 string, str2 string, str3 string, ...)
field(val T, val1 T, val2 T, val3 T, ...)
find_in_set(string str, string strList)
initcap(string A)
levenshtein(string A, string B)
locate(string substr, string str[, int pos])
octet_length(string str)
parse_url(string urlString, string partToExtract [, string keyToExtract])
printf(String format, Obj... args)
quote(String text)
regexp_extract(string subject, string pattern, int index)
repeat(string str, int n)
sentences(string str, string lang, string locale)
soundex(string A)
space(int n)
str_to_map(text[, delimiter1, delimiter2])
substring_index(string A, string delim, int count)
```

## Funciones de XPath no admitidas

Las funciones XPath de Hive, como `xpath`, `xpath_short` y `xpath_int` no se admiten en Athena para las vistas de Hive.

## Solución de problemas

Cuando utilice las vistas de Hive en Athena, es posible que se encuentre con los siguientes problemas:

- La vista **<view name>** está obsoleta: este mensaje suele indicar una discordancia de tipo entre la vista en Hive y Athena. Si la misma función en el [Hive LanguageManual UDF](#) y la documentación de las [funciones y operadores Presto](#) tiene firmas diferentes, pruebe convertir el tipo de datos que no coincida.
- Función no registrada: Athena no admite la función actualmente. Para obtener más información, consulte los datos mencionados anteriormente en este documento.

## Uso de la AWS CLI con almacenes de metadatos de Hive

Puede utilizar comandos de la CLI de `aws athena` para administrar los catálogos de datos de metaalmacén de Hive que utilice con Athena. Después de definir uno o más catálogos para utilizarlos con Athena, puede hacer referencia a esos catálogos en los comandos DDL y DML de `aws athena`.

### Uso de la AWS CLI para administrar catálogos de almacenes de metadatos de Hive

#### Registro de un catálogo: `create-data-catalog`

Para registrar un catálogo de datos, utilice el comando `create-data-catalog`. Utilice el parámetro `name` para especificar el nombre que desea utilizar para el catálogo. Pase el ARN de la función de Lambda a la opción `metadata-function` del argumento `parameters`. Para crear etiquetas para el nuevo catálogo, utilice el parámetro `tags` con uno o más pares de argumentos `Key=key, Value=value` separados por espacios.

En el siguiente ejemplo se registra el catálogo de metaalmacén de Hive denominado `hms-catalog-1`. El comando se ha formateado para fines de legibilidad.

```
$ aws athena create-data-catalog
--name "hms-catalog-1"
--type "HIVE"
--description "Hive Catalog 1"
--parameters "metadata-function=arn:aws:lambda:us-
east-1:111122223333:function:external-hms-service-v3, sdk-version=1.0"
--tags Key=MyKey, Value=MyValue
--region us-east-1
```

#### Visualización de los detalles del catálogo: `get-data-catalog`

Para mostrar los detalles de un catálogo, pase el nombre del catálogo al comando `get-data-catalog`, como en el ejemplo siguiente.

```
$ aws athena get-data-catalog --name "hms-catalog-1" --region us-east-1
```

El siguiente resultado de ejemplo está en formato JSON.

```
{
  "DataCatalog": {
    "Name": "hms-catalog-1",
```

```
    "Description": "Hive Catalog 1",
    "Type": "HIVE",
    "Parameters": {
      "metadata-function": "arn:aws:lambda:us-
east-1:111122223333:function:external-hms-service-v3",
      "sdk-version": "1.0"
    }
  }
}
```

### Enumeración de los catálogos registrados: list-data-catalogs

Para enumerar los catálogos registrados, utilice el comando `list-data-catalogs` y, opcionalmente, especifique una región, como en el ejemplo siguiente. Los catálogos enumerados siempre incluyen AWS Glue.

```
$ aws athena list-data-catalogs --region us-east-1
```

El siguiente resultado de ejemplo está en formato JSON.

```
{
  "DataCatalogs": [
    {
      "CatalogName": "AwsDataCatalog",
      "Type": "GLUE"
    },
    {
      "CatalogName": "hms-catalog-1",
      "Type": "HIVE",
      "Parameters": {
        "metadata-function": "arn:aws:lambda:us-
east-1:111122223333:function:external-hms-service-v3",
        "sdk-version": "1.0"
      }
    }
  ]
}
```

### Actualización de un catálogo: update-data-catalog

Para actualizar un catálogo de datos, utilice el comando `update-data-catalog`, como en el ejemplo siguiente. El comando se ha formateado para fines de legibilidad.

```
$ aws athena update-data-catalog
--name "hms-catalog-1"
--type "HIVE"
--description "My New Hive Catalog Description"
--parameters "metadata-function=arn:aws:lambda:us-
east-1:111122223333:function:external-hms-service-new, sdk-version=1.0"
--region us-east-1
```

### Eliminación de un catálogo: delete-data-catalog

Para eliminar un catálogo de datos, utilice el comando `delete-data-catalog`, como en el ejemplo siguiente.

```
$ aws athena delete-data-catalog --name "hms-catalog-1" --region us-east-1
```

### Visualización de los detalles de la base de datos: get-database

Para mostrar los detalles de una base de datos, pase el nombre del catálogo y la base de datos al comando `get-database`, como en el ejemplo siguiente.

```
$ aws athena get-database --catalog-name hms-catalog-1 --database-name mydb
```

El siguiente resultado de ejemplo está en formato JSON.

```
{
  "Database": {
    "Name": "mydb",
    "Description": "My database",
    "Parameters": {
      "CreatedBy": "Athena",
      "EXTERNAL": "TRUE"
    }
  }
}
```

### Enumeración de las bases de datos de un catálogo: list-databases

Para enumerar las bases de datos de un catálogo, utilice el comando `list-databases` y, como opción, especifique una región, como en el ejemplo siguiente.



```
$ aws athena list-databases --catalog-name AwsDataCatalog --region us-west-2
```

El siguiente resultado de ejemplo está en formato JSON.

```
{
  "DatabaseList": [
    {
      "Name": "default"
    },
    {
      "Name": "mycrawlerdatabase"
    },
    {
      "Name": "mydatabase"
    },
    {
      "Name": "sampledb",
      "Description": "Sample database",
      "Parameters": {
        "CreatedBy": "Athena",
        "EXTERNAL": "TRUE"
      }
    },
    {
      "Name": "tpch100"
    }
  ]
}
```

Visualización de los detalles de la tabla: `get-table-metadata`

Para mostrar los metadatos de una tabla, incluidos los nombres de columna y los tipos de datos, pase el nombre del catálogo, la base de datos y el nombre de tabla al comando `get-table-metadata`, como en el ejemplo siguiente.

```
$ aws athena get-table-metadata --catalog-name AwsDataCatalog --database-name mydb --
table-name cityuseragent
```

El siguiente resultado de ejemplo está en formato JSON.

```
{
```

```

"TableMetadata": {
  "Name": "cityuseragent",
  "CreateTime": 1586451276.0,
  "LastAccessTime": 0.0,
  "TableType": "EXTERNAL_TABLE",
  "Columns": [
    {
      "Name": "city",
      "Type": "string"
    },
    {
      "Name": "useragent1",
      "Type": "string"
    }
  ],
  "PartitionKeys": [],
  "Parameters": {
    "COLUMN_STATS_ACCURATE": "false",
    "EXTERNAL": "TRUE",
    "inputformat": "org.apache.hadoop.mapred.TextInputFormat",
    "last_modified_by": "hadoop",
    "last_modified_time": "1586454879",
    "location": "s3://DOC-EXAMPLE-BUCKET/",
    "numFiles": "1",
    "numRows": "-1",
    "outputformat":
"org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat",
    "rawDataSize": "-1",
    "serde.param.serialization.format": "1",
    "serde.serialization.lib":
"org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
    "totalSize": "61"
  }
}
}

```

Visualización de los metadatos de todas las tablas de una base de datos: `list-table-metadata`

Para mostrar los metadatos de todas las tablas de una base de datos, pase el nombre del catálogo y el nombre de la base de datos al comando `list-table-metadata`. El comando `list-table-metadata` es similar al comando `get-table-metadata`, excepto que no se especifica un nombre de tabla. Para limitar el número de resultados, puede utilizar la opción `--max-results`, como en el ejemplo siguiente.

```
$ aws athena list-table-metadata --catalog-name AwsDataCatalog --database-name sampledb
--region us-east-1 --max-results 2
```

El siguiente resultado de ejemplo está en formato JSON.

```
{
  "TableMetadataList": [
    {
      "Name": "cityuseragent",
      "CreateTime": 1586451276.0,
      "LastAccessTime": 0.0,
      "TableType": "EXTERNAL_TABLE",
      "Columns": [
        {
          "Name": "city",
          "Type": "string"
        },
        {
          "Name": "useragent1",
          "Type": "string"
        }
      ],
      "PartitionKeys": [],
      "Parameters": {
        "COLUMN_STATS_ACCURATE": "false",
        "EXTERNAL": "TRUE",
        "inputformat": "org.apache.hadoop.mapred.TextInputFormat",
        "last_modified_by": "hadoop",
        "last_modified_time": "1586454879",
        "location": "s3://DOC-EXAMPLE-BUCKET/",
        "numFiles": "1",
        "numRows": "-1",
        "outputformat":
"org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat",
        "rawDataSize": "-1",
        "serde.param.serialization.format": "1",
        "serde.serialization.lib":
"org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
        "totalSize": "61"
      }
    },
    {
      "Name": "clearinghouse_data",
```

```

    "CreateTime": 1589255544.0,
    "LastAccessTime": 0.0,
    "TableType": "EXTERNAL_TABLE",
    "Columns": [
      {
        "Name": "location",
        "Type": "string"
      },
      {
        "Name": "stock_count",
        "Type": "int"
      },
      {
        "Name": "quantity_shipped",
        "Type": "int"
      }
    ],
    "PartitionKeys": [],
    "Parameters": {
      "EXTERNAL": "TRUE",
      "inputformat": "org.apache.hadoop.mapred.TextInputFormat",
      "location": "s3://DOC-EXAMPLE-BUCKET/",
      "outputformat":
"org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat",
      "serde.param.serialization.format": "1",
      "serde.serialization.lib":
"org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
      "transient_lastDdlTime": "1589255544"
    }
  },
  "NextToken":
"eyJzYXN0RXZhbHVhdGVkS2V5Ijpw7IkhBU0hfs0VZIjpw7InMi0iJ0Ljk0YWZjYjk1MjJjNTQ1YmU4Y2I50WE5NTg0MjFjYjY"
}

```

## Ejecución de instrucciones DDL y DML

Cuando utiliza la AWS CLI para ejecutar instrucciones DDL y DML, puede pasar el nombre del catálogo de metaalmacén de Hive de dos maneras:

- Directamente en las instrucciones que son compatibles.
- Al parámetro de Catalog `--query-execution-context`.

## Instrucciones DDL

En el ejemplo siguiente se pasa el nombre del catálogo directamente como parte de la instrucción DDL `show create table`. El comando se ha formateado para fines de legibilidad.

```
$ aws athena start-query-execution
--query-string "show create table hms-catalog-1.hms_tpch_partitioned.lineitem"
--result-configuration "OutputLocation=s3://DOC-EXAMPLE-BUCKET/lambda/results"
```

La instrucción DDL `show create table` de ejemplo siguiente utiliza el parámetro `Catalog` de `--query-execution-context` para pasar el nombre del catálogo de metaalmacén de Hive `hms-catalog-1`. El comando se ha formateado para fines de legibilidad.

```
$ aws athena start-query-execution
--query-string "show create table lineitem"
--query-execution-context "Catalog=hms-catalog-1,Database=hms_tpch_partitioned"
--result-configuration "OutputLocation=s3://DOC-EXAMPLE-BUCKET/lambda/results"
```

## instrucciones DML

La siguiente instrucción DML `select` de ejemplo pasa directamente el nombre del catálogo a la consulta. El comando se ha formateado para fines de legibilidad.

```
$ aws athena start-query-execution
--query-string "select * from hms-catalog-1.hms_tpch_partitioned.customer limit 100"
--result-configuration "OutputLocation=s3://DOC-EXAMPLE-BUCKET/lambda/results"
```

La siguiente instrucción DML `select` de ejemplo utiliza el parámetro `Catalog` de `--query-execution-context` para pasar en el nombre del catálogo de metaalmacén de Hive `hms-catalog-1`. El comando se ha formateado para fines de legibilidad.


```
$ aws athena start-query-execution
--query-string "select * from customer limit 100"
--query-execution-context "Catalog=hms-catalog-1,Database=hms_tpch_partitioned"
--result-configuration "OutputLocation=s3://DOC-EXAMPLE-BUCKET/lambda/results"
```

## Implementación de referencia

Athena proporciona una implementación de referencia del conector para metaalmacén externo de Hive en GitHub.com en <https://github.com/aws-labs/aws-athena-hive-metastore>.

La implementación de referencia es un proyecto de [Apache Maven](#) que tiene los siguientes módulos:

- **hms-service-api**: contiene las operaciones de la API entre la función Lambda y los clientes del servicio Athena. Estas operaciones de API se definen en la interfaz `HiveMetaStoreService`. Debido a que este es un contrato de servicio, no debe cambiar nada en este módulo.
- **hms-lambda-handler**: un conjunto de controladores Lambda predeterminados que procesan todas las llamadas a la API del metaalmacén de Hive. La clase `MetadataHandler` es el distribuidor de todas las llamadas a la API. No es necesario cambiar este paquete.
- **hms-lambda-func**: una función de ejemplo de Lambda que tiene los siguientes componentes.
  - **HiveMetaStoreLambdaFunc**: un ejemplo de la función Lambda que amplía `MetadataHandler`.
  - **ThriftHiveMetaStoreClient**: un cliente de Thrift que se comunica con el metaalmacén de Hive. Este cliente está escrito para Hive 2.3.0. Si utiliza una versión de Hive distinta, es posible que tenga que actualizar esta clase para asegurarse de que los objetos de respuesta son compatibles.
  - **ThriftHiveMetaStoreClientFactory**: controla el comportamiento de la función Lambda. Por ejemplo, puede proporcionar su propio conjunto de proveedores de controladores anulando el método `getHandlerProvider()`.
- `hms.properties`: configura la función Lambda. La mayoría de los casos solo requieren actualizar las dos propiedades siguientes.
  - `hive.metastore.uris`: el URI del metaalmacén de Hive en el formato `thrift://<host_name>:9083`.
  - `hive.metastore.response.spill.location`: la ubicación de Amazon S3 para almacenar objetos de respuesta cuando los tamaños superan un umbral determinado (por ejemplo, 4 MB). El umbral se define en la propiedad `hive.metastore.response.spill.threshold`. No se recomienda cambiar el valor predeterminado.

 Note

Estas dos propiedades se pueden anular con las [variables de entorno de Lambda](#) `HMS_URIS` y `SPILL_LOCATION`. Utilice estas variables en lugar de volver a compilar el código fuente de la función de Lambda cuando desee utilizar la función con un metaalmacén de Hive o ubicación de desbordamiento diferente.

- **hms-lambda-layer**: un proyecto de ensamblaje de Maven que coloca a `hms-service-api`, `hms-lambda-handler` y sus dependencias en un archivo `.zip`. El archivo `.zip` se registra como una capa de Lambda para su uso por múltiples funciones de Lambda.
- **hms-lambda-rnp**: registra las respuestas de una función de Lambda y, a continuación, las usa para reproducir la respuesta. Puede usar este modelo para simular las respuestas de Lambda a fin de realizar pruebas.

Cree los artefactos usted mismo

La mayoría de los casos de uso no requieren que modifique la implementación de referencia. Sin embargo, si es necesario, puede modificar el código fuente, crear los artefactos usted mismo y cargarlos en una ubicación de Amazon S3.

Antes de crear los artefactos, actualice las propiedades `hive.metastore.uris` y `hive.metastore.response.spill.location` en el archivo `hms.properties` en el módulo `hms-lambda-func`.

Para construir los artefactos, debe tener instalado Apache Maven y ejecutar el comando `mvn install`. Esto genera el archivo `.zip` de capa en la carpeta de salida denominada `target` en el módulo `hms-lambda-layer` y el archivo `.jar` de la función Lambda en el módulo `hms-lambda-func`.

## Uso de consulta federada de Amazon Athena

Si tiene datos en orígenes distintos de Amazon S3, puede utilizar la consulta federada de Athena para consultar los datos en el lugar o crear canalizaciones que extraen datos de varios orígenes de datos y almacenarlos en Amazon S3. Con la consulta federada de Athena, puede ejecutar consultas SQL en datos almacenados en orígenes de datos relaciones, no relacionales, personalizados y de objetos.

Athena utiliza conectores de origen de datos que se ejecutan en AWS Lambda para ejecutar consultas federadas. Un conector de origen de datos es un fragmento de código que puede traducirse entre el origen de datos de destino y Athena. Puede pensar en un conector como una extensión del motor de consultas de Athena. Existen conectores de orígenes de datos de Athena para orígenes de datos como Amazon CloudWatch Logs, Amazon DynamoDB, Amazon DocumentDB y Amazon RDS, y orígenes de datos relacionales compatibles con JDBC como MySQL y PostgreSQL con licencia Apache 2.0. También puede utilizar el SDK de Athena Query Federation para escribir conectores personalizados. Para elegir, configurar e implementar un conector de origen

de datos en su cuenta, puede utilizar las consolas Athena y Lambda o AWS Serverless Application Repository. Después de implementar conectores de origen de datos, el conector se asocia a un catálogo que puede especificar en consultas SQL. Puede combinar instrucciones SQL de varios catálogos y abarcar varios orígenes de datos con una sola consulta.

Cuando se envía una consulta en un origen de datos, Athena invoca el conector correspondiente para identificar partes de las tablas que deben leerse, administrar el paralelismo y derribar los predicados de filtro. Según el usuario que envía la consulta, los conectores pueden proporcionar o restringir el acceso a elementos de datos específicos. Los conectores usan Apache Arrow como formato para devolver los datos solicitados en una consulta, lo que permite que los conectores se implementen en lenguajes como C, C++, Java, Python y Rust. Dado que los conectores se ejecutan en Lambda, se pueden utilizar para acceder a datos desde cualquier origen de datos en la nube o en las instalaciones accesible desde Lambda.

Para escribir su propio conector de origen de datos, puede utilizar el SDK de Athena Query Federation para personalizar uno de los conectores prediseñados que Amazon Athena proporciona y mantiene. Puede modificar una copia del código de origen desde el [repositorio de GitHub](#) y luego, utilizar la [herramienta de publicación de conectores](#) para crear su propio paquete de AWS Serverless Application Repository.

#### Note

Es posible que los desarrolladores de terceros hayan utilizado el SDK de Athena Query Federation para escribir conectores de origen de datos. Trabaje con su proveedor de conectores en caso de problemas de soporte o licencia con estos conectores de origen de datos. AWS no prueba ni admite estos conectores.

Para obtener una lista de conectores de orígenes de datos escritos y probados por Athena, consulte [Conectores de orígenes de datos disponibles](#).

Para obtener información sobre cómo escribir su propio conector de origen de datos, consulte [Example Athena connector](#) (Ejemplo de conector Athena) en GitHub.

## Consideraciones y limitaciones

- Versiones del motor: la consulta federada de Athena solo se admite en la versión 2 o una versión posterior del motor Athena. Para obtener más información acerca de las versiones de motor Athena, consulte [Control de versiones del motor Athena](#).



- **Vistas:** puede crear y consultar vistas en orígenes de datos federados. Las vistas federadas se almacenan en AWS Glue, no en el origen de datos subyacente. Para obtener más información, consulte [Consulta de vistas federadas](#).
- **Operaciones de escritura:** las operaciones de escritura como [INSERT INTO](#) no son compatibles. Si se intenta hacerlo, puede aparecer el mensaje de error This operation is currently not supported for external catalogs (Esta operación no es compatible actualmente con catálogos externos).
- **Precios:** para obtener información sobre precios, consulte [Precios de Amazon Athena](#).

**Controlador JDBC:** para utilizar el controlador JDBC con consultas federadas o un [metaalmacén externo de Hive](#), incluya `MetadataRetrieveMethod=ProxyAPI` en su cadena de conexión JDBC. Para obtener información acerca del controlador JDBC, consulte [Conexión a Amazon Athena con JDBC](#).

- **Secrets Manager:** para utilizar la característica de consulta federada de Athena con AWS Secrets Manager, debe configurar un punto de enlace privado de Amazon VPC para Secrets Manager. Para obtener más información, consulte [Creación de un punto de conexión privado de VPC de Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

Los conectores de origen de datos pueden requerir acceso a los siguientes recursos para funcionar correctamente. Si utiliza un conector prediseñado, compruebe la información del conector para asegurarse de que ha configurado correctamente la VPC. Además, asegúrese de que las entidades principales de IAM que ejecutan consultas y crean conectores dispongan de privilegios para las acciones necesarias. Para obtener más información, consulte [Ejemplos de políticas de permisos de IAM para permitir la consulta federada de Athena](#).

- **Amazon S3:** además de escribir los resultados de la consulta en la ubicación de resultados de la consulta de Athena en Amazon S3, los conectores de datos también escriben en un bucket de vertido en Amazon S3. Se requiere conectividad y permisos para esta ubicación de Amazon S3.
- **Athena:** los orígenes de datos necesitan conectividad con Athena y viceversa para verificar el estado de la consulta y evitar el sobreescaneo.
- **AWS Glue Data Catalog:** se requiere conectividad y permisos si el conector utiliza el catálogo de datos para metadatos complementarios o principales.

## Videos

Vea los siguientes videos para obtener más información sobre el uso de la consulta federada de Athena.

## Video: Analizar los resultados de la consulta federada de Amazon Athena en Amazon QuickSight

En el siguiente video, se muestra cómo analizar los resultados de una consulta federada de Athena en Amazon QuickSight.

[Analyze results of federated query in Amazon Athena in Amazon QuickSight](#) (Análisis de los resultados de consultas federadas de Amazon Athena en Amazon QuickSight)

## Video: Canalización de análisis de juegos

En el siguiente video, se muestra cómo implementar una canalización de datos escalable sin servidor para capturar, almacenar y analizar datos de telemetría de juegos y servicios mediante consultas federadas de Amazon Athena.

[Game analytics pipeline](#) (Canalización de análisis de juegos)

## Conectores de orígenes de datos disponibles

En esta sección se muestran los conectores de origen de datos de Athena prediseñados que se pueden utilizar para consultar una variedad de orígenes de datos externos en Amazon S3. Para utilizar un conector en sus consultas de Athena, configúrelo e impleméntelo en su cuenta.

### Consideraciones y limitaciones

- Algunos conectores creados previamente requieren que se cree una VPC y un grupo de seguridad antes de utilizar el conector. Para obtener información sobre la creación de las VPC, consulte [Creación de una VPC para un conector de origen de datos](#).
- Para utilizar la característica de consulta federada de Athena con AWS Secrets Manager, debe configurar un punto de conexión privado de Amazon VPC para Secrets Manager. Para obtener más información, consulte [Creación de un punto de conexión privado de VPC de Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.
- En el caso de los conectores que no admiten la inserción de predicados, las consultas que incluyen un predicado tardan mucho más en ejecutarse. Para conjuntos de datos pequeños, se escanean muy pocos datos y las consultas tardan un promedio de 2 minutos, aproximadamente. Sin embargo, para conjuntos de datos grandes, muchas consultas pueden agotar el tiempo de espera.
- Algunos orígenes de datos federados utilizan una terminología diferente de la de Athena para referirse a objetos de datos. Para obtener más información, consulte [Calificadores de nombres de tablas federadas y Athena](#).

- En el caso de los conectores que no admiten la paginación al enumerar tablas, el servicio web puede agotar el tiempo de espera si la base de datos tiene muchas tablas y metadatos. Los siguientes conectores admiten la paginación de las tablas de listados:
  - DocumentDB
  - DynamoDB
  - MySQL
  - OpenSearch
  - Oracle
  - PostgreSQL
  - Redshift
  - SQL Server


### Información adicional

- Para obtener información sobre la implementación de un conector de origen de datos de Athena, consulte [Implementación de un conector de origen de datos](#).
- Para obtener información sobre las consultas que utilizan conectores de orígenes de datos de Athena, consulte [Ejecución de consultas federadas](#).
- Para obtener información detallada sobre los conectores de origen de datos de Athena, consulte [Conectores disponibles](#) en GitHub.

### Conectores de origen de datos de Athena

- [Conector de Azure Data Lake Storage \(ADLS\) Gen2 de Amazon Athena](#)
- [Conector Azure Synapse de Amazon Athena](#)
- [Conector Cloudera Hive de Amazon Athena](#)
- [Conector Cloudera Impala de Amazon Athena](#)
- [Conector CloudWatch de Amazon Athena](#)
- [Conector de métricas de CloudWatch de Amazon Athena](#)
- [Conector AWS CMDB de Amazon Athena](#)
- [Conector para IBM Db2 de Amazon Athena](#)
- [Conector de Amazon Athena para IBM Db2 AS/400 \(Db2 iSeries\)](#)
- [Conector para DocumentDB de Amazon Athena](#)

- [Conector para DynamoDB de Amazon Athena](#)
- [Conector Google BigQuery de Amazon Athena](#)
- [Conector de Google Cloud Storage para Amazon Athena](#)
- [Conector para HBase de Amazon Athena](#)
- [Conector Hortonworks de Amazon Athena](#)
- [Conector Apache Kafka de Amazon Athena](#)
- [Conector para MSK de Amazon Athena](#)
- [Conector para MySQL de Amazon Athena](#)
- [Conector para Neptune de Amazon Athena](#)
- [Conector para OpenSearch de Amazon Athena](#)
- [Conector Oracle de Amazon Athena](#)
- [Conector para PostgreSQL de Amazon Athena](#)
- [Conector para Redis de Amazon Athena](#)
- [Conector para Redshift de Amazon Athena](#)
- [Conector SAP HANA de Amazon Athena](#)
- [Conector Snowflake de Amazon Athena](#)
- [Conector Microsoft SQL Server de Amazon Athena](#)
- [Conector Teradata de Amazon Athena](#)
- [Conector para Timestream de Amazon Athena](#)
- [Conector para TPC Benchmark DS \(TPC-DS\) de Amazon Athena](#)
- [Conector para Vertica de Amazon Athena](#)

 Note

El conector [AthenaJdbcConnector](#) (versión más reciente, 2022.4.1) ha quedado obsoleto. En su lugar, use un conector específico para bases de datos, como los de [MySQL](#), [Redshift](#) o [PostgreSQL](#).

## Conector de Azure Data Lake Storage (ADLS) Gen2 de Amazon Athena

El conector Gen2 de [Azure Data Lake Storage \(ADLS\)](#) de Amazon Athena permite que Amazon Athena ejecute consultas SQL en los datos almacenados en ADLS. Athena no puede acceder directamente a los archivos almacenados en el lago de datos.

- **Flujo de trabajo:** el conector implementa la interfaz JDBC, que utiliza el controlador `com.microsoft.sqlserver.jdbc.SQLServerDriver`. El conector pasa las consultas al motor de Azure Synapse, que luego accede al lago de datos.
- **Gestión de datos y S3:** por lo general, el conector de Lambda consulta los datos directamente sin transferirlos a Amazon S3. Sin embargo, cuando los datos devueltos por la función de Lambda superan los límites de Lambda, los datos se escriben en el bucket de distribución de Amazon S3 que especifique para que Athena pueda leer el exceso.
- **Autenticación AAD:** la AAD se puede utilizar como método de autenticación para el conector de Azure Synapse. Para utilizar AAD, la cadena de conexión JDBC que utiliza el conector debe contener los parámetros URL `authentication=ActiveDirectoryServicePrincipal`, `AADSecurePrincipalId` y `AADSecurePrincipalSecret`. Estos parámetros se pueden transferir directamente o mediante Secrets Manager.

### Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

### Limitaciones

- Las operaciones de escritura de DDL no son compatibles.
- En una configuración de multiplexor, el bucket de vertido y el prefijo se comparten en todas las instancias de base de datos.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.
- Los tipos de datos de marca de fecha y hora en condiciones de filtro se deben convertir a los tipos de datos adecuados.

## Términos

Los siguientes términos hacen referencia al conector de Azure Data Lake Storage Gen2.

- **Instancia de base de datos:** cualquier instancia de base de datos implementada en las instalaciones, en Amazon EC2 o en Amazon RDS.
- **Controlador:** un controlador de Lambda que accede a la instancia de base de datos. Un controlador puede ser para los metadatos o para los registros de datos.
- **Controlador de metadatos:** un controlador de Lambda que recupera los metadatos de la instancia de base de datos.
- **Controlador de registros:** un controlador de Lambda que recupera registros de datos de la instancia de base de datos.
- **Controlador compuesto:** un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.
- **Propiedad o parámetro:** una propiedad de base de datos que usan los controladores para extraer información de la base de datos. Estas propiedades se configuran como variables de entorno de Lambda.
- **Cadena de conexión:** una cadena de texto que se usa para establecer una conexión con una instancia de base de datos.
- **Catálogo:** un catálogo ajeno a AWS Glue registrado en Athena que es un prefijo obligatorio para la propiedad `connection_string`.
- **Controlador de multiplexación:** un controlador de Lambda que puede aceptar y usar varias conexiones a bases de datos.

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de Azure Data Lake Storage Gen2.

### Cadena de conexión

Use una cadena de conexión JDBC con el siguiente formato para conectarse a una instancia de base de datos.

```
datalakegentwo://${jdbc_connection_string}
```

## Uso de un controlador de multiplexación

Puede usar un multiplexor para conectarse a varias instancias de base de datos con una sola función de Lambda. Las solicitudes se enrutan por nombre de catálogo. Use las siguientes clases en Lambda.

Controlador	Clase
Controlador compuesto	<code>DataLakeGen2MuxCompositeHandler</code>
Controlador de metadatos	<code>DataLakeGen2MuxMetadataHandler</code>
Controlador de registros	<code>DataLakeGen2MuxRecordHandler</code>

## Parámetros del controlador de multiplexación

Parámetro	Descripción
<code><i>\$catalog_connection_string</i></code>	Obligatorio. Una cadena de conexión de instancia de base de datos. Agregue el prefijo a la variable de entorno con el nombre del catálogo usado en Athena. Por ejemplo, si el catálogo registrado en Athena es <code>mydatalakegentwocatalog</code> , el nombre de la variable de entorno será <code>mydatalakegentwocatalog_connection_string</code> .
<code>default</code>	Obligatorio. La cadena de conexión predeterminada. Esta cadena se usa cuando el catálogo es <code>lambda:\${ AWS_LAMBDA_FUNCTION_NAME }</code> .

Las siguientes propiedades de ejemplo son para una función de Lambda de `DataLakeGen2MUX` que admite dos instancias de base de datos: `dataLakegentwo1` (la predeterminada) y `dataLakegentwo2`.

Propiedad	Valor
default	<code>datalakegentwo://jdbc:sqlserver://adlsgentwo1. . <i>hostname:port</i>;databaseName= <i>database_name</i> ; \${<i>secret1_name</i> }</code>
<code>datalakegentwo_cat alog1_connection_s tring</code>	<code>datalakegentwo://jdbc:sqlserver://adlsgentwo1. . <i>hostname:port</i>;databaseName= <i>database_name</i> ; \${<i>secret1_name</i> }</code>
<code>datalakegentwo_cat alog2_connection_s tring</code>	<code>datalakegentwo://jdbc:sqlserver://adlsgentwo2. . <i>hostname:port</i>;databaseName= <i>database_name</i> ; \${<i>secret2_name</i> }</code>

## Proporcionar credenciales

Para proporcionar un nombre de usuario y una contraseña para la base de datos en la cadena de conexión JDBC, puede usar las propiedades de la cadena de conexión o AWS Secrets Manager.

- Cadena de conexión: se pueden especificar un nombre de usuario y una contraseña como propiedades en la cadena de conexión JDBC.

### Important

Como práctica recomendada en materia de seguridad, no utilice credenciales codificadas en las variables de entorno ni en las cadenas de conexión. Para obtener información sobre cómo transferir los secretos codificados a AWS Secrets Manager, consulte [Mover secretos codificados a AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

- AWS Secrets Manager: para utilizar la característica Consulta federada de Athena con AWS Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Puede poner el nombre de un secreto en AWS Secrets Manager, en la cadena de conexión JDBC. El conector reemplaza el nombre del secreto por los valores `username` y `password` de Secrets Manager.



Para las instancias de bases de datos de Amazon RDS, esta compatibilidad está estrechamente integrada. Si usa Amazon RDS, le recomendamos encarecidamente que use AWS Secrets Manager y la rotación de credenciales. Si la base de datos no usa Amazon RDS, guarde las credenciales como archivos JSON con el siguiente formato:

```
{"username": "${username}", "password": "${password}"}
```

### Ejemplo de cadena de conexión con nombre secreto

La siguiente cadena tiene el nombre secreto `${secret1_name}`.

```
datalakegentwo://jdbc:sqlserver://hostname:port;databaseName=database_name;  
${secret1_name}
```

El conector usa el nombre secreto para recuperar los secretos y proporcionar el nombre de usuario y la contraseña, como en el siguiente ejemplo.

```
datalakegentwo://  
jdbc:sqlserver://  
hostname:port;databaseName=database_name;user=user_name;password=password
```

### Uso de un controlador de conexión única

Puede usar los siguientes controladores de registros y metadatos de conexión única para conectarse a una sola instancia de Azure Data Lake Storage Gen2.

Tipo de controlador	Clase
Controlador compuesto	DataLakeGen2CompositeHandler
Controlador de metadatos	DataLakeGen2MetadataHandler
Controlador de registros	DataLakeGen2RecordHandler

## Parámetros de controlador de conexión única

Parámetro	Descripción
<code>default</code>	Obligatorio. La cadena de conexión predeterminada.

Los controladores de conexión única admiten una instancia de base de datos y deben proporcionar un parámetro de cadena de conexión `default`. Se omiten todas las demás cadenas de conexión.

La siguiente propiedad de ejemplo es para una sola instancia de Azure Data Lake Storage Gen2 compatible con una función de Lambda.

Propiedad	Valor
<code>default</code>	<code>datalakegentwo://jdbc:sqlserver:// <i>hostname:port</i>;database Name=;\${ <i>secret_name</i> }</code>

## Parámetros de vertido

El SDK de Lambda puede verter datos en Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación.

Parámetro	Descripción
<code>spill_bucket</code>	Obligatorio. Nombre del bucket de vertido.
<code>spill_prefix</code>	Obligatorio. Prefijo de la clave del bucket de vertido.
<code>spill_put_request_headers</code>	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud <code>putObject</code> de Amazon S3 que se usa para el vertido (por ejemplo, <code>{"x-amz-server-side-encryption" : "AES256"}</code> ). Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.

## Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos correspondientes para ADLS Gen2 y Arrow.

ADLS Gen2	Arrow
bit	TINYINT
tinyint	SMALLINT
smallint	SMALLINT
int	INT
bigint	BIGINT
decimal	DECIMAL
numérico	FLOAT8
smallmoney	FLOAT8
money	DECIMAL
float[24]	FLOAT4
float[53]	FLOAT8
real	FLOAT4
datetime	Date(MILLISECOND)
datetime2	Date(MILLISECOND)
smalldatetime	Date(MILLISECOND)
date	Date(DAY)
hora	VARCHAR
datetimeoffset	Date(MILLISECOND)

ADLS Gen2	Arrow
char[n]	VARCHAR
varchar[n/max]	VARCHAR

## Particiones y divisiones

Azure Data Lake Storage Gen2 usa almacenamiento de blobs de Gen2 compatible con Hadoop para almacenar archivos de datos. Los datos de estos archivos se consultan desde el motor de Azure Synapse. El motor de Azure Synapse trata los datos de Gen2 almacenados en los sistemas de archivos como tablas externas. Las particiones se implementan en función del tipo de datos. Si los datos ya se partieron y distribuyeron dentro del sistema de almacenamiento de Gen2, el conector recupera los datos como una sola división.

## Rendimiento

El conector Azure Data Lake Storage Gen2 muestra un rendimiento de consulta más lento cuando ejecuta varias consultas a la vez y está sujeto a limitaciones.

El conector Azure Data Lake Storage Gen2 de Athena inserta predicados para reducir los datos analizados en la consulta. Los predicados simples y las expresiones complejas se insertan en el conector para reducir la cantidad de datos analizados y disminuir el tiempo de ejecución de las consultas.

## Predicados

Un predicado es una expresión de la cláusula WHERE de una consulta SQL que da como resultado un valor booleano y filtra las filas en función de varias condiciones. El conector Azure Data Lake Storage Gen2 de Athena puede combinar estas expresiones e insertarlas directamente en Azure Data Lake Storage Gen2 para mejorar la funcionalidad y reducir la cantidad de datos analizados.

Los siguientes operadores del conector Azure Data Lake Storage Gen2 de Athena admiten la inserción de predicados:

- Booleano: AND, OR, NOT.
- Igualdad: EQUAL, NOT\_EQUAL, LESS\_THAN, LESS\_THAN\_OR\_EQUAL, GREATER\_THAN, GREATER\_THAN\_OR\_EQUAL, NULL\_IF, IS\_NULL.

- Aritmética: ADD, SUBTRACT, MULTIPLY, DIVIDE, MODULUS, NEGATE.
- Otros: LIKE\_PATTERN, IN.

## Ejemplo de inserción combinada

Para mejorar las capacidades de consulta, combine los tipos de inserciones, como en el siguiente ejemplo:

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%');
```

## Consultas de acceso directo

El conector de Azure Data Lake Storage Gen2 admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.

Para usar consultas de acceso directo con Azure Data Lake Storage Gen2, puede utilizar la siguiente sintaxis:

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  ))
```

El siguiente ejemplo de consulta envía una consulta a un origen de datos de Azure Data Lake Storage Gen2. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.

```
SELECT * FROM TABLE(
  system.query(
    query => 'SELECT * FROM customer LIMIT 10'
  ))
```

## Información sobre licencias

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

Para obtener la información más reciente sobre la versión del controlador JDBC, consulte el archivo [pom.xml](#) para el conector de Azure Data Lake Storage Gen2 en GitHub.com.

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector Azure Synapse de Amazon Athena

El conector de Amazon Athena de [Azure Synapse Analytics](#) permite que Amazon Athena ejecute consultas SQL en las bases de datos de Azure Synapse mediante JDBC.

## Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

## Limitaciones

- Las operaciones de escritura de DDL no son compatibles.
- En una configuración de multiplexor, el bucket de vertido y el prefijo se comparten en todas las instancias de base de datos.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.
- En condiciones de filtro, debe enviar los tipos de datos Date y Timestamp al tipo de dato correspondiente.
- Para buscar valores negativos de tipo Real y Float, use el operador <= o >=.
- Los tipos de datos binary, varbinary, image y rowversion no son compatibles.

## Términos

Los siguientes términos hacen referencia al conector de Synapse.

- **Instancia de base de datos:** cualquier instancia de base de datos implementada en las instalaciones, en Amazon EC2 o en Amazon RDS.
- **Controlador:** un controlador de Lambda que accede a la instancia de base de datos. Un controlador puede ser para los metadatos o para los registros de datos.
- **Controlador de metadatos:** un controlador de Lambda que recupera los metadatos de la instancia de base de datos.
- **Controlador de registros:** un controlador de Lambda que recupera registros de datos de la instancia de base de datos.
- **Controlador compuesto:** un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.
- **Propiedad o parámetro:** una propiedad de base de datos que usan los controladores para extraer información de la base de datos. Estas propiedades se configuran como variables de entorno de Lambda.
- **Cadena de conexión:** una cadena de texto que se usa para establecer una conexión con una instancia de base de datos.
- **Catálogo:** un catálogo ajeno a AWS Glue registrado en Athena que es un prefijo obligatorio para la propiedad `connection_string`.
- **Controlador de multiplexación:** un controlador de Lambda que puede aceptar y usar varias conexiones a bases de datos.

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de Synapse.

### Cadena de conexión

Use una cadena de conexión JDBC con el siguiente formato para conectarse a una instancia de base de datos.

```
synapse://${jdbc_connection_string}
```

## Uso de un controlador de multiplexación

Puede usar un multiplexor para conectarse a varias instancias de base de datos con una sola función de Lambda. Las solicitudes se enrutan por nombre de catálogo. Use las siguientes clases en Lambda.

Controlador	Clase
Controlador compuesto	<code>SynapseMuxCompositeHandler</code>
Controlador de metadatos	<code>SynapseMuxMetadataHandler</code>
Controlador de registros	<code>SynapseMuxRecordHandler</code>

## Parámetros del controlador de multiplexación

Parámetro	Descripción
<code><i>\$catalog_connection_string</i></code>	Obligatorio. Una cadena de conexión de instancia de base de datos. Agregue el prefijo a la variable de entorno con el nombre del catálogo usado en Athena. Por ejemplo, si el catálogo registrado en Athena es <code>mysynapsecatalog</code> , el nombre de la variable de entorno será <code>mysynapsecatalog_connection_string</code> .
<code>default</code>	Obligatorio. La cadena de conexión predeterminada. Esta cadena se usa cuando el catálogo es <code>lambda:\${AWS_LAMBDA_FUNCTION_NAME}</code> .

Las siguientes propiedades de ejemplo son para una función de Lambda de Synapse MUX que admite dos instancias de base de datos: `synapse1` (la predeterminada) y `synapse2`.

Propiedad	Valor
<code>default</code>	<code>synapse://jdbc:synapse://synapse1.hostname:port;databaseName= <i>&lt;database_name&gt;</i> ;\${secret1_name}</code>



Propiedad	Valor
synapse_catalog1_connection_string	synapse://jdbc:synapse://synapse1.hostname:port;databaseName= <i>&lt;database_name&gt;</i> ;\${secret1_name }
synapse_catalog2_connection_string	synapse://jdbc:synapse://synapse2.hostname:port;databaseName= <i>&lt;database_name&gt;</i> ;\${secret2_name }

## Proporcionar credenciales

Para proporcionar un nombre de usuario y una contraseña para la base de datos en la cadena de conexión JDBC, puede usar las propiedades de la cadena de conexión o AWS Secrets Manager.

- Cadena de conexión: se pueden especificar un nombre de usuario y una contraseña como propiedades en la cadena de conexión JDBC.

### Important

Como práctica recomendada en materia de seguridad, no utilice credenciales codificadas en las variables de entorno ni en las cadenas de conexión. Para obtener información sobre cómo transferir los secretos codificados a AWS Secrets Manager, consulte [Mover secretos codificados a AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

- AWS Secrets Manager: para utilizar la característica Consulta federada de Athena con AWS Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Puede poner el nombre de un secreto en AWS Secrets Manager, en la cadena de conexión JDBC. El conector reemplaza el nombre del secreto por los valores `username` y `password` de Secrets Manager.

Para las instancias de bases de datos de Amazon RDS, esta compatibilidad está estrechamente integrada. Si usa Amazon RDS, le recomendamos encarecidamente que use AWS Secrets

Manager y la rotación de credenciales. Si la base de datos no usa Amazon RDS, guarde las credenciales como archivos JSON con el siguiente formato:

```
{"username": "${username}", "password": "${password}"}
```

### Ejemplo de cadena de conexión con nombre secreto

La siguiente cadena tiene el nombre secreto `${secret_name}`.

```
synapse://jdbc:synapse://hostname:port;databaseName=<database_name>;${secret_name}
```

El conector usa el nombre secreto para recuperar los secretos y proporcionar el nombre de usuario y la contraseña, como en el siguiente ejemplo.

```
synapse://jdbc:synapse://  
hostname:port;databaseName=<database_name>;user=<user>;password=<password>
```

### Uso de un controlador de conexión única

Puede usar los siguientes controladores de registros y metadatos de conexión única para conectarse a una sola instancia de Synapse.

Tipo de controlador	Clase
Controlador compuesto	<code>SynapseCompositeHandler</code>
Controlador de metadatos	<code>SynapseMetadataHandler</code>
Controlador de registros	<code>SynapseRecordHandler</code>

### Parámetros de controlador de conexión única

Parámetro	Descripción
<code>default</code>	Obligatorio. La cadena de conexión predeterminada.

Los controladores de conexión única admiten una instancia de base de datos y deben proporcionar un parámetro de cadena de conexión default. Se omiten todas las demás cadenas de conexión.

La siguiente propiedad de ejemplo es para una sola instancia de Synapse compatible con una función de Lambda.

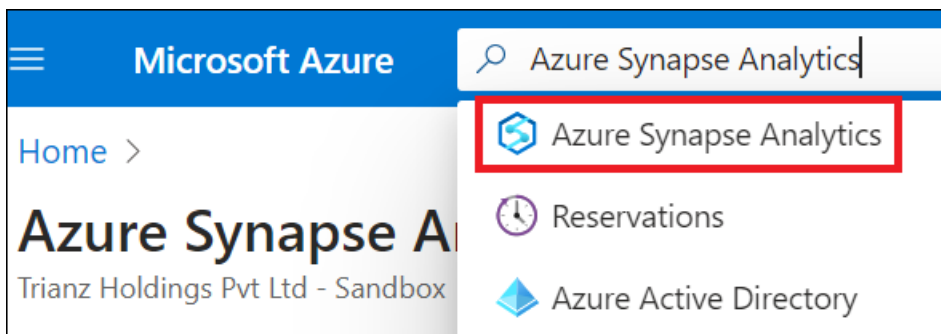
Propiedad	Valor
default	synapse://jdbc:sqlserver://hostname:port;databaseName= <i>&lt;database_name&gt;</i> ;\${ <i>secret_name</i> }

### Configuración de la autenticación con Active Directory

El conector Azure Synapse de Amazon Athena es compatible con la autenticación de Microsoft Active Directory. Antes de empezar, debe configurar un usuario administrativo en el portal de Microsoft Azure y, a continuación, usar AWS Secrets Manager para crear un secreto.

Para configurar el usuario de administración de Active Directory

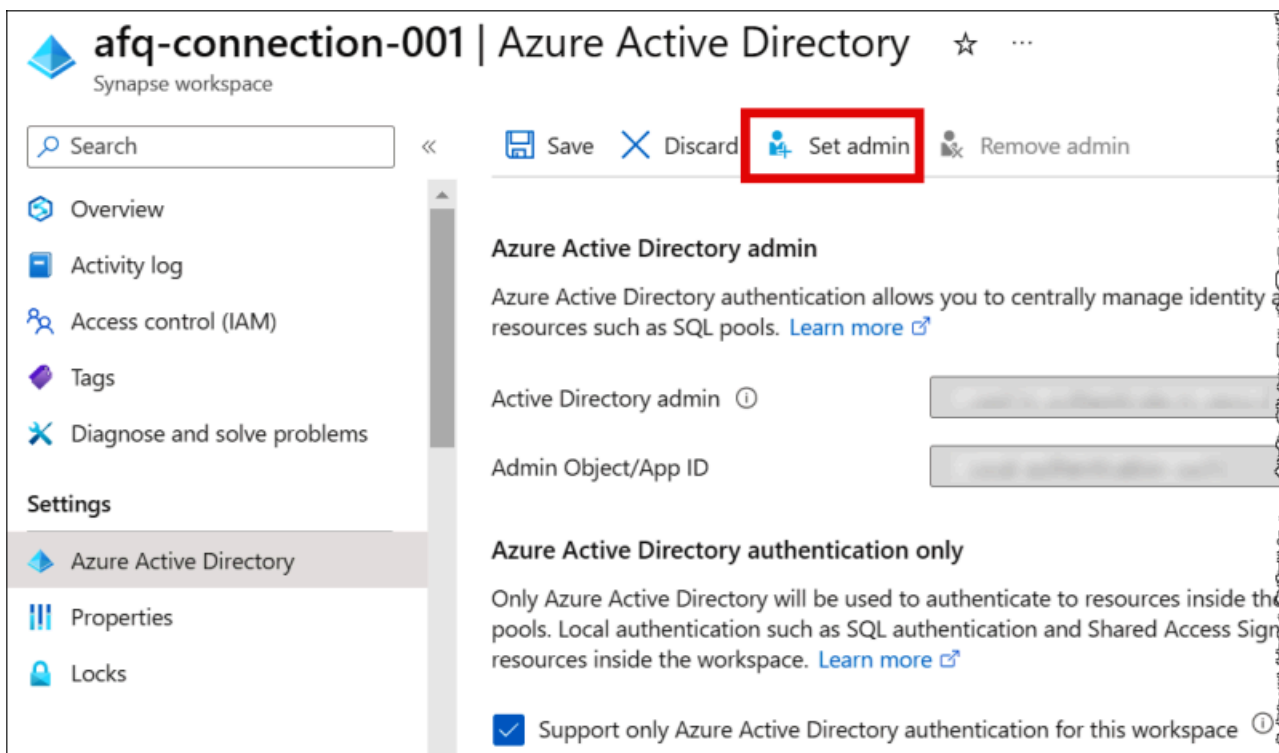
1. Con una cuenta que tenga privilegios administrativos, inicie sesión en el portal de Microsoft Azure en <https://portal.azure.com/>.
2. En el cuadro de búsqueda, escriba Azure Synapse Analytics y, a continuación, elija Azure Synapse Analytics.



3. Abra el menú de la izquierda.





4. En el panel de navegación, elija Azure Active Directory.
5. En la pestaña Configurar administrador, configure el administrador de Active Directory como un usuario nuevo o existente.



6. En AWS Secrets Manager, guarde las credenciales de nombre de usuario y contraseña del administrador. Para obtener información sobre cómo crear un secreto en Secrets Manager, consulte [Creación de un secreto de AWS Secrets Manager](#).

## Para ver el secreto en Secrets Manager

1. Abra la consola de Secrets Manager en <https://console.aws.amazon.com/secretsmanager/>.
2. En el panel de navegación, elija Secretos.
3. En la página Secretos, elija el vínculo al secreto.
4. En la página de detalles del secreto, elija Recuperar valor del secreto.

Key/value	Plaintext
Secret key	Secret value
username	
password	

## Modificación de la cadena de conexión

Para habilitar la autenticación de Active Directory para el conector, modifique la cadena de conexión mediante la siguiente sintaxis:

```
synapse://jdbc:synapse://
hostname:port;databaseName=database_name;authentication=ActiveDirectoryPassword;
{secret_name}
```

## Uso de ActiveDirectoryServicePrincipal

El conector Azure Synapse de Amazon Athena también es compatible con `ActiveDirectoryServicePrincipal`. Para habilitarlo, modifique la cadena de la siguiente manera.

```
synapse://jdbc:synapse://
hostname:port;databaseName=database_name;authentication=ActiveDirectoryServicePrincipal;
{secret_name}
```

Para `secret_name`, especifique el ID de la aplicación o del cliente como el nombre de usuario y el secreto de la identidad principal de un servicio en la contraseña.

## Parámetros de vertido

El SDK de Lambda puede verter datos en Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación.

Parámetro	Descripción
<code>spill_bucket</code>	Obligatorio. Nombre del bucket de vertido.
<code>spill_prefix</code>	Obligatorio. Prefijo de la clave del bucket de vertido.
<code>spill_put_request_headers</code>	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud <code>putObject</code> de Amazon S3 que se usa para el vertido (por ejemplo, <code>{"x-amz-server-side-encryption" : "AES256"}</code> ). Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.

## Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos correspondientes para Synapse y Apache Arrow.

Synapse	Arrow
bit	TINYINT
tinyint	SMALLINT
smallint	SMALLINT
int	INT
bigint	BIGINT
decimal	DECIMAL
numérico	FLOAT8

Synapse	Arrow
smallmoney	FLOAT8
money	DECIMAL
float[24]	FLOAT4
float[53]	FLOAT8
real	FLOAT4
datetime	Date(MILLISECOND)
datetime2	Date(MILLISECOND)
smalldatetime	Date(MILLISECOND)
date	Date(DAY)
hora	VARCHAR
datetimeoffset	Date(MILLISECOND)
char[n]	VARCHAR
varchar[n/max]	VARCHAR
nchar[n]	VARCHAR
nvarchar[n/max]	VARCHAR

## Particiones y divisiones

Una partición se representa mediante una sola columna de partición del tipo `varchar`. Synapse admite la partición por rango, por lo que la partición se implementa al extraer la columna de particiones y el rango de particiones de las tablas de metadatos de Synapse. Estos valores de rango se usan para crear las divisiones.

## Rendimiento

La selección de un subconjunto de columnas ralentiza significativamente el tiempo de ejecución de las consultas. El conector presenta limitaciones significativas debido a la simultaneidad.

El conector Synapse de Athena inserta predicados para reducir los datos analizados en la consulta. Los predicados simples y las expresiones complejas se insertan en el conector para reducir la cantidad de datos analizados y disminuir el tiempo de ejecución de las consultas.

## Predicados

Un predicado es una expresión de la cláusula WHERE de una consulta SQL que da como resultado un valor booleano y filtra las filas en función de varias condiciones. El conector Synapse de Athena puede combinar estas expresiones e insertarlas directamente en Synapse para mejorar la funcionalidad y reducir la cantidad de datos analizados.

Los siguientes operadores del conector Synapse de Athena admiten la inserción de predicados:

- Booleano: AND, OR, NOT.
- Igualdad: EQUAL, NOT\_EQUAL, LESS\_THAN, LESS\_THAN\_OR\_EQUAL, GREATER\_THAN, GREATER\_THAN\_OR\_EQUAL, NULL\_IF, IS\_NULL.
- Aritmética: ADD, SUBTRACT, MULTIPLY, DIVIDE, MODULUS, NEGATE.
- Otros: LIKE\_PATTERN, IN.

## Ejemplo de inserción combinada

Para mejorar las capacidades de consulta, combine los tipos de inserciones, como en el siguiente ejemplo:

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%');
```

## Consultas de acceso directo

El conector Synapse admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.



Para utilizar consultas de acceso directo, puede utilizar la siguiente sintaxis:

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'query string'  
    ))
```

En el siguiente ejemplo de consulta, se envía una consulta a un origen de datos de Synapse. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## Información sobre licencias

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

- Para obtener información sobre cómo utilizar Amazon QuickSight y Amazon Athena Federated Query a fin de crear paneles y visualizaciones de los datos almacenados en las bases de datos de Microsoft Azure Synapse, consulte el artículo [Realizar análisis multinube con Amazon QuickSight, Amazon Athena Federated Query y Microsoft Azure Synapse](#) en el Blog de macrodatos de AWS.
- Para obtener la información más reciente sobre la versión del controlador JDBC, consulte el archivo [pom.xml](#) para el conector de Synapse en GitHub.com.
- Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector Cloudera Hive de Amazon Athena

El conector Cloudera Hive de Amazon Athena permite que Athena ejecute consultas SQL en la distribución de [Cloudera Hive](#) con Hadoop. El conector transforma las consultas SQL de Athena en la sintaxis HiveQL equivalente.

## Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).
- Antes de usar este conector, debe configurar una VPC y un grupo de seguridad. Para obtener más información, consulte [Creación de una VPC para un conector de origen de datos](#).

## Limitaciones

- Las operaciones de escritura de DDL no son compatibles.
- En una configuración de multiplexor, el bucket de vertido y el prefijo se comparten en todas las instancias de base de datos.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.

## Términos

Los siguientes términos hacen referencia al conector de Cloudera Hive.

- Instancia de base de datos: cualquier instancia de base de datos implementada en las instalaciones, en Amazon EC2 o en Amazon RDS.
- Controlador: un controlador de Lambda que accede a la instancia de base de datos. Un controlador puede ser para los metadatos o para los registros de datos.
- Controlador de metadatos: un controlador de Lambda que recupera los metadatos de la instancia de base de datos.
- Controlador de registros: un controlador de Lambda que recupera registros de datos de la instancia de base de datos.
- Controlador compuesto: un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.
- Propiedad o parámetro: una propiedad de base de datos que usan los controladores para extraer información de la base de datos. Estas propiedades se configuran como variables de entorno de Lambda.

- Cadena de conexión: una cadena de texto que se usa para establecer una conexión con una instancia de base de datos.
- Catálogo: un catálogo ajeno a AWS Glue registrado en Athena que es un prefijo obligatorio para la propiedad `connection_string`.
- Controlador de multiplexación: un controlador de Lambda que puede aceptar y usar varias conexiones a bases de datos.

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de Cloudera Hive.

## Cadena de conexión

Use una cadena de conexión JDBC con el siguiente formato para conectarse a una instancia de base de datos.

```
hive://{jdbc_connection_string}
```

## Uso de un controlador de multiplexación

Puede usar un multiplexor para conectarse a varias instancias de base de datos con una sola función de Lambda. Las solicitudes se enrutan por nombre de catálogo. Use las siguientes clases en Lambda.

Controlador	Clase
Controlador compuesto	<code>HiveMuxCompositeHandler</code>
Controlador de metadatos	<code>HiveMuxMetadataHandler</code>
Controlador de registros	<code>HiveMuxRecordHandler</code>

## Parámetros del controlador de multiplexación

Parámetro	Descripción
<code>\$catalog_connection_string</code>	Obligatorio. Una cadena de conexión de instancia de base de datos. Agregue el prefijo a la variable de entorno con el nombre del catálogo usado en Athena. Por ejemplo, si el catálogo registrado en Athena es <code>myhivecatalog</code> , el nombre de la variable de entorno será <code>myhivecatalog_connection_string</code> .
<code>default</code>	Obligatorio. La cadena de conexión predeterminada. Esta cadena se usa cuando el catálogo es <code>lambda:\${AWS_LAMBDA_FUNCTION_NAME}</code> .

Las siguientes propiedades de ejemplo son para una función de Lambda de Hive MUX que admite dos instancias de base de datos: `hive1` (la predeterminada) y `hive2`.

Propiedad	Valor
<code>default</code>	<code>hive://jdbc:hive2://hive1:10000/default?\${Test/RDS/hive1}</code>
<code>hive2_catalog1_connection_string</code>	<code>hive://jdbc:hive2://hive1:10000/default?\${Test/RDS/hive1}</code>
<code>hive2_catalog2_connection_string</code>	<code>hive://jdbc:hive2://hive2:10000/default?UID=sample&amp;PWD=sample</code>

### Proporcionar credenciales

Para proporcionar un nombre de usuario y una contraseña para la base de datos en la cadena de conexión JDBC, puede usar las propiedades de la cadena de conexión o AWS Secrets Manager.

- Cadena de conexión: se pueden especificar un nombre de usuario y una contraseña como propiedades en la cadena de conexión JDBC.

**⚠ Important**

Como práctica recomendada en materia de seguridad, no utilice credenciales codificadas en las variables de entorno ni en las cadenas de conexión. Para obtener información sobre cómo transferir los secretos codificados a AWS Secrets Manager, consulte [Mover secretos codificados a AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

- AWS Secrets Manager: para utilizar la característica Consulta federada de Athena con AWS Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Puede poner el nombre de un secreto en AWS Secrets Manager, en la cadena de conexión JDBC. El conector reemplaza el nombre del secreto por los valores `username` y `password` de Secrets Manager.

Para las instancias de bases de datos de Amazon RDS, esta compatibilidad está estrechamente integrada. Si usa Amazon RDS, le recomendamos encarecidamente que use AWS Secrets Manager y la rotación de credenciales. Si la base de datos no usa Amazon RDS, guarde las credenciales como archivos JSON con el siguiente formato:

```
{"username": "${username}", "password": "${password}"}
```

**Ejemplo de cadena de conexión con nombre secreto**

La siguiente cadena tiene el nombre secreto `${Test/RDS/hive1}`.

```
hive://jdbc:hive2://hive1:10000/default?...&${Test/RDS/hive1}&...
```

El conector usa el nombre secreto para recuperar los secretos y proporcionar el nombre de usuario y la contraseña, como en el siguiente ejemplo.

```
hive://jdbc:hive2://hive1:10000/default?...&UID=sample2&PWD=sample2&...
```

Actualmente, el conector de Cloudera Hive reconoce las propiedades `UID` y `PWD` de JDBC.

## Uso de un controlador de conexión única

Puedes usar los siguientes controladores de registros y metadatos de conexión única para conectarte a una sola instancia de Cloudera Hive.

Tipo de controlador	Clase
Controlador compuesto	<code>HiveCompositeHandler</code>
Controlador de metadatos	<code>HiveMetadataHandler</code>
Controlador de registros	<code>HiveRecordHandler</code>

## Parámetros de controlador de conexión única

Parámetro	Descripción
<code>default</code>	Obligatorio. La cadena de conexión predeterminada.

Los controladores de conexión única admiten una instancia de base de datos y deben proporcionar un parámetro de cadena de conexión `default`. Se omiten todas las demás cadenas de conexión.

La siguiente propiedad de ejemplo es para una sola instancia de Cloudera Hive compatible con una función de Lambda.

Propiedad	Valor
<code>predeterminada</code>	<code>hive://jdbc:hive2://hive1:10000/default?secret=\${Test/RDS/hive1}</code>

## Parámetros de vertido

El SDK de Lambda puede verter datos en Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación.

Parámetro	Descripción
<code>spill_bucket</code>	Obligatorio. Nombre del bucket de vertido.
<code>spill_prefix</code>	Obligatorio. Prefijo de la clave del bucket de vertido.
<code>spill_put_request_headers</code>	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud <code>putObject</code> de Amazon S3 que se usa para el vertido (por ejemplo, <code>{"x-amz-server-side-encryption" : "AES256"}</code> ). Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.

### Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos correspondientes para JDBC, Cloudera Hive y Arrow.

JDBC	Cloudera Hive	Arrow
Booleano	Booleano	Bit
Entero	TINYINT	Pequeño
Short	SMALLINT	Smallint
Entero	INT	Int
Largo	BIGINT	Bigint
float	float4	Float4
Doble	float8	Float8
Date	date	DateDay
Timestamp	Marca de tiempo	DateMilli
Cadena	VARCHAR	Varchar

JDBC	Cloudera Hive	Arrow
Bytes	bytes	Varbinary
BigDecimal	Decimal	Decimal
ARRAY	N/D (ver nota)	Enumeración

### Note

Actualmente, Cloudera Hive no admite los tipos agregados ARRAY, MAP, STRUCT o UNIONTYPE. Las columnas de tipos agregados se tratan como columnas VARCHAR en SQL.

## Particiones y divisiones

Las particiones se usan para determinar cómo generar divisiones para el conector. Athena crea una columna sintética de tipo `varchar` que representa el esquema de partición de la tabla para ayudar al conector a generar divisiones. El conector no modifica la definición de la tabla real.

## Rendimiento

Cloudera Hive admite particiones estáticas. El conector Cloudera Hive de Athena puede recuperar datos de estas particiones en paralelo. Si quiere consultar conjuntos de datos muy grandes con una distribución uniforme de particiones, se recomienda encarecidamente la partición estática. El conector Cloudera Hive resiste las limitaciones debidas a la simultaneidad.

El conector Cloudera Hive de Athena inserta predicados para reducir los datos analizados en la consulta. Los predicados simples, las expresiones complejas y las cláusulas `LIMIT` se insertan en el conector para reducir la cantidad de datos analizados y disminuir el tiempo de ejecución de las consultas.

## Cláusulas LIMIT

Una instrucción `LIMIT N` reduce los datos analizados en la consulta. Con la inserción `LIMIT N`, el conector devuelve solo las filas `N` a Athena.



## Predicados

Un predicado es una expresión de la cláusula WHERE de una consulta SQL que da como resultado un valor booleano y filtra las filas en función de varias condiciones. El conector Cloudera Hive de Athena puede combinar estas expresiones e insertarlas directamente en Cloudera Hive para mejorar la funcionalidad y reducir la cantidad de datos analizados.

Los siguientes operadores del conector Cloudera Hive de Athena admiten la inserción de predicados:

- Booleano: AND, OR, NOT.
- Igualdad: EQUAL, NOT\_EQUAL, LESS\_THAN, LESS\_THAN\_OR\_EQUAL, GREATER\_THAN, GREATER\_THAN\_OR\_EQUAL, IS\_NULL.
- Aritmética: ADD, SUBTRACT, MULTIPLY, DIVIDE, MODULUS, NEGATE.
- Otros: LIKE\_PATTERN, IN.

### Ejemplo de inserción combinada

Para mejorar las capacidades de consulta, combine los tipos de inserciones, como en el siguiente ejemplo:

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
LIMIT 10;
```

### Consultas de acceso directo

El conector de Cloudera Hive admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.

Para usar consultas de acceso directo con Cloudera Hive, puede utilizar la siguiente sintaxis:

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  ))
```

El siguiente ejemplo de consulta envía una consulta a un origen de datos de Cloudera Hive. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## Información sobre licencias

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

Para obtener la información más reciente sobre la versión del controlador JDBC, consulte el archivo [pom.xml](#) para el conector de Cloudera Hive en GitHub.com.

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector Cloudera Impala de Amazon Athena

El conector de Cloudera Impala de Amazon Athena permite que Athena ejecute consultas SQL en la distribución de [Cloudera Impala](#). El conector transforma las consultas SQL de Athena en la sintaxis equivalente de Impala.

## Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).
- Antes de usar este conector, debe configurar una VPC y un grupo de seguridad. Para obtener más información, consulte [Creación de una VPC para un conector de origen de datos](#).

## Limitaciones

- Las operaciones de escritura de DDL no son compatibles.

- En una configuración de multiplexor, el bucket de vertido y el prefijo se comparten en todas las instancias de base de datos.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.

## Términos

Los siguientes términos hacen referencia al conector de Cloudera Impala.

- Instancia de base de datos: cualquier instancia de base de datos implementada en las instalaciones, en Amazon EC2 o en Amazon RDS.
- Controlador: un controlador de Lambda que accede a la instancia de base de datos. Un controlador puede ser para los metadatos o para los registros de datos.
- Controlador de metadatos: un controlador de Lambda que recupera los metadatos de la instancia de base de datos.
- Controlador de registros: un controlador de Lambda que recupera registros de datos de la instancia de base de datos.
- Controlador compuesto: un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.
- Propiedad o parámetro: una propiedad de base de datos que usan los controladores para extraer información de la base de datos. Estas propiedades se configuran como variables de entorno de Lambda.
- Cadena de conexión: una cadena de texto que se usa para establecer una conexión con una instancia de base de datos.
- Catálogo: un catálogo ajeno a AWS Glue registrado en Athena que es un prefijo obligatorio para la propiedad `connection_string`.
- Controlador de multiplexación: un controlador de Lambda que puede aceptar y usar varias conexiones a bases de datos.

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de Cloudera Impala.

## Cadena de conexión

Use una cadena de conexión JDBC con el siguiente formato para conectarse a un clúster de Impala.

```
impala://${jdbc_connection_string}
```

## Uso de un controlador de multiplexación

Puede usar un multiplexor para conectarse a varias instancias de base de datos con una sola función de Lambda. Las solicitudes se enrutan por nombre de catálogo. Use las siguientes clases en Lambda.

Controlador	Clase
Controlador compuesto	ImpalaMuxCompositeHandler
Controlador de metadatos	ImpalaMuxMetadataHandler
Controlador de registros	ImpalaMuxRecordHandler

## Parámetros del controlador de multiplexación

Parámetro	Descripción
<code>\${catalog_connection_string}</code>	Obligatorio. Cadena de conexión de clúster de Impala para un catálogo de Athena. Agregue el prefijo a la variable de entorno con el nombre del catálogo usado en Athena. Por ejemplo, si el catálogo registrado en Athena es <code>myimpalacatalog</code> , el nombre de la variable de entorno será <code>myimpalacatalog_connection_string</code> .
<code>default</code>	Obligatorio. La cadena de conexión predeterminada. Esta cadena se usa cuando el catálogo es <code>lambda:\${AWS_LAMBDA_FUNCTION_NAME}</code> .

Las siguientes propiedades de ejemplo son para una función de Lambda de Impala MUX que admite dos instancias de base de datos: `impala1` (la predeterminada) y `impala2`.

Propiedad	Valor
default	impala://jdbc:impala://some.impala.host.name:21050/?\${Test/impala1}
impala_catalog1_connection_string	impala://jdbc:impala://someother.impala.host.name:21050/?\${Test/impala1}
impala_catalog2_connection_string	impala://jdbc:impala://another.impala.host.name:21050/?UID=sample&PWD=sample

### Proporcionar credenciales

Para proporcionar un nombre de usuario y una contraseña para la base de datos en la cadena de conexión JDBC, puede usar las propiedades de la cadena de conexión o AWS Secrets Manager.

- Cadena de conexión: se pueden especificar un nombre de usuario y una contraseña como propiedades en la cadena de conexión JDBC.

#### Important

Como práctica recomendada en materia de seguridad, no utilice credenciales codificadas en las variables de entorno ni en las cadenas de conexión. Para obtener información sobre cómo transferir los secretos codificados a AWS Secrets Manager, consulte [Mover secretos codificados a AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

- AWS Secrets Manager: para utilizar la característica Consulta federada de Athena con AWS Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Puede poner el nombre de un secreto en AWS Secrets Manager, en la cadena de conexión JDBC. El conector reemplaza el nombre del secreto por los valores `username` y `password` de Secrets Manager.

Para las instancias de bases de datos de Amazon RDS, esta compatibilidad está estrechamente integrada. Si usa Amazon RDS, le recomendamos encarecidamente que use AWS Secrets Manager y la rotación de credenciales. Si la base de datos no usa Amazon RDS, guarde las credenciales como archivos JSON con el siguiente formato:

```
{"username": "${username}", "password": "${password}"}
```

## Ejemplo de cadena de conexión con nombre secreto

La siguiente cadena tiene el nombre secreto `${Test/impala1host}`.

```
impala://jdbc:impala://Impala1host:21050/?...&${Test/impala1host}&...
```

El conector usa el nombre secreto para recuperar los secretos y proporcionar el nombre de usuario y la contraseña, como en el siguiente ejemplo.

```
impala://jdbc:impala://Impala1host:21050/?...&UID=sample2&PWD=sample2&...
```

Actualmente, Cloudera Impala reconoce las propiedades UID y PWD de JDBC.

## Uso de un controlador de conexión única

Puedes usar los siguientes controladores de registros y metadatos de conexión única para conectarte a una sola instancia de Cloudera Impala.

Tipo de controlador	Clase
Controlador compuesto	<code>ImpalaCompositeHandler</code>
Controlador de metadatos	<code>ImpalaMetadataHandler</code>
Controlador de registros	<code>ImpalaRecordHandler</code>

## Parámetros de controlador de conexión única

Parámetro	Descripción
<code>default</code>	Obligatorio. La cadena de conexión predeterminada.

Los controladores de conexión única admiten una instancia de base de datos y deben proporcionar un parámetro de cadena de conexión default. Se omiten todas las demás cadenas de conexión.

La siguiente propiedad de ejemplo es para una sola instancia de Cloudera Impala compatible con una función de Lambda.

Propiedad	Valor
default	impala://jdbc:impala://Impala1host:21050/?secret=\${Test/impala1host}

### Parámetros de vertido

El SDK de Lambda puede verter datos en Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación.

Parámetro	Descripción
spill_bucket	Obligatorio. Nombre del bucket de vertido.
spill_prefix	Obligatorio. Prefijo de la clave del bucket de vertido.
spill_put_request_headers	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud putObject de Amazon S3 que se usa para el vertido (por ejemplo, {"x-amz-server-side-encryption" : "AES256"} ). Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.

### Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos correspondientes para JDBC, Cloudera Impala y Arrow.

JDBC	Cloudera Impala	Arrow
Booleano	Booleano	Bit

JDBC	Cloudera Impala	Arrow
Entero	TINYINT	Pequeño
Short	SMALLINT	Smallint
Entero	INT	Int
Largo	BIGINT	Bigint
float	float4	Float4
Doble	float8	Float8
Date	date	DateDay
Timestamp	Marca de tiempo	DateMilli
Cadena	VARCHAR	Varchar
Bytes	bytes	Varbinary
BigDecimal	Decimal	Decimal
ARRAY	N/D (ver nota)	Enumeración

#### Note

Actualmente, Cloudera Impala no admite los tipos agregados ARRAY, MAP, STRUCT o UNIONTYPE. Las columnas de tipos agregados se tratan como columnas VARCHAR en SQL.

## Particiones y divisiones

Las particiones se usan para determinar cómo generar divisiones para el conector. Athena crea una columna sintética de tipo `varchar` que representa el esquema de partición de la tabla para ayudar al conector a generar divisiones. El conector no modifica la definición de la tabla real.



## Rendimiento

Cloudera Impala admite particiones estáticas. El conector Cloudera Impala de Athena puede recuperar datos de estas particiones en paralelo. Si quiere consultar conjuntos de datos muy grandes con una distribución uniforme de particiones, se recomienda encarecidamente la partición estática. El conector Cloudera Impala resiste las limitaciones debidas a la simultaneidad.

El conector Cloudera Impala de Athena inserta predicados para reducir los datos analizados en la consulta. Los predicados simples, las expresiones complejas y las cláusulas LIMIT se insertan en el conector para reducir la cantidad de datos analizados y disminuir el tiempo de ejecución de las consultas.

### Cláusulas LIMIT

Una instrucción LIMIT N reduce los datos analizados en la consulta. Con la inserción LIMIT N, el conector devuelve solo las filas N a Athena.

### Predicados

Un predicado es una expresión de la cláusula WHERE de una consulta SQL que da como resultado un valor booleano y filtra las filas en función de varias condiciones. El conector Cloudera Impala de Athena puede combinar estas expresiones e insertarlas directamente en Cloudera Impala para mejorar la funcionalidad y reducir la cantidad de datos analizados.

Los siguientes operadores del conector Cloudera Impala de Athena admiten la inserción de predicados:

- Booleano: AND, OR, NOT.
- Igualdad: EQUAL, NOT\_EQUAL, LESS\_THAN, LESS\_THAN\_OR\_EQUAL, GREATER\_THAN, GREATER\_THAN\_OR\_EQUAL, IS\_DISTINCT\_FROM, NULL\_IF, IS\_NULL.
- Aritmética: ADD, SUBTRACT, MULTIPLY, DIVIDE, MODULUS, NEGATE.
- Otros: LIKE\_PATTERN, IN.

### Ejemplo de inserción combinada

Para mejorar las capacidades de consulta, combine los tipos de inserciones, como en el siguiente ejemplo:

```
SELECT *
```

```
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
LIMIT 10;
```

## Consultas de acceso directo

El conector de Cloudera Impala admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.

Para usar consultas de acceso directo con Cloudera Impala, puede utilizar la siguiente sintaxis:

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  )
)
```

El siguiente ejemplo de consulta envía una consulta a un origen de datos de Cloudera Impala. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.

```
SELECT * FROM TABLE(
  system.query(
    query => 'SELECT * FROM customer LIMIT 10'
  )
)
```

## Información sobre licencias

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

Para obtener la información más reciente sobre la versión del controlador JDBC, consulte el archivo [pom.xml](#) para el conector de Cloudera Impala en GitHub.com.

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector CloudWatch de Amazon Athena

El conector Amazon Athena CloudWatch permite que Amazon Athena se comunice con CloudWatch para que pueda consultar los datos de registro con SQL.

El conector asigna sus LogGroups como esquemas y cada LogStream como una tabla. El conector también asigna una vista especial `all_log_streams` que contiene todos los LogStreams del LogGroup. Esta vista le permite consultar todos los registros de un LogGroup a la vez en lugar de buscar a través de cada LogStream individualmente.

### Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

### Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de CloudWatch.

- `spill_bucket`: especifica el bucket de Amazon S3 para los datos que superen los límites de la función de Lambda.
- `spill_prefix`: (opcional) de forma predeterminada, se establece una subcarpeta en la carpeta especificada `spill_bucket` llamada `athena-federation-spill`. Le recomendamos configurar un [ciclo de vida de almacenamiento](#) de Amazon S3 en esta ubicación para eliminar vertidos de más de un número predeterminado de días u horas.
- `spill_put_request_headers`: (opcional) un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud `putObject` de Amazon S3 que se usa para el vertidos (por ejemplo, `{"x-amz-server-side-encryption" : "AES256"}`). Para ver otros encabezados posibles, consulte [PutObject](#) en la referencia de la API de Amazon Simple Storage Service.
- `kms_key_id`: (opcional) de forma predeterminada, los datos que se vierten a Amazon S3 se cifran mediante el modo de cifrado autenticado AES-GCM y una clave generada aleatoriamente. Para que la función de Lambda use claves de cifrado más seguras generadas por KMS, como `a7e63k4b-81oc-40db-a2a1-4d0en2cd8331`, puede especificar un ID de clave de KMS.
- `disable_spill_encryption`: (opcional) cuando se establece en `True`, desactiva el cifrado del vertido. El valor predeterminado es `False`, de modo que los datos que se vierten a S3 se cifran mediante AES-GCM, ya sea mediante una clave generada aleatoriamente o KMS para generar claves. La

desactivación del cifrado de vertido puede mejorar el rendimiento, especialmente si su ubicación de vertido usa [cifrado del servidor](#).

El conector también admite [control de congestión de AIMD](#) para gestionar los eventos de limitación de CloudWatch a través de la construcción `ThrottlingInvoker` del [SDK de federación de consultas de Amazon Athena](#). Para modificar el comportamiento de limitación predeterminado, puede configurar cualquiera de las siguientes variables de entorno opcionales:

- `throttle_initial_delay_ms`: el retraso de llamada inicial aplicado tras el primer evento de congestión. El valor predeterminado es 10 milisegundos.
- `throttle_max_delay_ms`: el retraso máximo entre llamadas. Puede derivar TPS dividiéndolo en 1000 ms. El valor predeterminado es 1000 milisegundos.
- `throttle_decrease_factor`: el factor por el que Athena reduce la tarifa de llamadas. El valor predeterminado es 0,5
- `throttle_increase_ms`: la velocidad a la que Athena disminuye el retraso de la llamada. El valor predeterminado es 10 milisegundos.

## Base de datos y tablas

El conector de CloudWatch Athena asigna sus LogGroups como esquemas, es decir, bases de datos, y cada LogStream como una tabla. El conector también asigna una vista especial `all_log_streams` que contiene todos los LogStreams del LogGroup. Esta vista le permite consultar todos los registros de un LogGroup a la vez en lugar de buscar a través de cada LogStream individualmente.

Todas las tablas asignadas por el conector de Athena CloudWatch tienen el siguiente esquema. Este esquema coincide con los campos proporcionados por Registros de CloudWatch.

- `log_stream`: VARCHAR que contiene el nombre del LogStream del que proviene la fila.
- `time`: INT64 que contiene la hora de la época en que se generó la línea de registro.
- `message`: VARCHAR que contiene el mensaje de registro.

## Ejemplos

En el siguiente ejemplo, se muestra cómo llevar a cabo una consulta SELECT en un LogStream especificado.

```
SELECT *
FROM "lambda:cloudwatch_connector_lambda_name"."log_group_path"."log_stream_name"
LIMIT 100
```

En el siguiente ejemplo, se muestra cómo usar la vista de `all_log_streams` para llevar a cabo una consulta en todos los LogStreams de un LogGroup especificado.

```
SELECT *
FROM "lambda:cloudwatch_connector_lambda_name"."log_group_path"."all_log_streams"
LIMIT 100
```

## Permisos necesarios

Para obtener información completa sobre las políticas de IAM que requiere este conector, consulte la sección `Políticas` del archivo [athena-cloudwatch.yaml](#). En la siguiente lista se resumen los permisos requeridos.

- Acceso de escritura a Amazon S3: el conector requiere acceso de escritura a una ubicación de Amazon S3 para volcar los resultados de consultas de gran tamaño.
- Athena GetQueryExecution: el conector usa este permiso para fallar rápidamente cuando finaliza la consulta ascendente de Athena.
- Lectura o escritura de Registros de CloudWatch: el conector usa este permiso para leer los datos de registro y escribir sus registros de diagnóstico.

## Rendimiento

El conector de Athena CloudWatch intenta optimizar las consultas contra CloudWatch mediante la paralelización de los escaneos de las transmisiones de registro necesarias para su consulta. Para determinados filtros de periodos de tiempo, la inserción de predicados se lleva a cabo tanto dentro de la función de Lambda como dentro de Registros de CloudWatch.

Para obtener el mejor rendimiento, use solo minúsculas en los nombres de los grupos de registros y flujos de registros. El uso combinado de mayúsculas y minúsculas hace que el conector realice una búsqueda que no distinga mayúsculas de minúsculas, lo que requiere un mayor esfuerzo computacional.

## Consultas de acceso directo

El conector CloudWatch admite [consultas de acceso directo](#) que utilizan la [sintaxis de consulta de información de registros de CloudWatch](#). Para obtener más información sobre la información de registros de CloudWatch, consulte [Análisis de los datos de registros con información de registros de CloudWatch](#) en la Guía del usuario de Registros de Amazon CloudWatch.

Para usar consultas de acceso directo con CloudWatch, puede utilizar la siguiente sintaxis:

```
SELECT * FROM TABLE(  
  system.query(  
    STARTTIME => 'start_time',  
    ENDTIME => 'end_time',  
    QUERYSTRING => 'query_string',  
    LOGGROUPNAMES => 'log_group-names',  
    LIMIT => 'max_number_of_results'  
  ))
```

El siguiente ejemplo de consulta de acceso directo de CloudWatch filtra el campo `duration` cuando no es igual a 1000.

```
SELECT * FROM TABLE(  
  system.query(  
    STARTTIME => '1710918615308',  
    ENDTIME => '1710918615972',  
    QUERYSTRING => 'fields @duration | filter @duration != 1000',  
    LOGGROUPNAMES => '/aws/lambda/cloudwatch-test-1',  
    LIMIT => '2'  
  ))
```

## Información sobre licencias

El proyecto del conector de CloudWatch de Amazon Athena se licencia en virtud de la [Licencia de Apache-2.0](#).

## Recursos adicionales de

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector de métricas de CloudWatch de Amazon Athena

El conector de Amazon Athena para métricas de CloudWatch permite que Amazon Athena consulte datos de métricas de CloudWatch con SQL.

Para obtener información sobre cómo publicar métricas de consulta en CloudWatch desde el propio servicio de Athena, consulte [Control de costos y supervisión de consultas con métricas y eventos de CloudWatch](#).

### Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

### Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de métricas de CloudWatch.

- `spill_bucket`: especifica el bucket de Amazon S3 para los datos que superen los límites de la función de Lambda.
- `spill_prefix`: (opcional) de forma predeterminada, se establece una subcarpeta en la carpeta especificada `spill_bucket` llamada `athena-federation-spill`. Le recomendamos configurar un [ciclo de vida de almacenamiento](#) de Amazon S3 en esta ubicación para eliminar vertidos de más de un número predeterminado de días u horas.
- `spill_put_request_headers`: (opcional) un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud `putObject` de Amazon S3 que se usa para el vertido (por ejemplo, `{"x-amz-server-side-encryption" : "AES256"}`). Para ver otros encabezados posibles, consulte [PutObject](#) en la referencia de la API de Amazon Simple Storage Service.
- `kms_key_id`: (opcional) de forma predeterminada, los datos que se vierten a Amazon S3 se cifran mediante el modo de cifrado autenticado AES-GCM y una clave generada aleatoriamente. Para que la función de Lambda use claves de cifrado más seguras generadas por KMS, como `a7e63k4b-81oc-40db-a2a1-4d0en2cd8331`, puede especificar un ID de clave de KMS.
- `disable_spill_encryption`: (opcional) cuando se establece en `True`, desactiva el cifrado del vertido. El valor predeterminado es `False`, de modo que los datos que se vierten a S3 se cifran mediante

AES-GCM, ya sea mediante una clave generada aleatoriamente o KMS para generar claves. La desactivación del cifrado de vertido puede mejorar el rendimiento, especialmente si su ubicación de vertido usa [cifrado del servidor](#).

El conector también admite [control de congestión de AIMD](#) para gestionar los eventos de limitación de CloudWatch a través de la construcción `ThrottlingInvoker` del [SDK de federación de consultas de Amazon Athena](#). Para modificar el comportamiento de limitación predeterminado, puede configurar cualquiera de las siguientes variables de entorno opcionales:

- `throttle_initial_delay_ms`: el retraso de llamada inicial aplicado tras el primer evento de congestión. El valor predeterminado es 10 milisegundos.
- `throttle_max_delay_ms`: el retraso máximo entre llamadas. Puede derivar TPS dividiéndolo en 1000 ms. El valor predeterminado es 1000 milisegundos.
- `throttle_decrease_factor`: el factor por el que Athena reduce la tarifa de llamadas. El valor predeterminado es 0,5
- `throttle_increase_ms`: la velocidad a la que Athena disminuye el retraso de la llamada. El valor predeterminado es 10 milisegundos.

## Base de datos y tablas

El conector de métricas de CloudWatch de Athena asigna sus espacios de nombres, dimensiones, métricas y valores de métricas en dos tablas en un único esquema denominado `default`.

### La tabla de métricas

La tabla `metrics` contiene las métricas disponibles definidas de forma exclusiva mediante una combinación de espacio de nombres, conjunto y nombre. La tabla `metrics` contiene las siguientes columnas.

- `namespace`: VARCHAR que contiene el espacio de nombres.
- `metric_name`: VARCHAR que contiene el nombre de la métrica.
- `dimensions`: LIST de objetos STRUCT compuestos por `dim_name` (VARCHAR) y `dim_value` (VARCHAR).
- `statistic`: LIST de estadísticas VARCHAR (por ejemplo, `p90`, `AVERAGE`, ...) disponibles para la métrica.



## La tabla `metric_samples`

La tabla `metric_samples` contiene las muestras de métricas disponibles para cada métrica de la tabla de `metrics`. La tabla `metric_samples` contiene las siguientes columnas.

- `namespace`: VARCHAR que contiene el espacio de nombres.
- `metric_name`: VARCHAR que contiene el nombre de la métrica.
- `dimensions`: LIST de objetos STRUCT compuestos por `dim_name` (VARCHAR) y `dim_value` (VARCHAR).
- `dim_name`: un campo de conveniencia VARCHAR que puede usar para filtrar fácilmente por un nombre de dimensión único.
- `dim_value`: un campo de conveniencia VARCHAR que puede usar para filtrar fácilmente un valor de dimensión única.
- `period`: INT que representa el “periodo” de la métrica en segundos (por ejemplo, una métrica de 60 segundos).
- `timestamp`: BIGINT que representa el tiempo de época en segundos para el que está destinada la muestra métrica.
- `value`: un FLOAT8 que contiene el valor de la muestra.
- `statistic`: VARCHAR que contiene el tipo de estadística de la muestra (por ejemplo, AVERAGE o p90).

## Permisos necesarios

Para obtener información completa sobre las políticas de IAM que requiere este conector, consulte la sección `Policies` del archivo [athena-cloudwatch-metrics.yaml](#). En la siguiente lista se resumen los permisos requeridos.

- Acceso de escritura a Amazon S3: el conector requiere acceso de escritura a una ubicación de Amazon S3 para volcar los resultados de consultas de gran tamaño.
- Athena GetQueryExecution: el conector usa este permiso para fallar rápidamente cuando finaliza la consulta ascendente de Athena.
- Métricas de CloudWatch de solo lectura: el conector usa este permiso para consultar los datos de las métricas.
- Escritura de Registros de CloudWatch: el conector usa este acceso para escribir sus registros de diagnóstico.

## Rendimiento

El conector de métricas de CloudWatch de Athena intenta optimizar las consultas con respecto a las métricas de CloudWatch paralelizando los escaneos de los flujos de registro necesarios para su consulta. Para determinados filtros de periodo de tiempo, métrica, espacio de nombres y dimensiones, la inserción de predicados se lleva a cabo tanto dentro de la función de Lambda como dentro de Registros de CloudWatch.

## Información sobre licencias

El proyecto del conector de métricas de CloudWatch de Amazon Athena se licencia en virtud de la [Licencia Apache-2.0](#).

## Recursos adicionales de

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector AWS CMDB de Amazon Athena

El conector de AWS CMDB de Amazon Athena permite a Amazon Athena comunicarse con varios servicios de AWS para que pueda consultarlos con SQL.

## Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de AWS CMDB.

- `spill_bucket`: especifica el bucket de Amazon S3 para los datos que superen los límites de la función de Lambda.
- `spill_prefix`: (opcional) de forma predeterminada, se establece una subcarpeta en la carpeta especificada `spill_bucket` llamada `athena-federation-spill`. Le recomendamos configurar un [ciclo de vida de almacenamiento](#) de Amazon S3 en esta ubicación para eliminar vertidos de más de un número predeterminado de días u horas.

- `spill_put_request_headers`: (opcional) un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud `putObject` de Amazon S3 que se usa para el vertidos (por ejemplo, `{"x-amz-server-side-encryption" : "AES256"}`). Para ver otros encabezados posibles, consulte [PutObject](#) en la referencia de la API de Amazon Simple Storage Service.
- `kms_key_id`: (opcional) de forma predeterminada, los datos que se vierten a Amazon S3 se cifran mediante el modo de cifrado autenticado AES-GCM y una clave generada aleatoriamente. Para que la función de Lambda use claves de cifrado más seguras generadas por KMS, como `a7e63k4b-81oc-40db-a2a1-4d0en2cd8331`, puede especificar un ID de clave de KMS.
- `disable_spill_encryption`: (opcional) cuando se establece en `True`, desactiva el cifrado del vertido. El valor predeterminado es `False`, de modo que los datos que se vierten a S3 se cifran mediante AES-GCM, ya sea mediante una clave generada aleatoriamente o KMS para generar claves. La desactivación del cifrado de vertido puede mejorar el rendimiento, especialmente si su ubicación de vertido usa [cifrado del servidor](#).
- `default_ec2_image_owner`: (opcional) cuando se establece, controla el propietario predeterminado de la imagen de Amazon EC2 que filtra [Imágenes de máquina de Amazon \(AMI\)](#). Si no establece este valor y la consulta en la tabla de imágenes de EC2 no incluye ningún filtro para el propietario, los resultados incluirán todas las imágenes públicas.

## Base de datos y tablas

El conector de AWS CMDB de Athena pone a disposición las siguientes bases de datos y tablas para consultar su inventario de recursos de AWS. Para obtener más información sobre las columnas disponibles en cada tabla, ejecute una instrucción `DESCRIBE database.table` mediante la consola o la API de Athena.

- `ec2`: esta base de datos contiene recursos relacionados con Amazon EC2, incluidos los siguientes.
- `ebs_volumes`: contiene detalles de sus volúmenes de Amazon EBS.
- `ec2_instances`: contiene detalles de las instancias de EC2.
- `ec2_images`: contiene detalles de las imágenes de las instancias de EC2.
- `routing_tables`: contiene detalles de las tablas de enrutamiento de la VPC.
- `security_groups`: contiene detalles de los grupos de seguridad.
- `subnets`: contiene detalles de las subredes de VPC.
- `vpcs`: contiene detalles de las VPC.

- `emr`: esta base de datos contiene recursos relacionados con Amazon EMR, incluidos los siguientes.
- `emr_clusters`: contiene detalles de los clústeres de EMR.
- `rds`: esta base de datos contiene recursos relacionados con Amazon RDS, incluidos los siguientes.
- `rds_instances`: contiene detalles de las instancias de RDS.
- `s3`: esta base de datos contiene recursos relacionados con RDS, incluidos los siguientes.
- `buckets`: contiene detalles de sus buckets de Amazon S3.
- `objects`: contiene detalles de sus objetos de Amazon S3, sin incluir su contenido.

## Permisos necesarios

Para obtener información completa sobre las políticas de IAM que requiere este conector, consulte la sección `Policies` del archivo [athena-aws-cmdb.yaml](#). En la siguiente lista se resumen los permisos requeridos.

- Acceso de escritura a Amazon S3: el conector requiere acceso de escritura a una ubicación de Amazon S3 para volcar los resultados de consultas de gran tamaño.
- Athena `GetQueryExecution`: el conector usa este permiso para fallar rápidamente cuando finaliza la consulta ascendente de Athena.
- Lista de S3: el conector utiliza este permiso para enumerar sus buckets y objetos de Amazon S3.
- Descripción de EC2: el conector utiliza este permiso para describir recursos como las instancias de Amazon EC2, los grupos de seguridad, las VPC y los volúmenes de Amazon EBS.
- Descripción o Lista de EMR: el conector usa este permiso para describir los clústeres de EMR.
- Descripción de RDS: el conector usa este permiso para describir las instancias de RDS.

## Rendimiento

Actualmente, el conector de AWS CMDB de Athena no admite escaneos paralelos. La inserción de predicados se lleva a cabo dentro de la función de Lambda. Siempre que sea posible, los predicados

parciales se envían a los servicios que se consultan. Por ejemplo, una consulta de los detalles de una instancia de Amazon EC2 específica llama a la API de EC2 con el ID de instancia específico para ejecutar una operación de descripción específica.

## Información sobre licencias

El proyecto del conector de AWS CMDB de Amazon Athena se licencia en virtud de la [Licencia de Apache-2.0](#).

## Recursos adicionales de

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector para IBM Db2 de Amazon Athena

El conector de Amazon Athena para Db2 permite que Amazon Athena ejecute consultas SQL en las bases de datos de IBM Db2 mediante JDBC.

## Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).
- Antes de usar este conector, debe configurar una VPC y un grupo de seguridad. Para obtener más información, consulte [Creación de una VPC para un conector de origen de datos](#).

## Limitaciones

- Las operaciones de escritura de DDL no son compatibles.
- En una configuración de multiplexor, el bucket de vertido y el prefijo se comparten en todas las instancias de base de datos.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.
- Los tipos de datos de marca de fecha y hora en condiciones de filtro se deben convertir a los tipos de datos adecuados.

## Términos

Los siguientes términos hacen referencia al conector para Db2.

- **Instancia de base de datos:** cualquier instancia de base de datos implementada en las instalaciones, en Amazon EC2 o en Amazon RDS.
- **Controlador:** un controlador de Lambda que accede a la instancia de base de datos. Un controlador puede ser para los metadatos o para los registros de datos.
- **Controlador de metadatos:** un controlador de Lambda que recupera los metadatos de la instancia de base de datos.
- **Controlador de registros:** un controlador de Lambda que recupera registros de datos de la instancia de base de datos.
- **Controlador compuesto:** un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.
- **Propiedad o parámetro:** una propiedad de base de datos que usan los controladores para extraer información de la base de datos. Estas propiedades se configuran como variables de entorno de Lambda.
- **Cadena de conexión:** una cadena de texto que se usa para establecer una conexión con una instancia de base de datos.
- **Catálogo:** un catálogo ajeno a AWS Glue registrado en Athena que es un prefijo obligatorio para la propiedad `connection_string`.
- **Controlador de multiplexación:** un controlador de Lambda que puede aceptar y usar varias conexiones a bases de datos.

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector para Db2.

### Cadena de conexión

Use una cadena de conexión JDBC con el siguiente formato para conectarse a una instancia de base de datos.

```
dbtwo://${jdbc_connection_string}
```

## Uso de un controlador de multiplexación

Puede usar un multiplexor para conectarse a varias instancias de base de datos con una sola función de Lambda. Las solicitudes se enrutan por nombre de catálogo. Use las siguientes clases en Lambda.

Controlador	Clase
Controlador compuesto	Db2MuxCompositeHandler
Controlador de metadatos	Db2MuxMetadataHandler
Controlador de registros	Db2MuxRecordHandler

## Parámetros del controlador de multiplexación

Parámetro	Descripción
<code><i>\$catalog_connection_string</i></code>	Obligatorio. Una cadena de conexión de instancia de base de datos. Agregue el prefijo a la variable de entorno con el nombre del catálogo usado en Athena. Por ejemplo, si el catálogo registrado en Athena es <code>mydbtwocatalog</code> , el nombre de la variable de entorno será <code>mydbtwocatalog_connection_string</code> .
<code>default</code>	Obligatorio. La cadena de conexión predeterminada. Esta cadena se usa cuando el catálogo es <code>lambda:\${AWS_LAMBDA_FUNCTION_NAME}</code> .

Las siguientes propiedades de ejemplo son para una función de Lambda de Db2 MUX que admite dos instancias de base de datos: `dbtwo1` (la predeterminada) y `dbtwo2`.

Propiedad	Valor
<code>default</code>	<code>dbtwo://jdbc:db2://dbtwo1.hostname:port/<i>database_name</i> :\${secret1_name}</code>

Propiedad	Valor
dbtwo_catalog1_connection_string	dbtwo://jdbc:db2://dbtwo1. hostname: port/ <i>database_name</i> :\${secret1_name }
dbtwo_catalog2_connection_string	dbtwo://jdbc:db2://dbtwo2. hostname: port/ <i>database_name</i> :\${secret2_name }

## Proporcionar credenciales

Para proporcionar un nombre de usuario y una contraseña para la base de datos en la cadena de conexión JDBC, puede usar las propiedades de la cadena de conexión o AWS Secrets Manager.

- Cadena de conexión: se pueden especificar un nombre de usuario y una contraseña como propiedades en la cadena de conexión JDBC.

### Important

Como práctica recomendada en materia de seguridad, no utilice credenciales codificadas en las variables de entorno ni en las cadenas de conexión. Para obtener información sobre cómo transferir los secretos codificados a AWS Secrets Manager, consulte [Mover secretos codificados a AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

- AWS Secrets Manager: para utilizar la característica Consulta federada de Athena con AWS Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Puede poner el nombre de un secreto en AWS Secrets Manager, en la cadena de conexión JDBC. El conector reemplaza el nombre del secreto por los valores `username` y `password` de Secrets Manager.

Para las instancias de bases de datos de Amazon RDS, esta compatibilidad está estrechamente integrada. Si usa Amazon RDS, le recomendamos encarecidamente que use AWS Secrets Manager y la rotación de credenciales. Si la base de datos no usa Amazon RDS, guarde las credenciales como archivos JSON con el siguiente formato:

```
{"username": "${username}", "password": "${password}"}
```



## Ejemplo de cadena de conexión con nombre secreto

La siguiente cadena tiene el nombre secreto `${secret_name}`.

```
dbtwo://jdbc:db2://hostname:port/database_name:${secret_name}
```

El conector usa el nombre secreto para recuperar los secretos y proporcionar el nombre de usuario y la contraseña, como en el siguiente ejemplo.

```
dbtwo://jdbc:db2://hostname:port/database_name:user=user_name;password=password;
```

## Uso de un controlador de conexión única

Puede usar los siguientes controladores de registros y metadatos de conexión única para conectarse a una sola instancia de Db2.

Tipo de controlador	Clase
Controlador compuesto	<code>Db2CompositeHandler</code>
Controlador de metadatos	<code>Db2MetadataHandler</code>
Controlador de registros	<code>Db2RecordHandler</code>

## Parámetros de controlador de conexión única

Parámetro	Descripción
<code>default</code>	Obligatorio. La cadena de conexión predeterminada.

Los controladores de conexión única admiten una instancia de base de datos y deben proporcionar un parámetro de cadena de conexión `default`. Se omiten todas las demás cadenas de conexión.

La siguiente propiedad de ejemplo es para una sola instancia de Db2 compatible con una función de Lambda.

Propiedad	Valor
default	dbtwo://jdbc:db2://hostname:port/ <i>database_name</i> :\${secret_name}

## Parámetros de vertido

El SDK de Lambda puede verter datos en Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación.

Parámetro	Descripción
spill_bucket	Obligatorio. Nombre del bucket de vertido.
spill_prefix	Obligatorio. Prefijo de la clave del bucket de vertido.
spill_put_request_headers	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud <code>putObject</code> de Amazon S3 que se usa para el vertido (por ejemplo, <code>{"x-amz-server-side-encryption" : "AES256"}</code> ). Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.

## Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos correspondientes para JDBC y Arrow.

Db2	Arrow
CHAR	VARCHAR
VARCHAR	VARCHAR
FECHA	DATEDAY
HORA	VARCHAR
MARCA DE TIEMPO	DATEMILLI

Db2	Arrow
DATETIME	DATEMILLI
BOOLEAN	BOOL
SMALLINT	SMALLINT
INTEGER	INT
BIGINT	BIGINT
DECIMAL	DECIMAL
REAL	FLOAT8
DOBLE	FLOAT8
DECFLOAT	FLOAT8

## Particiones y divisiones

Una partición se representa mediante una o varias columnas de partición de tipo `varchar`. El conector para Db2 crea particiones mediante los siguientes esquemas de organización.

- Distribución por hash
- Partición por intervalo
- Organización por dimensiones

El conector recupera los detalles de la partición, como el número de particiones y el nombre de la columna, de una o varias tablas de metadatos de Db2. Las divisiones se crean en función del número de particiones identificadas.

## Rendimiento

El conector Db2 de Athena inserta predicados para reducir los datos analizados en la consulta. Los predicados simples, las expresiones complejas y las cláusulas `LIMIT` se insertan en el conector para reducir la cantidad de datos analizados y disminuir el tiempo de ejecución de las consultas.

## Cláusulas LIMIT

Una instrucción `LIMIT N` reduce los datos analizados en la consulta. Con la inserción `LIMIT N`, el conector devuelve solo las filas `N` a Athena.

## Predicados

Un predicado es una expresión de la cláusula `WHERE` de una consulta SQL que da como resultado un valor booleano y filtra las filas en función de varias condiciones. El conector Db2 de Athena puede combinar estas expresiones e insertarlas directamente en Db2 para mejorar la funcionalidad y reducir la cantidad de datos analizados.

Los siguientes operadores del conector Db2 de Athena admiten la inserción de predicados:

- Booleano: `AND`, `OR`, `NOT`.
- Igualdad: `EQUAL`, `NOT_EQUAL`, `LESS_THAN`, `LESS_THAN_OR_EQUAL`, `GREATER_THAN`, `GREATER_THAN_OR_EQUAL`, `IS_DISTINCT_FROM`, `IS_NULL`.
- Aritmética: `ADD`, `SUBTRACT`, `MULTIPLY`, `DIVIDE`, `MODULUS`, `NEGATE`.
- Otros: `LIKE_PATTERN`, `IN`.

## Ejemplo de inserción combinada

Para mejorar las capacidades de consulta, combine los tipos de inserciones, como en el siguiente ejemplo:

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
LIMIT 10;
```

## Consultas de acceso directo

El conector Db2 admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.

Para usar consultas de acceso directo con Db2, puede utilizar la siguiente sintaxis:

```
SELECT * FROM TABLE(
```

```
system.query(  
    query => 'query string'  
))
```

El siguiente ejemplo de consulta envía una consulta a un origen de datos en Db2. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## Información sobre licencias

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

Para obtener la información más reciente sobre la versión del controlador JDBC, consulte el archivo [pom.xml](#) para el conector para Db2 en GitHub.com.

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector de Amazon Athena para IBM Db2 AS/400 (Db2 iSeries)

El conector de Amazon Athena para Db2 AS/400 permite que Amazon Athena ejecute consultas SQL en las bases de datos de IBM Db2 AS/400 (Db2 iSeries) mediante JDBC.

## Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).
- Antes de usar este conector, debe configurar una VPC y un grupo de seguridad. Para obtener más información, consulte [Creación de una VPC para un conector de origen de datos](#).

## Limitaciones

- Las operaciones de escritura de DDL no son compatibles.
- En una configuración de multiplexor, el bucket de vertido y el prefijo se comparten en todas las instancias de base de datos.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.
- Los tipos de datos de marca de fecha y hora en condiciones de filtro se deben convertir a los tipos de datos adecuados.

## Términos

Los siguientes términos hacen referencia al conector para Db2 AS/400.

- Instancia de base de datos: cualquier instancia de base de datos implementada en las instalaciones, en Amazon EC2 o en Amazon RDS.
- Controlador: un controlador de Lambda que accede a la instancia de base de datos. Un controlador puede ser para los metadatos o para los registros de datos.
- Controlador de metadatos: un controlador de Lambda que recupera los metadatos de la instancia de base de datos.
- Controlador de registros: un controlador de Lambda que recupera registros de datos de la instancia de base de datos.
- Controlador compuesto: un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.
- Propiedad o parámetro: una propiedad de base de datos que usan los controladores para extraer información de la base de datos. Estas propiedades se configuran como variables de entorno de Lambda.
- Cadena de conexión: una cadena de texto que se usa para establecer una conexión con una instancia de base de datos.
- Catálogo: un catálogo ajeno a AWS Glue registrado en Athena que es un prefijo obligatorio para la propiedad `connection_string`.
- Controlador de multiplexación: un controlador de Lambda que puede aceptar y usar varias conexiones a bases de datos.

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector para Db2 AS/400.

### Cadena de conexión

Use una cadena de conexión JDBC con el siguiente formato para conectarse a una instancia de base de datos.

```
db2as400://${jdbc_connection_string}
```

### Uso de un controlador de multiplexación

Puede usar un multiplexor para conectarse a varias instancias de base de datos con una sola función de Lambda. Las solicitudes se enrutan por nombre de catálogo. Use las siguientes clases en Lambda.

Controlador	Clase
Controlador compuesto	Db2MuxCompositeHandler
Controlador de metadatos	Db2MuxMetadataHandler
Controlador de registros	Db2MuxRecordHandler

### Parámetros del controlador de multiplexación

Parámetro	Descripción
<code>catalog_connection_string</code>	Obligatorio. Una cadena de conexión de instancia de base de datos. Agregue el prefijo a la variable de entorno con el nombre del catálogo usado en Athena. Por ejemplo, si el catálogo registrado en Athena es <code>mydb2as400catalog</code> , el nombre de la variable de entorno será <code>mydb2as400catalog_connection_string</code> .

Parámetro	Descripción
default	Obligatorio. La cadena de conexión predeterminada. Esta cadena se usa cuando el catálogo es <code>lambda:\${ AWS_LAMBDA_FUNCTION_NAME }</code> .

Las siguientes propiedades de ejemplo son para una función de Lambda de Db2 MUX que admite dos instancias de base de datos: `db2as4001` (la predeterminada) y `db2as4002`.

Propiedad	Valor
default	<code>db2as400://jdbc:as400:// &lt;ip_address&gt; ;&lt;properties&gt; ;:\${&lt;secret name&gt;};</code>
<code>db2as400_catalog1_connection_string</code>	<code>db2as400://jdbc:as400://db2as4001. hostname/ :\${ secret1_name }</code>
<code>db2as400_catalog2_connection_string</code>	<code>db2as400://jdbc:as400://db2as4002. hostname/ :\${ secret2_name }</code>
<code>db2as400_catalog3_connection_string</code>	<code>db2as400://jdbc:as400:// &lt;ip_address&gt; ;user=&lt;username&gt; ;password= &lt;password &gt; ;&lt;properties&gt; ;</code>

## Proporcionar credenciales

Para proporcionar un nombre de usuario y una contraseña para la base de datos en la cadena de conexión JDBC, puede usar las propiedades de la cadena de conexión o AWS Secrets Manager.

- Cadena de conexión: se pueden especificar un nombre de usuario y una contraseña como propiedades en la cadena de conexión JDBC.

### Important

Como práctica recomendada en materia de seguridad, no utilice credenciales codificadas en las variables de entorno ni en las cadenas de conexión. Para obtener información sobre



cómo transferir los secretos codificados a AWS Secrets Manager, consulte [Mover secretos codificados a AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

- AWS Secrets Manager: para utilizar la característica Consulta federada de Athena con AWS Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Puede poner el nombre de un secreto en AWS Secrets Manager, en la cadena de conexión JDBC. El conector reemplaza el nombre del secreto por los valores `username` y `password` de Secrets Manager.

Para las instancias de bases de datos de Amazon RDS, esta compatibilidad está estrechamente integrada. Si usa Amazon RDS, le recomendamos encarecidamente que use AWS Secrets Manager y la rotación de credenciales. Si la base de datos no usa Amazon RDS, guarde las credenciales como archivos JSON con el siguiente formato:

```
{"username": "${username}", "password": "${password}"}
```

Ejemplo de cadena de conexión con nombre secreto

La siguiente cadena tiene el nombre secreto `${secret_name}`.

```
db2as400://jdbc:as400://<ip_address>;<properties>;:${<secret_name>;};
```

El conector usa el nombre secreto para recuperar los secretos y proporcionar el nombre de usuario y la contraseña, como en el siguiente ejemplo.

```
db2as400://jdbc:as400://<ip_address>;user=<username>;password=<password>;<properties>;
```

Uso de un controlador de conexión única

Puede usar los siguientes controladores de registros y metadatos de conexión única para conectarse a una sola instancia de Db2 AS/400.

Tipo de controlador	Clase
Controlador compuesto	Db2CompositeHandler

Tipo de controlador	Clase
Controlador de metadatos	Db2MetadataHandler
Controlador de registros	Db2RecordHandler

### Parámetros de controlador de conexión única

Parámetro	Descripción
default	Obligatorio. La cadena de conexión predeterminada.

Los controladores de conexión única admiten una instancia de base de datos y deben proporcionar un parámetro de cadena de conexión default. Se omiten todas las demás cadenas de conexión.

La siguiente propiedad de ejemplo es para una sola instancia de Db2 AS/400 compatible con una función de Lambda.

Propiedad	Valor
default	db2as400://jdbc:as400:// <i>&lt;ip_address&gt;</i> ; <i>&lt;properties&gt;</i> ;: \${ <i>&lt;secret_name&gt;</i> };

### Parámetros de vertido

El SDK de Lambda puede verter datos en Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación.

Parámetro	Descripción
spill_bucket	Obligatorio. Nombre del bucket de vertido.
spill_prefix	Obligatorio. Prefijo de la clave del bucket de vertido.

Parámetro	Descripción
<code>spill_put_request_headers</code>	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud <code>putObject</code> de Amazon S3 que se usa para el vertido (por ejemplo, <code>{"x-amz-server-side-encryption" : "AES256"}</code> ). Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.

## Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos correspondientes para JDBC y Apache Arrow.

Db2 AS/400	Arrow
CHAR	VARCHAR
VARCHAR	VARCHAR
FECHA	DATEDAY
HORA	VARCHAR
MARCA DE TIEMPO	DATEMILLI
DATETIME	DATEMILLI
BOOLEAN	BOOL
SMALLINT	SMALLINT
INTEGER	INT
BIGINT	BIGINT
DECIMAL	DECIMAL

Db2 AS/400	Arrow
REAL	FLOAT8
DOBLE	FLOAT8
DECFLOAT	FLOAT8

## Particiones y divisiones

Una partición se representa mediante una o varias columnas de partición de tipo `varchar`. El conector para Db2 AS/400 crea particiones mediante los siguientes esquemas de organización.

- Distribución por hash
- Partición por intervalo
- Organización por dimensiones

El conector recupera detalles de la partición, como son el número de particiones y el nombre de la columna, de una o varias tablas de metadatos de Db2 AS/400. Las divisiones se crean en función del número de particiones identificadas.

## Rendimiento

Para mejorar el rendimiento, use la inserción de predicados para realizar consultas desde Athena, como en los siguientes ejemplos.

```
SELECT * FROM "lambda:<LAMBDA_NAME>". "<SCHEMA_NAME>". "<TABLE_NAME>"
WHERE integercol = 2147483647
```

```
SELECT * FROM "lambda: <LAMBDA_NAME>". "<SCHEMA_NAME>". "<TABLE_NAME>"
WHERE timestampcol >= TIMESTAMP '2018-03-25 07:30:58.878'
```

## Consultas de acceso directo

El conector para Db2 AS/400 admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.

Para utilizar consultas de acceso directo con Db2 AS/400, puede utilizar la siguiente sintaxis:

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'query string'  
    ))
```

El siguiente ejemplo de consulta envía una consulta a un origen de datos de Db2 AS/400. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## Información sobre licencias

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

Para obtener la información más reciente sobre la versión del controlador JDBC, consulte el archivo [pom.xml](#) para el conector para Db2 AS/400 en GitHub.com.

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector para DocumentDB de Amazon Athena

El conector de DocumentDB de Amazon Athena permite a Athena comunicarse con sus instancias de DocumentDB para que pueda consultar los datos de DocumentDB con SQL. El conector también funciona con cualquier punto de conexión compatible con MongoDB.

A diferencia de los almacenes de datos relacionales tradicionales, las colecciones de Amazon DocumentDB no tienen ningún esquema establecido. DocumentDB no tiene un almacén de metadatos. Cada entrada de una colección de DocumentDB puede tener diferentes campos y tipos de datos.

El conector de DocumentDB admite dos mecanismos para generar información de esquema de tabla: inferencia básica de esquemas y metadatos de AWS Glue Data Catalog.

La inferencia de esquemas es la opción predeterminada. Esta opción escanea una pequeña cantidad de documentos de la colección, forma una unión de todos los campos y fuerza los campos que tienen tipos de datos no superpuestos. Esta opción funciona bien para colecciones que, en su mayoría, tienen entradas uniformes.

Para las recopilaciones con una mayor variedad de tipos de datos, el conector admite la recuperación de metadatos de AWS Glue Data Catalog. Si el conector ve una base de datos y una tabla de AWS Glue que coinciden con los nombres de la colección y la base de datos de DocumentDB, obtiene la información del esquema de la tabla de AWS Glue correspondiente. Al crear la tabla de AWS Glue, le recomendamos que la convierta en un superconjunto de todos los campos a los que quiera acceder desde su colección de DocumentDB.

Si Lake Formation está habilitado en la cuenta, el rol de IAM del conector de Lambda federado de Athena que haya implementado en AWS Serverless Application Repository debe tener acceso de lectura en Lake Formation para AWS Glue Data Catalog.

### Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

### Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de DocumentDB.

- `spill_bucket`: especifica el bucket de Amazon S3 para los datos que superen los límites de la función de Lambda.
- `spill_prefix`: (opcional) de forma predeterminada, se establece una subcarpeta en la carpeta especificada `spill_bucket` llamada `athena-federation-spill`. Le recomendamos configurar un [ciclo de vida de almacenamiento](#) de Amazon S3 en esta ubicación para eliminar vertidos de más de un número predeterminado de días u horas.
- `spill_put_request_headers`: (opcional) un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud `putObject` de Amazon S3 que se usa para el vertidos (por ejemplo, `{"x-amz-server-side-encryption" : "AES256"}`). Para ver otros encabezados posibles, consulte [PutObject](#) en la referencia de la API de Amazon Simple Storage Service.

- `kms_key_id`: (opcional) de forma predeterminada, los datos que se vierten a Amazon S3 se cifran mediante el modo de cifrado autenticado AES-GCM y una clave generada aleatoriamente. Para que la función de Lambda use claves de cifrado más seguras generadas por KMS, como `a7e63k4b-81oc-40db-a2a1-4d0en2cd8331`, puede especificar un ID de clave de KMS.
- `disable_spill_encryption`: (opcional) cuando se establece en `True`, desactiva el cifrado del vertido. El valor predeterminado es `False`, de modo que los datos que se vierten a S3 se cifran mediante AES-GCM, ya sea mediante una clave generada aleatoriamente o KMS para generar claves. La desactivación del cifrado de vertido puede mejorar el rendimiento, especialmente si su ubicación de vertido usa [cifrado del servidor](#).
- `disable_glue`: (opcional) si está presente y se establece en `true` (verdadero), el conector no intentará recuperar metadatos complementarios de AWS Glue.
- `glue_catalog`: (opcional) use esta opción para especificar un [catálogo de AWS Glue entre cuentas](#). De forma predeterminada, el conector intenta obtener los metadatos de su propia cuenta de AWS Glue.
- `default_docdb`: si está presente, especifica una cadena de conexión de DocumentDB que se utilizará cuando no exista ninguna variable de entorno específica del catálogo.
- `disable_projection_and_casing`: (opcional) desactiva la proyección y la distinción entre mayúsculas y minúsculas. Utilícelo si desea consultar tablas de Amazon DocumentDB que utilizan nombres de columna que distinguen mayúsculas de minúsculas. El parámetro `disable_projection_and_casing` utiliza los siguientes valores para especificar el comportamiento de las mayúsculas y minúsculas y la asignación de columnas:
  - `false` (falso): esta es la configuración predeterminada. La proyección está habilitada y el conector espera que todos los nombres de las columnas estén en minúsculas.
  - `true` (verdadero): desactiva la proyección y la distinción entre mayúsculas y minúsculas. Al usar la el parámetro `disable_projection_and_casing`, tenga en cuenta los siguientes puntos:
    - El uso del parámetro puede dar lugar a un mayor uso del ancho de banda. Además, si su función de Lambda no está en la misma Región de AWS que su origen de datos, se generarán mayores costos por transferencia estándar entre regiones de AWS como resultado del mayor uso del ancho de banda. Para obtener más información sobre los costos por transferencia entre regiones, consulte [AWS Data Transfer Charges for Server and Serverless Architectures](#) en el blog de la Red de socios de AWS.
- Debido a que se transfiere una mayor cantidad de bytes y a que esta requiere un mayor tiempo de deserialización, la latencia general puede aumentar.

- `enable_case_insensitive_match`: (Opcional) cuando `true` realiza búsquedas sin distinción de mayúsculas y minúsculas en nombres de esquemas y tablas en Amazon DocumentDB. El valor predeterminado es `false`. Utilícelo si la consulta contiene nombres de tablas o esquemas en mayúsculas.

## Especificación de cadenas de conexión

Puede proporcionar una o más propiedades que definan los detalles de conexión de DocumentDB para las instancias de DocumentDB que utiliza con el conector. Para ello, defina una variable de entorno de Lambda que corresponda al nombre del catálogo que quiere usar en Athena. Por ejemplo, suponga que quiere usar las siguientes consultas para consultar dos instancias de DocumentDB diferentes desde Athena:

```
SELECT * FROM "docdb_instance_1".database.table
```

```
SELECT * FROM "docdb_instance_2".database.table
```

Antes de poder usar estas dos instrucciones SQL, debe agregar dos variables de entorno a la función de Lambda: `docdb_instance_1` y `docdb_instance_2`. El valor de cada una debe ser una cadena de conexión de DocumentDB con el siguiente formato:

```
mongodb://:@/?ssl=true&ssl_ca_certs=rds-combined-ca-bundle.pem&replicaSet=rs0
```

## Uso de los secretos

Si lo desea, puede usar AWS Secrets Manager para obtener parte o la totalidad del valor de los detalles de la cadena de conexión. Para utilizar la característica Consulta federada de Athena con Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Si usa la sintaxis `${my_secret}` para poner el nombre de un secreto de Secrets Manager en la cadena de conexión, el conector reemplaza `${my_secret}` por los valores de texto plano de Secrets Manager. Los secretos deben almacenarse como un secreto de texto plano con valor `<username>:<password>`. Los secretos almacenados como `{username:<username>, password:<password>}` no se pasarán correctamente a la cadena de conexión.



Los secretos también se pueden usar para toda la cadena de conexión, y el nombre de usuario y la contraseña se pueden definir dentro del secreto.

Por ejemplo, supongamos que establece la variable de entorno de Lambda para `docdb_instance_1` en el siguiente valor:

```
mongodb://${docdb_instance_1_creds}@myhostname.com:123/?ssl=true&ssl_ca_certs=rds-combined-ca-bundle.pem&replicaSet=rs0
```

El SDK de federación de consultas de Athena intenta recuperar automáticamente un secreto llamado `docdb_instance_1_creds` de Secrets Manager e inyecta ese valor en lugar de `${docdb_instance_1_creds}`. Cualquier parte de la cadena de conexión que se incluya en la combinación de caracteres `${ }` se interpreta como un secreto de Secrets Manager. Si especifica un nombre secreto que el conector no puede encontrar en Secrets Manager, el conector no reemplaza el texto.

## Configuración de bases de datos y tablas en AWS Glue

Debido a que la capacidad de inferencia de esquemas integrada del conector escanea un número limitado de documentos y solo admite un subconjunto de tipos de datos, es posible que quiera usar AWS Glue para obtener metadatos en su lugar.

Habilitar una tabla de AWS Glue para usar con Amazon DocumentDB, debe tener una base de datos y tabla de AWS Glue para la base de datos y la colección de DocumentDB para las que desea proporcionar metadatos complementarios.

Para usar una tabla de AWS Glue para metadatos complementarios

1. Al editar la tabla y la base de datos en la consola de AWS Glue, agregue la siguiente propiedad de tabla.
  - `docdb-metadata-flag`: esta propiedad indica al conector de DocumentDB que el conector puede usar la tabla para metadatos adicionales. Puede proporcionar cualquier valor para `docdb-metadata-flag` siempre y cuando la `docdb-metadata-flag` esté presente en la lista de propiedades de la tabla.
2. (Opcional) Agregue la propiedad de la tabla `sourceTable`. Esta propiedad define el nombre de la tabla de origen en Amazon DocumentDB. Use esta opción si las reglas de denominación de tablas de AWS Glue le impiden crear una tabla de AWS Glue con el mismo nombre que la tabla de Amazon DocumentDB. Por ejemplo, no se permiten mayúsculas en nombres de tablas de AWS Glue, pero se permiten en los de tablas de Amazon DocumentDB.

- (Opcional) Agregue la propiedad de tabla `columnMapping`. Esta propiedad define las asignaciones de nombres de columnas. Use esta propiedad si las reglas de denominación de columnas de AWS Glue le impiden crear una tabla de AWS Glue con los mismos nombres que la tabla de Amazon DocumentDB. Esto puede resultar útil porque se permiten mayúsculas en nombres de columnas de Amazon DocumentDB, pero no se permiten en los nombres de columnas de AWS Glue.

Se espera que el valor de la propiedad `columnMapping` sea un conjunto de asignaciones en el formato `col1=Col1,col2=Col2`.

#### Note

La asignación de columnas solo se aplica a los nombres de las columnas de nivel superior y no a los campos anidados.

Tras agregar la propiedad de la tabla `columnMapping` de AWS Glue, puede eliminar la variable de entorno de Lambda `disable_projection_and_casing`.

- Asegúrese de usar los tipos de datos adecuados para AWS Glue, como se indica en este documento.

## Compatibilidad con tipos de datos

En esta sección, se enumeran los tipos de datos que el conector de DocumentDB utiliza para la inferencia de esquemas y los tipos de datos cuando se utilizan metadatos de AWS Glue.

### Tipos de datos de inferencia de esquemas

La característica de inferencia de esquemas del conector de DocumentDB intenta inferir valores que pertenecen a uno de los siguientes tipos de datos. En la tabla se muestran los tipos de datos correspondientes para Amazon DocumentDB, Java y Apache Arrow.

Apache Arrow	Java o DocDB
VARCHAR	Cadena
INT	Entero

Apache Arrow	Java o DocDB
BIGINT	Largo
BIT	Booleano
FLOAT4	Flotante
FLOAT8	Doble
TIMESTAMPSEC	Date
VARCHAR	ObjectId
LIST	Enumeración
STRUCT	Documento

### Tipos de datos de AWS Glue

Si usa AWS Glue para obtener metadatos adicionales, puede configurar los siguientes tipos de datos. En la tabla se muestran los tipos de datos correspondientes para AWS Glue y Apache Arrow.

AWS Glue	Apache Arrow
int	INT
bigint	BIGINT
double	FLOAT8
float	FLOAT4
boolean	BIT
binario	VARBINARY
cadena	VARCHAR
Enumeración	LIST

AWS Glue	Apache Arrow
Struct	STRUCT

## Permisos necesarios

Para obtener información completa sobre las políticas de IAM que requiere este conector, consulte la sección `Policies` del archivo [athena-docdb.yaml](#). En la siguiente lista se resumen los permisos requeridos.

- **Acceso de escritura a Amazon S3:** el conector requiere acceso de escritura a una ubicación de Amazon S3 para volcar los resultados de consultas de gran tamaño.
- **Athena GetQueryExecution:** el conector usa este permiso para fallar rápidamente cuando finaliza la consulta ascendente de Athena.
- **AWS Glue Data Catalog:** el conector de DocumentDB requiere acceso de solo lectura a AWS Glue Data Catalog para obtener información sobre el esquema.
- **Registros de CloudWatch:** el conector requiere acceso a Registros de CloudWatch para almacenar registros.
- **Acceso de lectura a AWS Secrets Manager:** si decide almacenar los detalles del punto de conexión de DocumentDB en Secrets Manager, debe conceder al conector acceso a esos secretos.
- **Acceso a la VPC:** el conector requiere la capacidad de conectar y desconectar interfaces a la VPC para que pueda conectarse a ella y comunicarse con las instancias de DocumentDB.

## Rendimiento

El conector para Amazon DocumentDB de Athena no admite actualmente análisis paralelos. En cambio, intenta insertar predicados como parte de sus consultas de DocumentDB, y los predicados de los índices de la colección de DocumentDB dan lugar a una cantidad mucho menor de datos analizados.

La función de Lambda inserta proyecciones para reducir los datos analizados por la consulta. Sin embargo, a veces, la selección de un subconjunto de columnas provoca un tiempo de ejecución de consultas más prolongado. Las cláusulas `LIMIT` reducen la cantidad de datos analizados; sin embargo, si no proporciona un predicado, es probable que las consultas `SELECT` con una cláusula `LIMIT` analicen al menos 16 MB de datos.

## Consultas de acceso directo

El conector de Amazon DocumentDB de Athena admite [consultas de acceso directo](#) y está basado en NoSQL. Para obtener más información sobre consultas en Amazon DocumentDB, diríjase a [Consultas](#) en la Guía para desarrolladores de Amazon DocumentDB.

Para usar consultas de acceso directo con Amazon DocumentDB, utilice la siguiente sintaxis:

```
SELECT * FROM TABLE(  
    system.query(  
        database => 'database_name',  
        collection => 'collection_name',  
        filter => '{query_syntax}'  
    ))
```

El siguiente ejemplo consulta la base de datos `example` de la colección TPCDS y filtra todos los libros con el título `Bill of Rights`.

```
SELECT * FROM TABLE(  
    system.query(  
        database => 'example',  
        collection => 'tpcds',  
        filter => '{title: "Bill of Rights"}'  
    ))
```

## Recursos adicionales de

- Para obtener información sobre cómo usar [Amazon Athena Federated Query](#) a fin de conectar una base de datos de MongoDB a [Amazon QuickSight](#) y crear cuadros de mando y visualizaciones, consulte el artículo [Visualización de datos de MongoDB desde Amazon QuickSight mediante Amazon Athena Federated Query](#) en el Blog de macrodatos de AWS.
- Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector para DynamoDB de Amazon Athena

El conector Amazon Athena DynamoDB permite que Amazon Athena se comuniquen con DynamoDB para que pueda consultar las tablas con SQL. Las operaciones de escritura como [INSERT INTO](#) no son compatibles.

Si Lake Formation está habilitado en la cuenta, el rol de IAM del conector de Lambda federado de Athena que haya implementado en AWS Serverless Application Repository debe tener acceso de lectura en Lake Formation para AWS Glue Data Catalog.

## Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de DynamoDB.

- `spill_bucket`: especifica el bucket de Amazon S3 para los datos que superen los límites de la función de Lambda.
- `spill_prefix`: (opcional) de forma predeterminada, se establece una subcarpeta en la carpeta especificada `spill_bucket` llamada `athena-federation-spill`. Le recomendamos configurar un [ciclo de vida de almacenamiento](#) de Amazon S3 en esta ubicación para eliminar vertidos de más de un número predeterminado de días u horas.
- `spill_put_request_headers`: (opcional) un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud `putObject` de Amazon S3 que se usa para el vertido (por ejemplo, `{"x-amz-server-side-encryption" : "AES256"}`). Para ver otros encabezados posibles, consulte [PutObject](#) en la referencia de la API de Amazon Simple Storage Service.
- `kms_key_id`: (opcional) de forma predeterminada, los datos que se vierten a Amazon S3 se cifran mediante el modo de cifrado autenticado AES-GCM y una clave generada aleatoriamente. Para que la función de Lambda use claves de cifrado más seguras generadas por KMS, como `a7e63k4b-81oc-40db-a2a1-4d0en2cd8331`, puede especificar un ID de clave de KMS.
- `disable_spill_encryption`: (opcional) cuando se establece en `True`, desactiva el cifrado del vertido. El valor predeterminado es `False`, de modo que los datos que se vierten a S3 se cifran mediante AES-GCM, ya sea mediante una clave generada aleatoriamente o KMS para generar claves. La desactivación del cifrado de vertido puede mejorar el rendimiento, especialmente si su ubicación de vertido usa [cifrado del servidor](#).
- `disable_glue`: (opcional) si está presente y se establece en `true` (verdadero), el conector no intentará recuperar metadatos complementarios de AWS Glue.

- `glue_catalog`: (opcional) use esta opción para especificar un [catálogo de AWS Glue entre cuentas](#). De forma predeterminada, el conector intenta obtener los metadatos de su propia cuenta de AWS Glue.
- `disable_projection_and_casing`: (opcional) desactiva la proyección y la distinción entre mayúsculas y minúsculas. Use esta opción si quiere consultar las tablas de DynamoDB que distinguen entre mayúsculas y minúsculas en sus nombres de columna y no desea especificar ninguna propiedad `columnMapping` en la tabla de AWS Glue.

El parámetro `disable_projection_and_casing` utiliza los siguientes valores para especificar el comportamiento de las mayúsculas y minúsculas y la asignación de columnas:

- `auto`: desactiva la proyección y la distinción entre mayúsculas y minúsculas cuando se detecta un tipo no admitido con anterioridad y la asignación de nombres de columnas no se establece en la tabla. Este es el valor predeterminado.
- `always`: desactiva la proyección y la distinción entre mayúsculas y minúsculas de forma incondicional. Esta opción resulta útil cuando en los nombres de las columnas de DynamoDB se distingue entre mayúsculas y minúsculas, pero no desea especificar ninguna asignación de nombres de columna.

Al usar la el parámetro `disable_projection_and_casing`, tenga en cuenta los siguientes puntos:

- El uso del parámetro puede dar lugar a un mayor uso del ancho de banda. Además, si su función de Lambda no está en la misma Región de AWS que su origen de datos, se generarán mayores costos por transferencia estándar entre regiones de AWS como resultado del mayor uso del ancho de banda. Para obtener más información sobre los costos por transferencia entre regiones, consulte [AWS Data Transfer Charges for Server and Serverless Architectures](#) en el blog de la Red de socios de AWS.
- Debido a que se transfiere una mayor cantidad de bytes y a que esta requiere un mayor tiempo de deserialización, la latencia general puede aumentar.

## Configuración de bases de datos y tablas en AWS Glue

Debido a que la capacidad de inferencia de esquemas integrada del conector es limitada, es posible que quiera usar AWS Glue para obtener metadatos. Para ello, debe tener una base de datos y una tabla en AWS Glue. Para habilitar su uso con DynamoDB, debe editar sus propiedades.

## Para editar propiedades de base de datos en la consola de AWS Glue

1. Inicie sesión en la AWS Management Console y abra la consola de AWS Glue en <https://console.aws.amazon.com/glue/>.
2. Seleccione la pestaña Databases (Bases de datos).

En la página Databases (Bases de datos), puede editar una base de datos existente o seleccionar Add database (Agregar base de datos) para crear una.

3. En la lista de bases de datos, seleccione el enlace de la base de datos que desee editar.
4. Elija Editar.
5. En la página Update a database (Actualizar una base de datos), en Location (Ubicación), agregue la cadena **dynamo-db-flag**. Esta palabra clave indica que la base de datos contiene tablas que el conector de DynamoDB de Athena utiliza para metadatos suplementarios y se requiere para bases de datos de AWS Glue que no sean default. La propiedad dynamo-db-flag resulta útil para filtrar las bases de datos de cuentas con muchas bases de datos.

## Para editar propiedades de tabla en la consola de AWS Glue

1. Inicie sesión en la AWS Management Console y abra la consola de AWS Glue en <https://console.aws.amazon.com/glue/>.
2. Seleccione la pestaña Tablas.

En la pestaña Tablas, edite una tabla existente. Para obtener información sobre cómo agregar tablas manualmente o con un rastreador, consulte [Trabajo con tablas en la consola de AWS Glue](#) en la Guía para desarrolladores de AWS Glue.

3. En la lista de tablas, elija el vínculo de la tabla que desea editar.
4. Seleccione Actions (Acciones), Edit queue (Editar cola).
5. En la página Editar tabla, en la sección Propiedades de la tabla , agregue las siguientes propiedades de tabla según corresponda. Si utiliza el rastreador de DynamoDB de AWS Glue, estas propiedades se configuran automáticamente.
  - dynamodb: cadena que indica al conector de DynamoDB de Athena que la tabla se puede usar para metadatos adicionales. Ingrese la cadena dynamodb en las propiedades de tabla en un campo denominado classification (clasificación) (coincidencia exacta).



**Note**

La página Definir propiedades de la tabla, que forma parte del proceso de creación de la tabla en la consola AWS Glue, tiene una sección de Formato de datos con un campo de Clasificación. No se puede entrar ni elegir dynamodb aquí. En su lugar, después de crear la tabla, siga los pasos para editarla e introducir `classification` y `dynamodb` como un par clave-valor en la sección Propiedades de la tablas.

- `sourceTable`: propiedad de tabla opcional que define el nombre de la tabla de origen en DynamoDB. Use esta opción si las reglas de denominación de tablas de AWS Glue le impiden crear una tabla de AWS Glue con el mismo nombre que la de DynamoDB. Por ejemplo, no se permiten mayúsculas en nombres de tablas de AWS Glue, pero se permiten en los de tablas de DynamoDB.
- `columnMapping`: propiedad de tabla opcional que define las asignaciones de los nombres de columna. Use esta opción si las reglas de denominación de columnas de AWS Glue le impiden crear una tabla de AWS Glue con los mismos nombres de columnas que la de DynamoDB. Por ejemplo, no se permiten mayúsculas en nombres de columnas de AWS Glue, pero se permiten en los de columnas de DynamoDB. Se espera que el valor de la propiedad tenga el formato `col1=Col1, col2=Col2`. Tenga en cuenta que la asignación de columnas solo se aplica a los nombres de las columnas de nivel superior y no a los campos anidados.
- `defaultTimeZone`: propiedad de tabla opcional que se aplica a los valores `date` o `datetime` que no tienen una zona horaria explícita. Establecer este valor es una práctica recomendada para evitar discrepancias entre la zona horaria predeterminada del origen de datos y la zona horaria de la sesión de Athena.
- `datetimeFormatMapping`: propiedad de tabla opcional que especifica el formato de `date` o `datetime` que se utilizará al analizar datos de una columna del tipo de datos `date` o `timestamp` de AWS Glue. Si no se especifica esta propiedad, el conector intenta [inferir](#) un formato ISO-8601. Si el conector no puede inferir el formato de `date` o `datetime` o analizar la cadena sin procesar, el valor se omite del resultado.

El formato del valor `datetimeFormatMapping` debe ser `col1=someformat1, col2=someformat2`. A continuación, se muestran ejemplos de formatos:

```
yyyyMMdd'T'HHmmss
```

```
ddMMyyyy'T'HH:mm:ss
```

Si tu columna tiene los valores de `date` o `datetime` sin zona horaria y desea utilizar la columna de la cláusula `WHERE`, defina la propiedad `datetimeFormatMapping` de la columna.

6. Si define las columnas manualmente, asegúrese de utilizar los tipos de datos adecuados. Si usó un rastreador, valide las columnas y los tipos que este descubra.

## Permisos necesarios

Para obtener información completa sobre las políticas de IAM que requiere este conector, consulte la sección `Policies` del archivo [athena-dynamodb.yaml](#). En la siguiente lista se resumen los permisos requeridos.

- Acceso de escritura a Amazon S3: el conector requiere acceso de escritura a una ubicación de Amazon S3 para volcar los resultados de consultas de gran tamaño.
- Athena `GetQueryExecution`: el conector usa este permiso para fallar rápidamente cuando finaliza la consulta ascendente de Athena.
- AWS Glue Data Catalog: el conector de DynamoDB requiere acceso de solo lectura al AWS Glue Data Catalog para obtener información sobre el esquema.
- Registros de CloudWatch: el conector requiere acceso a Registros de CloudWatch para almacenar registros.
- Acceso de lectura de DynamoDB: el conector utiliza las operaciones de la API: `DescribeTable`, `ListSchemas`, `ListTables`, `Query` y `Scan`.

## Rendimiento

El conector de DynamoDB de Athena admite análisis paralelos e intenta insertar predicados como parte de sus consultas de DynamoDB. Un predicado de clave hash con valores distintos `X` da como resultado llamadas de consulta `X` a DynamoDB. Todos los demás escenarios de predicados dan como resultado un número `Y` de llamadas de escaneo, donde `Y` se determina heurísticamente en función del tamaño de la tabla y su rendimiento aprovisionado. Sin embargo, seleccionar un subconjunto de columnas da como resultado que en ocasiones se prolongue el tiempo de ejecución de las consultas.

Las cláusulas `LIMIT` y los predicados simples se insertan y pueden reducir la cantidad de datos analizados, lo que permite disminuir el tiempo de ejecución de las consultas.

## Cláusulas LIMIT

Una instrucción `LIMIT N` reduce los datos analizados en la consulta. Con la inserción `LIMIT N`, el conector devuelve solo las filas `N` a Athena.

## Predicados

Un predicado es una expresión de la cláusula `WHERE` de una consulta SQL que da como resultado un valor booleano y filtra las filas en función de varias condiciones. Para mejorar la funcionalidad y reducir la cantidad de datos analizados, el conector DynamoDB de Athena puede combinar estas expresiones e insertarlas directamente en DynamoDB.

Los siguientes operadores del conector DynamoDB de Athena admiten la inserción de predicados:

- Booleano: `Y`
- Igualdad: `EQUAL`, `NOT_EQUAL`, `LESS_THAN`, `LESS_THAN_OR_EQUAL`, `GREATER_THAN`, `GREATER_THAN_OR_EQUAL`, `IS_NULL`.

## Ejemplo de inserción combinada

Para mejorar las capacidades de consulta, combine los tipos de inserciones, como en el siguiente ejemplo:

```
SELECT *
FROM my_table
WHERE col_a > 10 and col_b < 10
LIMIT 10
```

Para obtener información sobre cómo usar la inserción de predicados a fin de mejorar el rendimiento en las consultas federadas, incluida DynamoDB, consulte el artículo [Improve federated queries with predicate pushdown in Amazon Athena](#) (Mejorar las consultas federadas con la inserción de predicados en Amazon Athena) en el Blog de macrodatos de AWS.

## Consultas de acceso directo

El conector de DynamoDB admite [consultas de acceso directo](#) y utiliza la sintaxis PartiQL. No se admite la operación de la API de [GetItem](#) de DynamoDB. Para obtener más información sobre las consultas en DynamoDB mediante PartiQL, consulte [PartiQL select statements for DynamoDB](#) en la Guía para desarrolladores de Amazon DynamoDB.

Para usar consultas de acceso directo con DynamoDB, utilice la siguiente sintaxis:

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'query_string'  
    ))
```

En el siguiente ejemplo de consulta de acceso directo de DynamoDB, se usa PartiQL para devolver una lista de dispositivos Fire TV Stick que tienen una propiedad DateWatched posterior al 24/12/22.

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT Devices  
                FROM WatchList  
                WHERE Devices.FireStick.DateWatched[0] > '12/24/22''  
    ))
```

## Resolución de problemas

### Varios filtros en una columna de claves de clasificación

Mensaje de error: KeyConditionExpressions solo debe contener una condición por clave

Causa: este problema puede producirse en la versión 3 del motor de Athena en consultas que tienen un filtro de límite inferior y uno superior en una columna de claves de clasificación de DynamoDB. Como DynamoDB no admite más de una condición de filtro en una clave de clasificación, se produce un error cuando el conector intenta insertar una consulta que tiene ambas condiciones aplicadas.

Solución: actualice el conector a la versión 2023.11.1 o una versión posterior. Para obtener instrucciones sobre cómo actualizar un conector, consulte [Actualización de un conector de origen de datos](#).

### Costos

Los costos de uso del conector dependen de los recursos de AWS subyacentes que se utilicen. Dado que las consultas que utilizan exploraciones pueden consumir una gran cantidad de [unidades de capacidad de lectura \(RCU\)](#), considere cuidadosamente la información de [precios de Amazon DynamoDB](#).

## Recursos adicionales de

- Para obtener una introducción sobre cómo usar el conector DynamoDB de Amazon Athena, consulte [Acceder, consultar y unirse a las tablas de Amazon DynamoDB con Athena](#) en la guía Patrones de orientación prescriptiva de AWS.
- Para obtener información sobre cómo usar el conector DynamoDB de Amazon Athena con Amazon DynamoDB, Athena y Amazon QuickSight a fin de crear un panel de control sencillo, consulte el artículo [Consultar tablas multicuentas de Amazon DynamoDB mediante Amazon Athena Federated Query](#) en el Blog de macrodatos de AWS.
- Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector Google BigQuery de Amazon Athena

El conector de Google [BigQuery](#) de Amazon Athena permite que Amazon Athena ejecute consultas de SQL en los datos de Google BigQuery.

### Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

### Limitaciones

- Las funciones de Lambda tienen un valor máximo de tiempo de espera de 15 minutos. Cada división ejecuta una consulta en BigQuery y debe finalizar con tiempo suficiente para almacenar los resultados para que Athena los lea. Si se agota el tiempo de espera de la función de Lambda, se produce un error en la consulta.
- Google BigQuery distingue entre mayúsculas y minúsculas. El conector intenta corregir las mayúsculas y minúsculas de los nombres del conjunto de datos y de las tablas, pero no corrige las mayúsculas y minúsculas de los ID del proyecto. Esto es necesario porque Athena pone en minúsculas todos los metadatos. Estas correcciones hacen muchas llamadas adicionales a Google BigQuery.
- Los tipos de datos binarios no son compatibles.

- Debido a los límites de cuota y simultaneidad de Google BigQuery, el conector puede encontrar problemas con los límites de cuota de Google. Para evitar estos problemas, aplique tantas restricciones a Google BigQuery como sea posible. Para obtener información sobre cuotas de BigQuery, consulte [Cuotas y límites](#) en la documentación de Google BigQuery.

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de Google BigQuery.

- `spill_bucket`: especifica el bucket de Amazon S3 para los datos que superen los límites de la función de Lambda.
- `spill_prefix`: (opcional) de forma predeterminada, se establece una subcarpeta en la carpeta especificada `spill_bucket` llamada `athena-federation-spill`. Le recomendamos configurar un [ciclo de vida de almacenamiento](#) de Amazon S3 en esta ubicación para eliminar vertidos de más de un número predeterminado de días u horas.
- `spill_put_request_headers`: (opcional) un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud `putObject` de Amazon S3 que se usa para el vertidos (por ejemplo, `{"x-amz-server-side-encryption" : "AES256"}`). Para ver otros encabezados posibles, consulte [PutObject](#) en la referencia de la API de Amazon Simple Storage Service.
- `kms_key_id`: (opcional) de forma predeterminada, los datos que se vierten a Amazon S3 se cifran mediante el modo de cifrado autenticado AES-GCM y una clave generada aleatoriamente. Para que la función de Lambda use claves de cifrado más seguras generadas por KMS, como `a7e63k4b-81oc-40db-a2a1-4d0en2cd8331`, puede especificar un ID de clave de KMS.
- `disable_spill_encryption`: (opcional) cuando se establece en `True`, desactiva el cifrado del vertido. El valor predeterminado es `False`, de modo que los datos que se vierten a S3 se cifran mediante AES-GCM, ya sea mediante una clave generada aleatoriamente o KMS para generar claves. La desactivación del cifrado de vertido puede mejorar el rendimiento, especialmente si su ubicación de vertido usa [cifrado del servidor](#).
- `gcp_project_id`: el ID del proyecto, no el nombre del proyecto, que contiene los conjuntos de datos de los que debe leer el conector (por ejemplo, `semiotic-primer-1234567`).
- `secret_manager_gcp_creds_name`: el nombre del secreto de AWS Secrets Manager que contiene sus credenciales de BigQuery en formato JSON (por ejemplo, `GoogleCloudPlatformCredentials`).

- `big_query_endpoint`: (opcional) la URL de un punto de conexión privado de BigQuery. Use este parámetro cuando desee acceder a BigQuery a través de un punto de conexión privado.

## Divisiones y vistas

Como el conector de BigQuery usa la API de lectura de almacenamiento de BigQuery para consultar tablas y la API de almacenamiento de BigQuery no admite vistas, el conector usa el cliente de BigQuery con una sola división de vistas.

## Rendimiento

Para consultar las tablas, el conector de BigQuery usa la API de lectura de almacenamiento de BigQuery, que utiliza un protocolo basado en RPC que proporciona un acceso rápido al almacenamiento administrado por BigQuery. Para obtener más información sobre la API de lectura de almacenamiento de BigQuery, consulte [Cómo usar la API de lectura de almacenamiento de BigQuery para leer datos de tablas](#) en la documentación de Google Cloud.

La selección de un subconjunto de columnas acelera significativamente el tiempo de ejecución de las consultas y reduce la cantidad de datos analizados. El conector está sujeto a errores de consulta a medida que aumenta la simultaneidad y, por lo general, es un conector lento.

El conector de Google BigQuery de Athena inserta predicados para reducir los datos analizados en la consulta. Las cláusulas `LIMIT`, las cláusulas `ORDER BY`, los predicados simples y las expresiones complejas se insertan en el conector para reducir la cantidad de datos analizados y disminuir el tiempo de ejecución de las consultas.

## Cláusulas `LIMIT`

Una instrucción `LIMIT N` reduce los datos analizados en la consulta. Con la inserción `LIMIT N`, el conector devuelve solo las filas `N` a Athena.

## Consultas `N` principales

Una consulta `N` principal especifica el orden del conjunto de resultados y un límite en la cantidad de filas devueltas. Puede usar este tipo de consulta para determinar los valores `N` principales máximos o `N` principales mínimos de sus conjuntos de datos. Con la inserción `N` principal, el conector devuelve solo las filas `N` ordenadas a Athena.

## Predicados

Un predicado es una expresión de la cláusula WHERE de una consulta SQL que da como resultado un valor booleano y filtra las filas en función de varias condiciones. El conector de Google BigQuery de Athena puede combinar estas expresiones e insertarlas directamente en Google BigQuery para mejorar la funcionalidad y reducir la cantidad de datos analizados.

Los siguientes operadores del conector de Google BigQuery de Athena admiten la inserción de predicados:

- Booleano: AND, OR, NOT.
- Igualdad: EQUAL, NOT\_EQUAL, LESS\_THAN, LESS\_THAN\_OR\_EQUAL, GREATER\_THAN, GREATER\_THAN\_OR\_EQUAL, IS\_DISTINCT\_FROM, NULL\_IF, IS\_NULL.
- Aritmética: ADD, SUBTRACT, MULTIPLY, DIVIDE, MODULUS, NEGATE.
- Otros: LIKE\_PATTERN, IN.

### Ejemplo de inserción combinada

Para mejorar las capacidades de consulta, combine los tipos de inserciones, como en el siguiente ejemplo:

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
ORDER BY col_a DESC
LIMIT 10;
```

### Consultas de acceso directo

El conector de Google BigQuery admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.

Para utilizar las consultas de acceso directo con Google BigQuery, puede usar la siguiente sintaxis:

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
```



```
))
```

El siguiente ejemplo de consulta envía una consulta a un origen de datos de Google BigQuery. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    )  
)
```

## Información sobre licencias

El proyecto del conector de Google BigQuery de Amazon Athena se licencia en virtud de la [Licencia Apache-2.0](#).

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector de Google Cloud Storage para Amazon Athena

El conector de Google Cloud Storage para Amazon Athena permite que Amazon Athena ejecute consultas en archivos Parquet y CSV almacenados en un bucket de Google Cloud Storage (GCS). Después de agrupar uno o varios archivos Parquet o CSV en una carpeta particionada o no particionada de un bucket de GCS, puede organizarlos en una tabla de base de datos de [AWS Glue](#).

Si Lake Formation está habilitado en la cuenta, el rol de IAM del conector de Lambda federado de Athena que haya implementado en AWS Serverless Application Repository debe tener acceso de lectura en Lake Formation para AWS Glue Data Catalog.

## Requisitos previos

- Configure una base de datos y una tabla de AWS Glue que se correspondan con el bucket y las carpetas de Google Cloud Storage. Para conocer los pasos, consulte [Configuración de bases de datos y tablas en AWS Glue](#) más adelante en este documento.

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

## Limitaciones

- Las operaciones de escritura de DDL no son compatibles.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.
- Actualmente, el conector solo admite el tipo VARCHAR para columnas de partición (`string` o `varchar` en un esquema de tabla de AWS Glue). Otros tipos de campos de partición generan errores cuando se los consulta en Athena.

## Términos

Los siguientes términos tienen relación con el conector de GCS.

- Controlador: controlador de Lambda que accede al bucket de GCS. Un controlador puede ser para los metadatos o para los registros de datos.
- Controlador de metadatos: controlador de Lambda que recupera metadatos del bucket de GCS.
- Controlador de registros: controlador de Lambda que recupera registros de datos del bucket de GCS.
- Controlador compuesto: controlador de Lambda que recupera tanto metadatos como registros de datos del bucket de GCS.

## Tipos de archivo admitidos

El conector GCS admite los tipos de archivo Parquet y CSV.

### Note

Asegúrese de no colocar tanto archivos CSV como Parquet en el mismo bucket o ruta de GCS. Si lo hace, se puede producir un error de tiempo de ejecución cuando se intenten leer archivos Parquet como CSV o viceversa.

## Parámetros

Utilice las variables de entorno de Lambda de esta sección para configurar el conector de GCS.

- `spill_bucket`: especifica el bucket de Amazon S3 para los datos que superen los límites de la función de Lambda.
- `spill_prefix`: (opcional) de forma predeterminada, se establece una subcarpeta en la carpeta especificada `spill_bucket` llamada `athena-federation-spill`. Le recomendamos configurar un [ciclo de vida de almacenamiento](#) de Amazon S3 en esta ubicación para eliminar vertidos de más de un número predeterminado de días u horas.
- `spill_put_request_headers`: (opcional) un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud `putObject` de Amazon S3 que se usa para el vertido (por ejemplo, `{"x-amz-server-side-encryption" : "AES256"}`). Para ver otros encabezados posibles, consulte [PutObject](#) en la referencia de la API de Amazon Simple Storage Service.
- `kms_key_id`: (opcional) de forma predeterminada, los datos que se vierten a Amazon S3 se cifran mediante el modo de cifrado autenticado AES-GCM y una clave generada aleatoriamente. Para que la función de Lambda use claves de cifrado más seguras generadas por KMS, como `a7e63k4b-81oc-40db-a2a1-4d0en2cd8331`, puede especificar un ID de clave de KMS.
- `disable_spill_encryption`: (opcional) cuando se establece en `True`, desactiva el cifrado del vertido. El valor predeterminado es `False`, de modo que los datos que se vierten a S3 se cifran mediante AES-GCM, ya sea mediante una clave generada aleatoriamente o KMS para generar claves. La desactivación del cifrado de vertido puede mejorar el rendimiento, especialmente si su ubicación de vertido usa [cifrado del servidor](#).
- `secret_manager_gcp_creds_name`: nombre del secreto de AWS Secrets Manager que contiene las credenciales de GCS en formato JSON (por ejemplo, `GoogleCloudPlatformCredentials`).

## Configuración de bases de datos y tablas en AWS Glue

Dado que la capacidad de inferencia de esquemas integrada del conector de GCS es limitada, se recomienda utilizar AWS Glue para los metadatos. En los siguientes procedimientos se muestra cómo crear una base de datos y una tabla en AWS Glue a las que se puede acceder desde Athena.

### Creación de una base de datos en AWS Glue

Puede utilizar la consola de AWS Glue para crear una base de datos y utilizarla con el conector de GCS.

## Para crear una base de datos en AWS Glue

1. Inicie sesión en la AWS Management Console y abra la consola de AWS Glue en <https://console.aws.amazon.com/glue/>.
2. Seleccione Databases (Bases de datos) en el panel de navegación.
3. Elija Agregar una base de datos.
4. En Name (Nombre), ingrese un nombre para la base de datos que desee utilizar con el conector de GCS.
5. En Location (Ubicación), especifique `s3://google-cloud-storage-flag`. Esta ubicación indica al conector de GCS que la base de datos de AWS Glue contiene tablas para consultar los datos de GCS en Athena. El conector reconoce las bases de datos de Athena que tienen esta marca e ignora aquellas que no la tienen.
6. Elija Crear base de datos.

## Creación de una tabla en AWS Glue

Ahora puede crear una tabla para la base de datos. Cuando se crea una tabla de AWS Glue para utilizarla con el conector de GCS, se deben especificar metadatos adicionales.

## Para crear una tabla en la consola de AWS Glue

1. En la consola de AWS Glue, seleccione Tables (Tablas) en el panel de navegación.
2. En la página Tables (Tablas), seleccione Add table (Agregar tabla).
3. En la página Set table properties (Establecer propiedades de la tabla), ingrese la siguiente información:
  - Name (Nombre): nombre exclusivo para la tabla.
  - Database (Base de datos): seleccione la base de datos de AWS Glue que haya creado para el conector de GCS.
  - Include path (Incluir ruta): en la sección Data store (Almacén de datos), en Include path (Incluir ruta), ingrese la ubicación URI de GCS con el prefijo `gs://` (por ejemplo, `gs://gcs_table/data/`). Si tiene una o varias carpetas de particiones, no las incluya en la ruta.

**Note**

Cuando se introduce la ruta de tabla no `s3://`, la consola de AWS Glue muestra un error. Puede omitir este error. La tabla se creará correctamente.

- Data format (Formato de datos): en Classification (Clasificación), seleccione CSV o Parquet.
4. Seleccione Siguiente.
  5. En la página Choose or define schema (Seleccionar o definir esquema), se recomienda encarecidamente definir un esquema de tabla, aunque no es obligatorio. Si no se define un esquema, el conector de GCS intentará inferirlo automáticamente.

Realice una de las siguientes acciones:

- Si desea que el conector de GCS intente inferir un esquema automáticamente, seleccione Next (Siguiente), y luego Create (Crear).
- Para definir un esquema por sí mismo, ejecute los pasos de la siguiente sección.

## Definición de un esquema de tabla en AWS Glue

Definir un esquema de tabla en AWS Glue requiere más pasos, pero ofrece un mayor control sobre el proceso de creación de tablas.

Para definir un esquema para una tabla en AWS Glue

1. En la página Choose or define schema (Seleccionar o definir esquema), seleccione Add (Agregar).
2. Utilice el cuadro de diálogo Add schema entry (Agregar entrada de esquema) para proporcionar un nombre de columna y un tipo de datos.
3. Para designar la columna como columna de partición, seleccione la opción Set as partition key (Establecer como clave de partición).
4. Seleccione Save (Guardar) para guardar la columna.
5. Seleccione Add (Agregar) para agregar otra columna.
6. Cuando haya terminado de agregar columnas, seleccione Next (Siguiente).
7. En la página Review and create (Revisar y crear), revise la tabla y, a continuación, seleccione Create (Crear).

- Si el esquema contiene información sobre particiones, ejecute los pasos de la siguiente sección para agregar un patrón de partición a las propiedades de la tabla en AWS Glue.

### Adición de un patrón de partición a propiedades de tabla en AWS Glue

Si los buckets de GCS tienen particiones, se debe agregar el patrón de partición a las propiedades de la tabla en AWS Glue.

Para agregar información de partición a propiedades de tabla en AWS Glue

- En la página de detalles de la tabla que haya creado en AWS Glue, seleccione Actions (Acciones), Edit table (Editar tabla).
- En la página Edit table (Editar tabla), desplácese hacia abajo hasta la sección Table properties (Propiedades de la tabla).
- Seleccione Add (Agregar) para agregar una clave de partición.
- En Clave, escriba **partition.pattern**. Esta clave define el patrón de ruta de la carpeta.
- En Value (Valor), introduzca un patrón de ruta de carpeta como **StateName=\${statename}/ZipCode=\${zipcode}/**, por ejemplo, donde **statename** y **zipcode**, que van dentro de **\${}**, son nombres de columnas de partición. El conector de GCS admite esquemas de partición Hive y no Hive.
- Cuando termine de actualizar las etiquetas, elija Guardar.
- Para ver las propiedades de tabla que acaba de crear, seleccione la pestaña Advanced properties (Propiedades avanzadas).

En este punto, puede navegar hasta la consola de Athena. La base de datos y la tabla que haya creado en AWS Glue están disponibles para realizar consultas en Athena.

### Compatibilidad con tipos de datos

En las siguientes tablas se muestran los tipos de datos admitidos para CSV y Parquet.

#### CSV

Naturaleza de los datos	Tipo de datos inferidos
Los datos parecen un número	BIGINT

Naturaleza de los datos	Tipo de datos inferidos
Los datos parecen una cadena	VARCHAR
Los datos parecen de punto flotante (flotantes, dobles o decimales)	DOBLE
Los datos parecen una fecha	Timestamp
Datos que contienen valores verdadero/falso	BOOL

## Parquet

PARQUET	Athena (Arrow)
BINARIO	VARCHAR
BOOLEAN	BOOL
DOBLE	DOBLE
ENUM	VARCHAR
FIXED_LEN _BYTE_ARRAY	DECIMAL
FLOAT	FLOAT (32 bits)
INT32	<ol style="list-style-type: none"> <li>INT32</li> <li>DATEDAY (cuando el tipo lógico de la columna de Parquet es DATE)</li> </ol>
INT64	<ol style="list-style-type: none"> <li>INT64</li> <li>TIMESTAMP (cuando el tipo lógico de la columna de Parquet es TIMESTAMP)</li> </ol>
INT96	Timestamp
MAP	MAP

PARQUET	Athena (Arrow)
STRUCT	STRUCT
LIST	LIST

## Permisos necesarios

Para obtener información completa sobre las políticas de IAM que requiere este conector, consulte la sección `Policies` del archivo [athena-gcs.yaml](#). En la siguiente lista se resumen los permisos requeridos.

- Acceso de escritura a Amazon S3: el conector requiere acceso de escritura a una ubicación de Amazon S3 para volcar los resultados de consultas de gran tamaño.
- Athena `GetQueryExecution`: el conector usa este permiso para fallar rápidamente cuando finaliza la consulta ascendente de Athena.
- AWS Glue Data Catalog: el conector de GCS requiere acceso de solo lectura a AWS Glue Data Catalog para obtener información sobre un esquema.
- Registros de CloudWatch: el conector requiere acceso a Registros de CloudWatch para almacenar registros.

## Rendimiento

Cuando el esquema de la tabla contiene campos de partición y la propiedad de tabla `partition.pattern` está configurada correctamente, se puede incluir el campo de partición en la cláusula `WHERE` de las consultas. En esas consultas, el conector de GCS utiliza las columnas de partición para ajustar la ruta de la carpeta de GCS y evitar analizar archivos innecesarios en las carpetas de GCS.

En el caso de los conjuntos de datos Parquet, seleccionar un subconjunto de columnas da como resultado que se analicen menos datos. Esto suele traducirse en un menor tiempo de ejecución de consultas cuando se aplica la proyección de columnas.

En el caso de los conjuntos de datos CSV, no se admite la proyección de columnas y no se reduce la cantidad de datos que se analizan.

Las cláusulas `LIMIT` reducen la cantidad de datos analizados; sin embargo, si no se proporciona un predicado, es probable que las consultas `SELECT` con una cláusula `LIMIT` analicen al menos 16 MB



de datos. El conector de GCS analiza más datos para los conjuntos de datos de mayor tamaño que para los conjuntos de datos más pequeños, independientemente de la cláusula LIMIT aplicada. Por ejemplo, la consulta `SELECT * LIMIT 10000` analiza más datos para un conjunto de datos subyacente de mayor tamaño que para uno más pequeño.

## Información sobre licencias

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector para HBase de Amazon Athena

El conector de Amazon Athena HBase permite a Amazon Athena comunicarse con las instancias de Apache HBase para que pueda consultar los datos de HBase con SQL.

A diferencia de los almacenes de datos relacionales tradicionales, las colecciones de HBase no tienen ningún esquema establecido. HBase no tiene un almacén de metadatos. Cada entrada de una colección de HBase puede tener diferentes campos y tipos de datos.

El conector de HBase admite dos mecanismos para generar información de esquema de la tabla: inferencia básica de esquemas y metadatos de AWS Glue Data Catalog.

La inferencia de esquemas es la opción predeterminada. Esta opción escanea una pequeña cantidad de documentos de la colección, forma una unión de todos los campos y fuerza a los campos que no tienen tipos de datos superpuestos. Esta opción funciona bien para colecciones que, en su mayoría, tienen entradas uniformes.

Para las recopilaciones con una mayor variedad de tipos de datos, el conector admite la recuperación de metadatos de AWS Glue Data Catalog. Si el conector ve una base de datos y una tabla de AWS Glue que coinciden con los nombres del espacio de nombres y de las colecciones de HBase, obtiene la información del esquema de la tabla de AWS Glue. Al crear la tabla de AWS Glue, le recomendamos que la convierta en un superconjunto de todos los campos a los que quiera acceder desde su colección de HBase.

Si Lake Formation está habilitado en la cuenta, el rol de IAM del conector de Lambda federado de Athena que haya implementado en AWS Serverless Application Repository debe tener acceso de lectura en Lake Formation para AWS Glue Data Catalog.

## Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de HBase.

- `spill_bucket`: especifica el bucket de Amazon S3 para los datos que superen los límites de la función de Lambda.
- `spill_prefix`: (opcional) de forma predeterminada, se establece una subcarpeta en la carpeta especificada `spill_bucket` llamada `athena-federation-spill`. Le recomendamos configurar un [ciclo de vida de almacenamiento](#) de Amazon S3 en esta ubicación para eliminar vertidos de más de un número predeterminado de días u horas.
- `spill_put_request_headers`: (opcional) un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud `putObject` de Amazon S3 que se usa para el vertido (por ejemplo, `{"x-amz-server-side-encryption" : "AES256"}`). Para ver otros encabezados posibles, consulte [PutObject](#) en la referencia de la API de Amazon Simple Storage Service.
- `kms_key_id`: (opcional) de forma predeterminada, los datos que se vierten a Amazon S3 se cifran mediante el modo de cifrado autenticado AES-GCM y una clave generada aleatoriamente. Para que la función de Lambda use claves de cifrado más seguras generadas por KMS, como `a7e63k4b-81oc-40db-a2a1-4d0en2cd8331`, puede especificar un ID de clave de KMS.
- `disable_spill_encryption`: (opcional) cuando se establece en `True`, desactiva el cifrado del vertido. El valor predeterminado es `False`, de modo que los datos que se vierten a S3 se cifran mediante AES-GCM, ya sea mediante una clave generada aleatoriamente o KMS para generar claves. La desactivación del cifrado de vertido puede mejorar el rendimiento, especialmente si su ubicación de vertido usa [cifrado del servidor](#).
- `disable_glue`: (opcional) si está presente y se establece en `true` (verdadero), el conector no intentará recuperar metadatos complementarios de AWS Glue.

- `glue_catalog`: (opcional) use esta opción para especificar un [catálogo de AWS Glue entre cuentas](#). De forma predeterminada, el conector intenta obtener los metadatos de su propia cuenta de AWS Glue.
- `default_hbase`: si está presente, especifica una cadena de conexión de HBase que se utilizará cuando no exista ninguna variable de entorno específica del catálogo.

## Especificación de cadenas de conexión

Puede proporcionar una o más propiedades que definan los detalles de conexión de HBase para las instancias de HBase que utiliza con el conector. Para ello, defina una variable de entorno de Lambda que corresponda al nombre del catálogo que quiere usar en Athena. Por ejemplo, suponga que quiere usar las siguientes consultas para consultar dos instancias de HBase diferentes desde Athena:

```
SELECT * FROM "hbase_instance_1".database.table
```

```
SELECT * FROM "hbase_instance_2".database.table
```

Antes de poder usar estas dos instrucciones SQL, debe agregar dos variables de entorno a la función de Lambda: `hbase_instance_1` y `hbase_instance_2`. El valor de cada una debe ser una cadena de conexión de HBase con el siguiente formato:

```
master_hostname:hbase_port:zookeeper_port
```

## Uso de los secretos

Si lo desea, puede usar AWS Secrets Manager para obtener parte o la totalidad del valor de los detalles de la cadena de conexión. Para utilizar la característica Consulta federada de Athena con Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Si usa la sintaxis `${my_secret}` para poner el nombre de un secreto de Secrets Manager en la cadena de conexión, el conector reemplaza el nombre secreto por los valores de nombre de usuario y contraseña de Secrets Manager.

Por ejemplo, supongamos que establece la variable de entorno de Lambda para `hbase_instance_1` en el siguiente valor:

```
`${hbase_host_1}:${hbase_master_port_1}:${hbase_zookeeper_port_1}
```

El SDK de federación de consultas de Athena intenta recuperar automáticamente un secreto llamado `hbase_instance_1_creds` de Secrets Manager e inyecta ese valor en lugar de ``${hbase_instance_1_creds}`. Cualquier parte de la cadena de conexión que se incluya en la combinación de caracteres ``${ }` se interpreta como un secreto de Secrets Manager. Si especifica un nombre secreto que el conector no puede encontrar en Secrets Manager, el conector no reemplaza el texto.`

## Configuración de bases de datos y tablas en AWS Glue

La inferencia de esquema integrada del conector solo admite valores que se serializan en HBase como cadenas (por ejemplo, `String.valueOf(int)`). Debido a que la capacidad de inferencia de esquemas integrada del conector es limitada, es posible que quiera usar AWS Glue como alternativa para obtener metadatos. Para habilitar una tabla de AWS Glue con la intención de usarla con HBase, debe disponer de una tabla y una base de datos de AWS Glue con nombres que coincidan con el espacio de nombres y la tabla de HBase a los que quiere proporcionar metadatos complementarios. El uso de convenciones de nomenclatura de familias de columnas de HBase es opcional y no obligatorio.

Para usar una tabla de AWS Glue para metadatos complementarios

1. Al editar la tabla y la base de datos en la consola de AWS Glue, agregue las siguientes propiedades de tablas:
  - `hbase-metadata-flag`: esta propiedad indica al conector de HBase que este puede usar la tabla para obtener metadatos adicionales. Puede proporcionar cualquier valor para `hbase-metadata-flag` siempre y cuando la `hbase-metadata-flag` esté presente en la lista de propiedades de la tabla.
  - `hbase-native-storage-flag`: utilice este indicador para alternar los dos modos de serialización de valores admitidos por el conector. De forma predeterminada, cuando este campo no está presente, el conector asume que todos los valores se almacenan en HBase como cadenas. Como tal, intentará analizar tipos de datos como INT, BIGINT y DOUBLE de HBase como cadenas. Si este campo se establece con cualquier valor de la tabla en AWS Glue, el conector cambia al modo de almacenamiento “nativo” e intenta leer INT, BIGINT, BIT y DOUBLE como bytes mediante las siguientes funciones:

```
ByteBuffer.wrap(value).getInt()
```

```
ByteBuffer.wrap(value).getLong()  
ByteBuffer.wrap(value).get()  
ByteBuffer.wrap(value).getDouble()
```

2. Asegúrese de usar los tipos de datos adecuados para AWS Glue, como se indica en este documento.

## Modelado de familias de columnas

El conector de HBase de Athena admite dos formas de modelar familias de columnas de HBase: nombres totalmente cualificados (aplanados) como `family:column`, o usar objetos STRUCT.

En el modelo STRUCT, el nombre del campo STRUCT debe coincidir con la familia de columnas y los elementos secundarios de STRUCT deben coincidir con los nombres de las columnas de la familia. Sin embargo, dado que las lecturas de columnas y de inserción de predicados aún no son totalmente compatibles con los tipos complejos como STRUCT, no se recomienda usar STRUCT en este momento.

En la siguiente imagen, se muestra una tabla configurada en AWS Glue que utiliza una combinación de los dos enfoques.

Edit table
Delete table
View properties
Compare versions
Edit schema

**Name** transactions

**Description**

**Database** hbase\_payments

**Classification** Unknown

**Location** s3://[redacted]/

**Connection**

**Deprecated** No

**Last updated** Wed Oct 23 12:30:00 GMT-400 2019

**Serde parameters** serialization.format 1

**Table properties** hbase-metadata-flag hbase-metadata-flag


Schema Showing: 1 - 13 of 13 < >

	Column name	Data type	Partition key	Comment
1	summary:order_id	string		summary family, id of the order that this transaction is for
2	summary:customer_id	bigint		summary family, id of the customer that this transaction is for
3	summary:status	string		summary family, status of the transaction
4	summary:auth	string		summary family, auth code for the transaction
5	summary:cc_id	int		summary family, last for of the credit card used for the transaction
6	summary:amount	double		summary family, the amount of the transaction
7	details:fee	double		details family, Fee the transaction network charged to process the tx
8	details:bank	string		details family, the bank baking the transaction
9	details:network	string		details family, the network that was used to clear the tx
10	details:days_payable	int		details family, the number of days this transaction will likely spend in accounts receivable
11	details:latency	int		details family, the latency (millis) of the transaction
12	details:fraud_score	int		details family, the score given to this tx by our fraud algo
13	struct_family	STRUCT		sample column family modeled as a STRUCT and containing two columns (col1, col2)

## Compatibilidad con tipos de datos

El conector recupera todos los valores de HBase como el tipo de byte básico. A continuación, en función de cómo haya definido las tablas en AWS Glue Data Catalog, asigna los valores a uno de los tipos de datos de Apache Arrow en la siguiente tabla.

Tipo de datos AWS Glue	Tipo de datos de Apache Arrow
int	INT
bigint	BIGINT
double	FLOAT8
float	FLOAT4
boolean	BIT
binario	VARBINARY
cadena	VARCHAR

 Note

Si no usa AWS Glue para complementar los metadatos, la inferencia de esquemas del conector utiliza solo los tipos de datos BIGINT, FLOAT8 y VARCHAR.

## Permisos necesarios

Para obtener información completa sobre las políticas de IAM que requiere este conector, consulte la sección `Policies` del archivo [athena-hbase.yaml](#). En la siguiente lista se resumen los permisos requeridos.

- Acceso de escritura a Amazon S3: el conector requiere acceso de escritura a una ubicación de Amazon S3 para volcar los resultados de consultas de gran tamaño.
- Athena GetQueryExecution: el conector usa este permiso para fallar rápidamente cuando finaliza la consulta ascendente de Athena.
- AWS Glue Data Catalog: el conector de HBase requiere acceso de solo lectura a AWS Glue Data Catalog para obtener información sobre el esquema.
- Registros de CloudWatch: el conector requiere acceso a Registros de CloudWatch para almacenar registros.

- Acceso de lectura de AWS Secrets Manager: si decide almacenar los detalles del punto de conexión de HBase en Secrets Manager, debe conceder al conector acceso a esos secretos.
- Acceso a la VPC: el conector requiere la capacidad de conectar y desconectar interfaces a la VPC para que pueda conectarse a ella y comunicarse con las instancias de HBase.

## Rendimiento

El conector de HBase de Athena intenta paralelizar las consultas con la instancia de HBase al leer cada servidor de región en paralelo. El conector HBase de Athena inserta predicados para reducir los datos analizados en la consulta.

La función de Lambda también realiza la inserción de proyecciones para reducir los datos analizados por la consulta. Sin embargo, a veces, la selección de un subconjunto de columnas provoca un tiempo de ejecución de consultas más prolongado. Las cláusulas LIMIT reducen la cantidad de datos analizados; sin embargo, si no proporciona un predicado, es probable que las consultas SELECT con una cláusula LIMIT analicen al menos 16 MB de datos.

HBase es propenso a errores en las consultas y a tiempos de ejecución de consulta variables. Es probable que tenga que realizar las consultas varias veces para que se ejecuten correctamente. El conector para HBase resiste las limitaciones debidas a la simultaneidad.

## Consultas de acceso directo

El conector HBase admite [consultas de acceso directo](#) y está basado en NoSQL. Para obtener información sobre cómo consultar Apache HBase, visite [Consultas en HBase](#) en la documentación de Apache.

Para utilizar consultas de acceso directo con HBase, utilice la siguiente sintaxis:

```
SELECT * FROM TABLE(  
    system.query(  
        database => 'database_name',  
        collection => 'collection_name',  
        filter => '{query_syntax}'  
    ))
```

En el siguiente ejemplo, se filtran las consultas de acceso directo de HBase para los empleados de 24 o 30 años de la colección `employee` de la base de datos `default`.

```
SELECT * FROM TABLE(  

```



```
system.query(  
    DATABASE => 'default',  
    COLLECTION => 'employee',  
    FILTER => 'SingleColumnValueFilter('personaldata', 'age', =,  
    'binary:30')' ||  
    ' OR SingleColumnValueFilter('personaldata', 'age', =,  
    'binary:24')'  
    ))
```

## Información sobre licencias

El proyecto del conector de HBase de Amazon Athena tiene una [Licencia Apache-2.0](#).

## Recursos adicionales de

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector Hortonworks de Amazon Athena

El conector Hortonworks de Amazon Athena permite que Amazon Athena ejecute consultas SQL en la plataforma de datos de Cloudera [Hortonworks](#). El conector transforma las consultas SQL de Athena en la sintaxis HiveQL equivalente.

## Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

## Limitaciones

- Las operaciones de escritura de DDL no son compatibles.
- En una configuración de multiplexor, el bucket de vertido y el prefijo se comparten en todas las instancias de base de datos.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.

## Términos

Los siguientes términos hacen referencia al conector de Hortonworks Hive.

- **Instancia de base de datos:** cualquier instancia de base de datos implementada en las instalaciones, en Amazon EC2 o en Amazon RDS.
- **Controlador:** un controlador de Lambda que accede a la instancia de base de datos. Un controlador puede ser para los metadatos o para los registros de datos.
- **Controlador de metadatos:** un controlador de Lambda que recupera los metadatos de la instancia de base de datos.
- **Controlador de registros:** un controlador de Lambda que recupera registros de datos de la instancia de base de datos.
- **Controlador compuesto:** un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.
- **Propiedad o parámetro:** una propiedad de base de datos que usan los controladores para extraer información de la base de datos. Estas propiedades se configuran como variables de entorno de Lambda.
- **Cadena de conexión:** una cadena de texto que se usa para establecer una conexión con una instancia de base de datos.
- **Catálogo:** un catálogo ajeno a AWS Glue registrado en Athena que es un prefijo obligatorio para la propiedad `connection_string`.
- **Controlador de multiplexación:** un controlador de Lambda que puede aceptar y usar varias conexiones a bases de datos.

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de Hortonworks Hive.

### Cadena de conexión

Use una cadena de conexión JDBC con el siguiente formato para conectarse a una instancia de base de datos.

```
hive://${jdbc_connection_string}
```

## Uso de un controlador de multiplexación

Puede usar un multiplexor para conectarse a varias instancias de base de datos con una sola función de Lambda. Las solicitudes se enrutan por nombre de catálogo. Use las siguientes clases en Lambda.

Controlador	Clase
Controlador compuesto	HiveMuxCompositeHandler
Controlador de metadatos	HiveMuxMetadataHandler
Controlador de registros	HiveMuxRecordHandler

## Parámetros del controlador de multiplexación

Parámetro	Descripción
<code><i>\$catalog_connection_string</i></code>	Obligatorio. Una cadena de conexión de instancia de base de datos. Agregue el prefijo a la variable de entorno con el nombre del catálogo usado en Athena. Por ejemplo, si el catálogo registrado en Athena es <code>myhivecatalog</code> , el nombre de la variable de entorno será <code>myhivecatalog_connection_string</code> .
<code>default</code>	Obligatorio. La cadena de conexión predeterminada. Esta cadena se usa cuando el catálogo es <code>lambda:\${AWS_LAMBDA_FUNCTION_NAME}</code> .

Las siguientes propiedades de ejemplo son para una función de Lambda de Hive MUX que admite dos instancias de base de datos: `hive1` (la predeterminada) y `hive2`.

Propiedad	Valor
<code>default</code>	<code>hive://jdbc:hive2://hive1:10000/default?\${Test/RDS/hive1}</code>

Propiedad	Valor
hive_catalog1_connection_string	hive://jdbc:hive2://hive1:10000/default?\${Test/RDS/hive1}
hive_catalog2_connection_string	hive://jdbc:hive2://hive2:10000/default?UID=sample&PWD=sample

## Proporcionar credenciales

Para proporcionar un nombre de usuario y una contraseña para la base de datos en la cadena de conexión JDBC, puede usar las propiedades de la cadena de conexión o AWS Secrets Manager.

- Cadena de conexión: se pueden especificar un nombre de usuario y una contraseña como propiedades en la cadena de conexión JDBC.

### Important

Como práctica recomendada en materia de seguridad, no utilice credenciales codificadas en las variables de entorno ni en las cadenas de conexión. Para obtener información sobre cómo transferir los secretos codificados a AWS Secrets Manager, consulte [Mover secretos codificados a AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

- AWS Secrets Manager: para utilizar la característica Consulta federada de Athena con AWS Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Puede poner el nombre de un secreto en AWS Secrets Manager, en la cadena de conexión JDBC. El conector reemplaza el nombre del secreto por los valores `username` y `password` de Secrets Manager.

Para las instancias de bases de datos de Amazon RDS, esta compatibilidad está estrechamente integrada. Si usa Amazon RDS, le recomendamos encarecidamente que use AWS Secrets Manager y la rotación de credenciales. Si la base de datos no usa Amazon RDS, guarde las credenciales como archivos JSON con el siguiente formato:

```
{"username": "${username}", "password": "${password}"}
```

## Ejemplo de cadena de conexión con nombre secreto

La siguiente cadena tiene el nombre secreto `${Test/RDS/hive1host}`.

```
hive://jdbc:hive2://hive1host:10000/default?...&${Test/RDS/hive1host}&...
```

El conector usa el nombre secreto para recuperar los secretos y proporcionar el nombre de usuario y la contraseña, como en el siguiente ejemplo.

```
hive://jdbc:hive2://hive1host:10000/default?...&UID=sample2&PWD=sample2&...
```

Actualmente, el conector de Hortonworks Hive reconoce las propiedades UID y PWD de JDBC.

## Uso de un controlador de conexión única

Puede usar los siguientes controladores de registros y metadatos de conexión única para conectarse a una sola instancia de Hortonworks Hive.

Tipo de controlador	Clase
Controlador compuesto	<code>HiveCompositeHandler</code>
Controlador de metadatos	<code>HiveMetadataHandler</code>
Controlador de registros	<code>HiveRecordHandler</code>

## Parámetros de controlador de conexión única

Parámetro	Descripción
<code>default</code>	Obligatorio. La cadena de conexión predeterminada.

Los controladores de conexión única admiten una instancia de base de datos y deben proporcionar un parámetro de cadena de conexión `default`. Se omiten todas las demás cadenas de conexión.

La siguiente propiedad de ejemplo es para una sola instancia de Hortonworks Hive compatible con una función de Lambda.

Propiedad	Valor
default	hive://jdbc:hive2://hive1host:10000/default?secret=\${Test/RDS/hive1host}

### Parámetros de vertido

El SDK de Lambda puede verter datos en Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación.

Parámetro	Descripción
spill_bucket	Obligatorio. Nombre del bucket de vertido.
spill_prefix	Obligatorio. Prefijo de la clave del bucket de vertido.
spill_put_request_headers	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud <code>putObject</code> de Amazon S3 que se usa para el vertido (por ejemplo, <code>{"x-amz-server-side-encryption" : "AES256"}</code> ). Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.

### Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos correspondientes para JDBC, Hortonworks Hive y Arrow.

JDBC	Hortonworks Hive	Arrow
Booleano	Booleano	Bit
Entero	TINYINT	Pequeño

JDBC	Hortonworks Hive	Arrow
Short	SMALLINT	Smallint
Entero	INT	Int
Largo	BIGINT	Bigint
float	float4	Float4
Doble	float8	Float8
Date	date	DateDay
Timestamp	Marca de tiempo	DateMilli
Cadena	VARCHAR	Varchar
Bytes	bytes	Varbinary
BigDecimal	Decimal	Decimal
ARRAY	N/D (ver nota)	Enumeración

### Note

Actualmente, Hortonworks Hive no admite los tipos agregados ARRAY, MAP, STRUCT o UNIONTYPE. Las columnas de tipos agregados se tratan como columnas VARCHAR en SQL.

## Particiones y divisiones

Las particiones se usan para determinar cómo generar divisiones para el conector. Athena crea una columna sintética de tipo `varchar` que representa el esquema de partición de la tabla para ayudar al conector a generar divisiones. El conector no modifica la definición de la tabla real.

## Rendimiento

Hortonworks Hive admite particiones estáticas. El conector Hortonworks Hive de Athena puede recuperar datos de estas particiones en paralelo. Si quiere consultar conjuntos de datos muy grandes

con una distribución uniforme de particiones, se recomienda encarecidamente la partición estática. La selección de un subconjunto de columnas acelera significativamente el tiempo de ejecución de las consultas y reduce la cantidad de datos analizados. El conector de Hortonworks Hive resiste las limitaciones debidas a la simultaneidad.

El conector Hortonworks Hive de Athena inserta predicados para reducir los datos analizados en la consulta. Los predicados simples, las expresiones complejas y las cláusulas LIMIT se insertan en el conector para reducir la cantidad de datos analizados y disminuir el tiempo de ejecución de las consultas.

## Cláusulas LIMIT

Una instrucción LIMIT N reduce los datos analizados en la consulta. Con la inserción LIMIT N, el conector devuelve solo las filas N a Athena.

## Predicados

Un predicado es una expresión de la cláusula WHERE de una consulta SQL que da como resultado un valor booleano y filtra las filas en función de varias condiciones. El conector Hortonworks Hive de Athena puede combinar estas expresiones e insertarlas directamente en Hortonworks Hive para mejorar la funcionalidad y reducir la cantidad de datos analizados.

Los siguientes operadores del conector Hortonworks Hive de Athena admiten la inserción de predicados:

- Booleano: AND, OR, NOT.
- Igualdad: EQUAL, NOT\_EQUAL, LESS\_THAN, LESS\_THAN\_OR\_EQUAL, GREATER\_THAN, GREATER\_THAN\_OR\_EQUAL, IS\_NULL.
- Aritmética: ADD, SUBTRACT, MULTIPLY, DIVIDE, MODULUS, NEGATE.
- Otros: LIKE\_PATTERN, IN.

## Ejemplo de inserción combinada

Para mejorar las capacidades de consulta, combine los tipos de inserciones, como en el siguiente ejemplo:

```
SELECT *  
FROM my_table
```



```
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
LIMIT 10;
```

## Consultas de acceso directo

El conector de Hortonworks Hive admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.

Para utilizar consultas de acceso directo con Hortonworks Hive, puede usar la siguiente sintaxis:

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  )
)
```

El siguiente ejemplo de consulta envía una consulta a un origen de datos de Hortonworks Hive. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.

```
SELECT * FROM TABLE(
  system.query(
    query => 'SELECT * FROM customer LIMIT 10'
  )
)
```

## Información sobre licencias

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

Para obtener la información más reciente sobre la versión del controlador JDBC, consulte el archivo [pom.xml](#) para el conector de Hortonworks Hive en GitHub.com.

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector Apache Kafka de Amazon Athena

El conector de Amazon Athena para Apache Kafka permite que Amazon Athena ejecute consultas SQL en los temas de Apache Kafka. Use este conector para ver los temas de [Apache Kafka](#) como tablas y los mensajes como filas en Athena.

### Requisitos previos

Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

### Limitaciones

- Las operaciones de escritura de DDL no son compatibles.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.
- Los tipos de datos de marca de fecha y hora en condiciones de filtro se deben convertir a los tipos de datos adecuados.
- Los tipos de datos de fecha y hora no son compatibles con el tipo de archivo CSV y se tratan como valores varchar.
- No se admite la asignación a campos JSON anidados. El conector solo asigna campos de nivel superior.
- El conector no admite tipos complejos. Los tipos complejos se interpretan como cadenas.
- Para extraer valores JSON complejos o trabajar con ellos, use las funciones relacionadas con JSON disponibles en Athena. Para obtener más información, consulte [Extracción de datos de JSON](#).
- El conector no permite el acceso a metadatos de mensajes de Kafka.

### Términos

- Controlador de metadatos: un controlador de Lambda que recupera los metadatos de la instancia de base de datos.
- Controlador de registros: un controlador de Lambda que recupera registros de datos de la instancia de base de datos.

- Controlador compuesto: un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.
- Punto de conexión de Kafka: una cadena de texto que establece una conexión con una instancia de Kafka.

## Compatibilidad con clústeres

El conector Kafka se puede usar con los siguientes tipos de clústeres.

- Kafka independiente: una conexión directa con Kafka (autenticada o no autenticada).
- Confluent: una conexión directa con Confluent Kafka. Para obtener información sobre el uso de Athena con datos de Confluent Kafka, consulte [Visualize Confluent data in Amazon QuickSight using Amazon Athena](#) (Visualización de datos de Confluent en Amazon QuickSight con Amazon Athena) en el blog sobre inteligencia empresarial de AWS.

## Conexión a Confluent

Para la conexión a Confluent, se deben seguir los siguientes pasos:

1. Genere una clave de API desde Confluent.
2. Guarde el nombre de usuario y la contraseña de la clave de API de Confluent en AWS Secrets Manager.
3. Proporcione el nombre secreto de la variable de entorno `secrets_manager_secret` en el conector Kafka.
4. Siga los pasos de la sección [Configuración del conector Kafka](#) de este documento.

## Métodos de autenticación compatibles

El conector admite los siguientes métodos de autenticación.

- [SSL](#)
- [SASL/SCRAM](#)
- SASL/PLAIN
- SASL/PLAINTEXT
- NO\_AUTH
- Kafka autogestionado y Confluent Platform: SSL, SASL/SCRAM, SASL/PLAINTEXT, NO\_AUTH.

- Kafka autogestionado y Confluent Cloud: SASL/PLAIN.

Para obtener más información, consulte [Configuración de la autenticación del conector Kafka de Athena](#).

Formatos compatibles de datos de entrada

El conector admite los siguientes formatos de datos de entrada.

- JSON
- CSV

Parámetros

Use las variables de entorno de Lambda mencionadas en esta sección para configurar el conector Kafka de Athena.

- `auth_type`: especifica el tipo de autenticación del clúster. El conector admite los siguientes tipos de autenticación:
  - `NO_AUTH`: conexión directa a Kafka (por ejemplo, a un clúster de Kafka implementado en una instancia de EC2 que no usa autenticación).
  - `SASL_SSL_PLAIN`: este método usa el protocolo de seguridad SASL\_SSL y el mecanismo PLAIN SASL. Para obtener más información, consulte [Configuración de SASL](#) en la documentación de Apache Kafka.
  - `SASL_PLAINTEXT_PLAIN`: este método usa el protocolo de seguridad SASL\_PLAINTEXT y el mecanismo PLAIN SASL. Para obtener más información, consulte [Configuración de SASL](#) en la documentación de Apache Kafka.
  - `SASL_SSL_SCRAM_SHA512`: puede usar este tipo de autenticación para controlar el acceso a los clústeres de Apache Kafka. Este método almacena el nombre de usuario y la contraseña en AWS Secrets Manager. El secreto debe estar asociado al clúster de Kafka. Para obtener más información, consulte [Autenticación mediante SASL/SCRAM](#) en la documentación de Apache Kafka.
  - `SASL_PLAINTEXT_SCRAM_SHA512`: este método usa el protocolo de seguridad SASL\_PLAINTEXT y el mecanismo SCRAM\_SHA512 SASL. Este método usa el nombre de usuario y la contraseña almacenados en AWS Secrets Manager. Para obtener más información, consulte la sección [Configuración de SASL](#) en la documentación de Apache Kafka.

- **SSL:** la autenticación SSL utiliza archivos del almacén de claves y el almacén de confianza para conectarse con el clúster de Apache Kafka. Debe generar los archivos del almacén de confianza y el almacén de claves, cargarlos en un bucket de Amazon S3 y proporcionar la referencia a Amazon S3 cuando implemente el conector. El almacén de claves, el almacén de confianza y la clave SSL se almacenan en el AWS Secrets Manager. El cliente debe proporcionar la clave secreta de AWS cuando se implemente el conector. Para obtener más información, consulte [Cifrado y autenticación mediante SSL](#) en la documentación de Apache Kafka.

Para obtener más información, consulte [Configuración de la autenticación del conector Kafka de Athena](#).

- **certificates\_s3\_reference:** la ubicación de Amazon S3 que contiene los certificados (los archivos del almacén de claves y del almacén de confianza).
- **disable\_spill\_encryption:** (opcional) cuando se establece en `True`, desactiva el cifrado del vertido. El valor predeterminado es `False`, de modo que los datos que se vierten a S3 se cifran mediante AES-GCM, ya sea mediante una clave generada aleatoriamente o KMS para generar claves. La desactivación del cifrado de vertido puede mejorar el rendimiento, especialmente si su ubicación de vertido usa [cifrado del servidor](#).
- **kafka\_endpoint:** los detalles del punto de conexión que se van a proporcionar a Kafka.
- **secrets\_manager\_secret:** el nombre del secreto de AWS en el que se guardan las credenciales.
- **Parámetros de vertido:** las funciones de Lambda almacenan (“vierten”) temporalmente los datos que no caben en la memoria de Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación. Use los parámetros de la siguiente tabla para especificar la ubicación de vertido.

Parámetro	Descripción
<code>spill_bucket</code>	Obligatorio. El nombre del bucket de Amazon S3 en el que la función de Lambda puede verter datos.
<code>spill_prefix</code>	Obligatorio. El prefijo del bucket de vertido donde la función de Lambda puede verter datos.
<code>spill_put_request_headers</code>	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud <code>putObject</code> de Amazon S3 que se usa para el vertido (por ejemplo, <code>{"x-amz-server-side-encryption" : "AES256"}</code> ).

Parámetro	Descripción
	Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.
	<ul style="list-style-type: none"> <li>ID de subred: uno o más ID de subred correspondientes a la subred que la función de Lambda puede usar para acceder al origen de datos.</li> <li>Clúster público de Kafka o clúster de Confluent Cloud estándar: asocie el conector a una subred privada que tenga una puerta de enlace NAT.</li> <li>Clúster de Confluent Cloud con conectividad privada: asocie el conector a una subred privada que tenga una ruta al clúster de Confluent Cloud. <ul style="list-style-type: none"> <li>En el caso de <a href="#">AWS Transit Gateway</a>, las subredes deben estar en una VPC conectada a la misma puerta de enlace de tránsito que utiliza Confluent Cloud.</li> <li>Para el <a href="#">emparejamiento de VPC</a>, las subredes deben estar en una VPC que esté sincronizada con la VPC de Confluent Cloud.</li> <li>Para <a href="#">AWS PrivateLink</a>, las subredes deben estar en una VPC que tenga una ruta a los puntos de conexión de VPC que se conectan a Confluent Cloud.</li> </ul> </li> </ul>

#### Note

Si se implementa el conector en una VPC con el fin de acceder a recursos privados y también se desea conectar a un servicio de acceso público como Confluent, se debe asociar el conector a una subred privada que tenga una puerta de enlace NAT. Para obtener más información, consulte [Puerta de enlace NAT](#) en la Guía del usuario de Amazon VPC.

## Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos correspondientes compatibles con Kafka y Apache Arrow.

Kafka	Arrow
CHAR	VARCHAR
VARCHAR	VARCHAR

Kafka	Arrow
MARCA DE TIEMPO	MILLISECOND
FECHA	DAY
BOOLEAN	BOOL
SMALLINT	SMALLINT
INTEGER	INT
BIGINT	BIGINT
DECIMAL	FLOAT8
DOBLE	FLOAT8

## Particiones y divisiones

Los temas de Kafka se dividen en particiones. Cada partición está ordenada. Cada mensaje de una partición tiene un ID incremental denominado desplazamiento. Cada partición de Kafka se divide a su vez en múltiples divisiones para su procesamiento en paralelo. Los datos están disponibles durante el periodo de retención configurado en los clústeres de Kafka.

## Prácticas recomendadas

Como práctica recomendada, use la inserción de predicados cuando realice consultas en Athena, como en los siguientes ejemplos.

```
SELECT *
FROM "kafka_catalog_name"."glue_schema_registry_name"."glue_schema_name"
WHERE integercol = 2147483647
```

```
SELECT *
FROM "kafka_catalog_name"."glue_schema_registry_name"."glue_schema_name"
WHERE timestampcol >= TIMESTAMP '2018-03-25 07:30:58.878'
```

## Configuración del conector Kafka

Antes de poder usar el conector, debe configurar el clúster de Apache Kafka, usar [AWS Glue Schema Registry](#) para definir el esquema y configurar la autenticación del conector.

Cuando trabaje con AWS Glue Schema Registry, tenga en cuenta los siguientes puntos:

- Asegúrese de que el texto del campo Description (Descripción) de AWS Glue Schema Registry incluya la cadena {AthenaFederationKafka}. Esta cadena de marcadores es obligatoria para los registros de AWS Glue que use con el conector Kafka de Amazon Athena.
- Para obtener el mejor rendimiento, use solo minúsculas en los nombres de las bases de datos y tablas. El uso combinado de mayúsculas y minúsculas hace que el conector realice una búsqueda que no distinga mayúsculas de minúsculas, lo que requiere un mayor esfuerzo computacional.

Para configurar el entorno de Apache Kafka y AWS Glue Schema Registry

1. Configure su entorno de Apache Kafka.
2. Cargue el archivo de descripción del tema de Kafka (es decir, su esquema) en formato JSON en AWS Glue Schema Registry. Para obtener más información, consulte [Integración con AWS Glue Schema Registry](#) en la Guía para desarrolladores de AWS Glue. Para obtener más esquemas de ejemplo, consulte la siguiente sección.

### Ejemplos de esquemas para AWS Glue Schema Registry

Use el formato de los ejemplos de esta sección cuando cargue el esquema en [AWS Glue Schema Registry](#).

#### Ejemplo de esquema de tipo JSON

En el siguiente ejemplo, el esquema que se va a crear en AWS Glue Schema Registry especifica json como el valor de dataFormat y usa datatypejson para topicName.

#### Note

El valor de topicName debe usar la misma distinción entre mayúsculas y minúsculas que el nombre del tema en Kafka.

```
{
```



```
"topicName": "datatypejson",
"message": {
  "dataFormat": "json",
  "fields": [
    {
      "name": "intcol",
      "mapping": "intcol",
      "type": "INTEGER"
    },
    {
      "name": "varcharcol",
      "mapping": "varcharcol",
      "type": "VARCHAR"
    },
    {
      "name": "booleancol",
      "mapping": "booleancol",
      "type": "BOOLEAN"
    },
    {
      "name": "bigintcol",
      "mapping": "bigintcol",
      "type": "BIGINT"
    },
    {
      "name": "doublecol",
      "mapping": "doublecol",
      "type": "DOUBLE"
    },
    {
      "name": "smallintcol",
      "mapping": "smallintcol",
      "type": "SMALLINT"
    },
    {
      "name": "tinyintcol",
      "mapping": "tinyintcol",
      "type": "TINYINT"
    },
    {
      "name": "datecol",
      "mapping": "datecol",
      "type": "DATE",
      "formatHint": "yyyy-MM-dd"
    }
  ]
}
```

```
    },
    {
      "name": "timestampcol",
      "mapping": "timestampcol",
      "type": "TIMESTAMP",
      "formatHint": "yyyy-MM-dd HH:mm:ss.SSS"
    }
  ]
}
}
```

### Ejemplo de esquema de tipo CSV

En el siguiente ejemplo, el esquema que se va a crear en AWS Glue Schema Registry especifica csv como el valor de dataFormat y usa datatypecsvbulk para topicName. El valor de topicName debe usar la misma distinción entre mayúsculas y minúsculas que el nombre del tema en Kafka.

```
{
  "topicName": "datatypecsvbulk",
  "message": {
    "dataFormat": "csv",
    "fields": [
      {
        "name": "intcol",
        "type": "INTEGER",
        "mapping": "0"
      },
      {
        "name": "varcharcol",
        "type": "VARCHAR",
        "mapping": "1"
      },
      {
        "name": "booleancol",
        "type": "BOOLEAN",
        "mapping": "2"
      },
      {
        "name": "bigintcol",
        "type": "BIGINT",
        "mapping": "3"
      },
    ],
  }
}
```

```

    {
      "name": "doublecol",
      "type": "DOUBLE",
      "mapping": "4"
    },
    {
      "name": "smallintcol",
      "type": "SMALLINT",
      "mapping": "5"
    },
    {
      "name": "tinyintcol",
      "type": "TINYINT",
      "mapping": "6"
    },
    {
      "name": "floatcol",
      "type": "DOUBLE",
      "mapping": "7"
    }
  ]
}
}

```

## Configuración de la autenticación del conector Kafka de Athena

Puede usar diversos métodos para autenticarse en su clúster de Apache Kafka, entre los que se incluyen SSL, SASL/SCRAM, SASL/PLAIN y SASL/PLAINTEXT.

En la siguiente tabla, se muestran los tipos de autenticación del conector, el protocolo de seguridad y el mecanismo SASL de cada uno. Para obtener más información, consulte la sección [Seguridad](#) de la documentación de Apache Kafka.

auth_type	security.protocol	sasl.mechanism	Compatibilidad de tipos de clúster
SASL_SSL_PLAIN	SASL_SSL	PLAIN	<ul style="list-style-type: none"> <li>Kafka autogestionado</li> <li>Confluent Platform</li> <li>Confluent Cloud</li> </ul>

auth_type	security.protocol	sasl.mechanism	Compatibilidad de tipos de clúster
SASL_PLAINTEXT_PLAIN	SASL_PLAINTEXT	PLAIN	<ul style="list-style-type: none"> <li>• Kafka autogestionado</li> <li>• Confluent Platform</li> </ul>
SASL_SSL_SCRAM_SHA512	SASL_SSL	SCRAM-SHA-512	<ul style="list-style-type: none"> <li>• Kafka autogestionado</li> <li>• Confluent Platform</li> </ul>
SASL_PLAINTEXT_SCRAM_SHA512	SASL_PLAINTEXT	SCRAM-SHA-512	<ul style="list-style-type: none"> <li>• Kafka autogestionado</li> <li>• Confluent Platform</li> </ul>
SSL	SSL	N/A	<ul style="list-style-type: none"> <li>• Kafka autogestionado</li> <li>• Confluent Platform</li> </ul>

## SSL

Si el clúster se ha autenticado con SSL, debe generar los archivos del almacén de confianza y del almacén de claves y cargarlos en el bucket de Amazon S3. Debe proporcionar esta referencia de Amazon S3 al implementar el conector. El almacén de claves, el almacén de confianza y la clave SSL se almacenan en el AWS Secrets Manager. Debe proporcionar la clave secreta de AWS al implementar el conector.

Para obtener información sobre cómo crear un secreto en Secrets Manager, consulte [Creación de un secreto de AWS Secrets Manager](#).

Para usar este tipo de autenticación, establezca las variables de entorno como se muestra en la siguiente tabla.

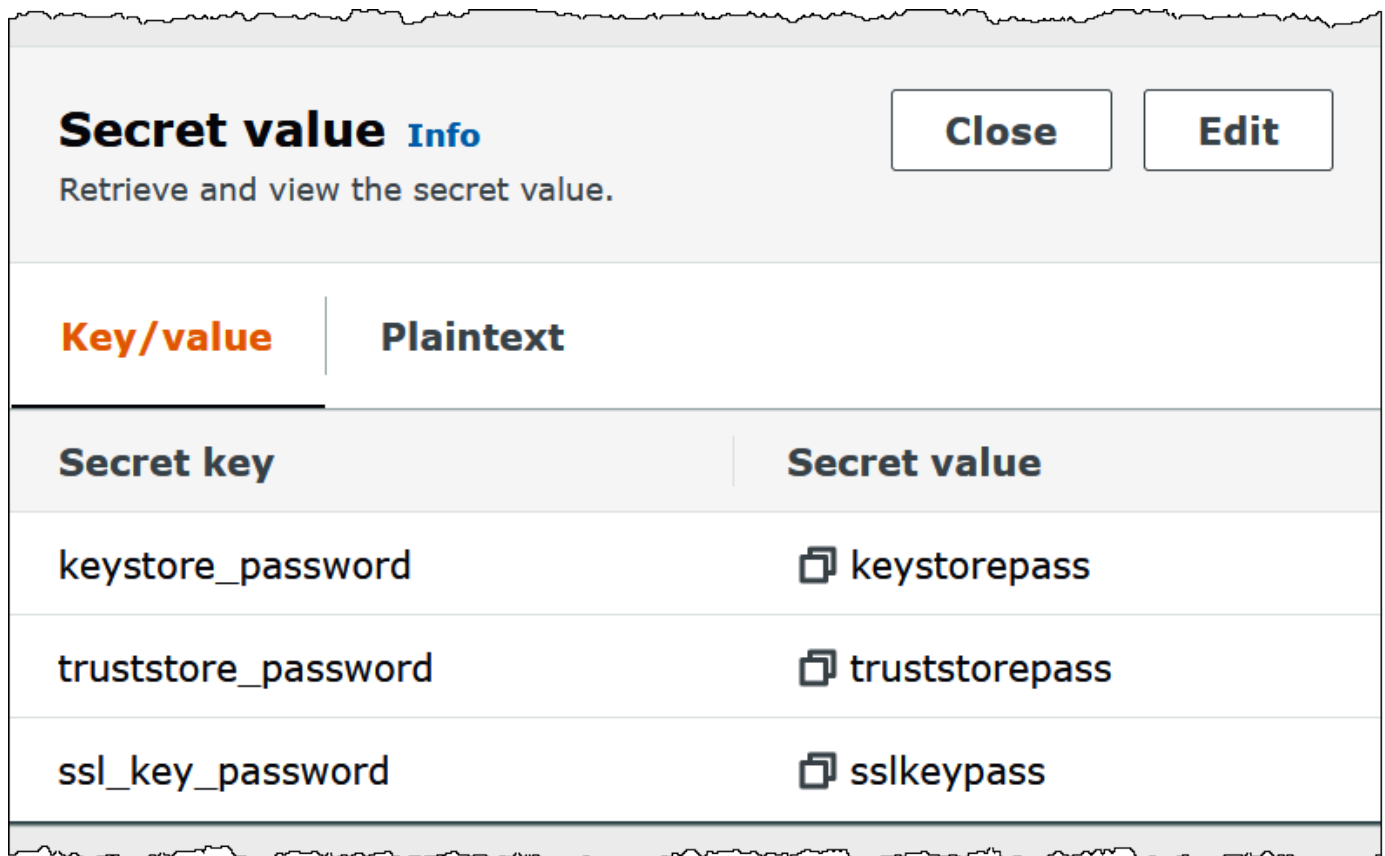
Parámetro	Valor
auth_type	SSL
certificates_s3_reference	La ubicación de Amazon S3 que contiene los certificados.
secrets_manager_secret	El nombre de su clave secreta de AWS.

Después de crear un secreto en Secrets Manager, puede verlo en la consola de Secrets Manager.

Para ver el secreto en Secrets Manager

1. Abra la consola de Secrets Manager en <https://console.aws.amazon.com/secretsmanager/>.
2. En el panel de navegación, elija Secretos.
3. En la página Secretos, elija el vínculo al secreto.
4. En la página de detalles del secreto, elija Retrieve secret value (Recuperar valor del secreto).

La siguiente imagen muestra un secreto de ejemplo con tres pares de clave y valor: keystore\_password, truststore\_password y ssl\_key\_password.



Para obtener más información sobre cómo usar SSL con Kafka, consulte [Cifrado y autenticación mediante SSL](#) en la documentación de Apache Kafka.

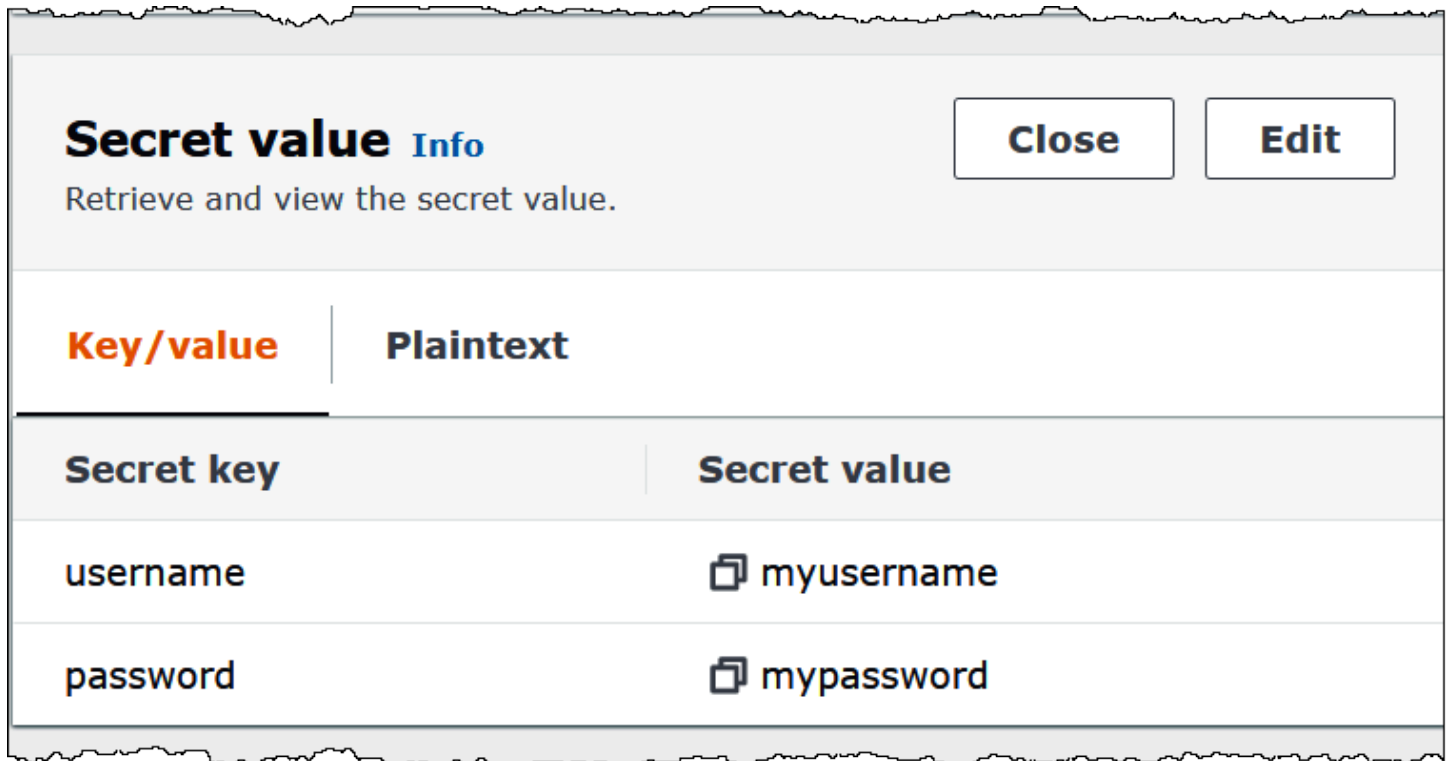
## SASL/SCRAM

Si su clúster usa la autenticación con SCRAM, proporcione la clave de Secrets Manager que está asociada al clúster cuando implemente el conector. Las credenciales de AWS del usuario (clave secreta y clave de acceso) se usan para autenticarse en el clúster.

Establezca las variables de entorno como se muestra en la siguiente tabla.

Parámetro	Valor
auth_type	SASL_SSL_SCRAM_SHA512
secrets_manager_secret	El nombre de su clave secreta de AWS.

La siguiente imagen muestra un secreto de ejemplo en la consola de Secrets Manager con dos pares de clave y valor: uno para username y otro para password.



Para obtener más información sobre cómo usar SASL/SCRAM con Kafka, consulte [Autenticación mediante SASL/SCRAM](#) en la documentación de Apache Kafka.

## Información sobre licencias

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector para MSK de Amazon Athena

El conector de Amazon Athena para [Amazon MSK](#) permite que Amazon Athena ejecute consultas SQL en los temas de Apache Kafka. Use este conector para ver los temas de [Apache Kafka](#) como tablas y los mensajes como filas en Athena. Para obtener información adicional, consulte [Analizar datos de streaming en tiempo real en Amazon MSK con Amazon Athena](#) en el Blog de macrodatos de AWS.

## Requisitos previos

Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

## Limitaciones

- Las operaciones de escritura de DDL no son compatibles.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.
- Los tipos de datos de marca de fecha y hora en condiciones de filtro se deben convertir a los tipos de datos adecuados.
- Los tipos de datos de fecha y hora no son compatibles con el tipo de archivo CSV y se tratan como valores varchar.
- No se admite la asignación a campos JSON anidados. El conector solo asigna campos de nivel superior.
- El conector no admite tipos complejos. Los tipos complejos se interpretan como cadenas.

- Para extraer valores JSON complejos o trabajar con ellos, use las funciones relacionadas con JSON disponibles en Athena. Para obtener más información, consulte [Extracción de datos de JSON](#).
- El conector no permite el acceso a metadatos de mensajes de Kafka.

## Términos

- Controlador de metadatos: un controlador de Lambda que recupera los metadatos de la instancia de base de datos.
- Controlador de registros: un controlador de Lambda que recupera registros de datos de la instancia de base de datos.
- Controlador compuesto: un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.
- Punto de conexión de Kafka: una cadena de texto que establece una conexión con una instancia de Kafka.

## Compatibilidad con clústeres

El conector para MSK se puede usar con los siguientes tipos de clústeres.

- Clúster provisionado por MSK: usted especifica, supervisa y escala manualmente la capacidad del clúster.
- Clúster sin servidor de MSK: proporciona capacidad bajo demanda que se escala automáticamente en correspondencia con la E/S de la aplicación.
- Kafka independiente: una conexión directa con Kafka (autenticada o no autenticada).

## Métodos de autenticación compatibles

El conector admite los siguientes métodos de autenticación.

- [SASL/IAM](#)
- [SSL](#)
- [SASL/SCRAM](#)
- SASL/PLAIN
- SASL/PLAINTEXT



- NO\_AUTH

Para obtener más información, consulte [Configuración de la autenticación del conector de Athena MSK](#).

## Formatos compatibles de datos de entrada

El conector admite los siguientes formatos de datos de entrada.

- JSON
- CSV

## Parámetros

Use las variables de entorno de Lambda mencionadas en esta sección para configurar el conector para MSK de Athena.

- `auth_type`: especifica el tipo de autenticación del clúster. El conector admite los siguientes tipos de autenticación:
  - `NO_AUTH`: conexión directa a Kafka sin autenticación (por ejemplo, a un clúster de Kafka implementado en una instancia de EC2 que no usa autenticación).
  - `SASL_SSL_PLAIN`: este método usa el protocolo de seguridad SASL\_SSL y el mecanismo PLAIN SASL.
  - `SASL_PLAINTEXT_PLAIN`: este método usa el protocolo de seguridad SASL\_PLAINTEXT y el mecanismo PLAIN SASL.

### Note

Apache Kafka admite los tipos de autenticación `SASL_SSL_PLAIN` y `SASL_PLAINTEXT_PLAIN`; sin embargo, Amazon MSK no los admite.

- `SASL_SSL_AWS_MSK_IAM`: el control de acceso de IAM para Amazon MSK le permite gestionar tanto la autenticación como la autorización de su clúster de MSK. Las credenciales de AWS de usuario (clave secreta y clave de acceso) se usan para conectarse al clúster. Para obtener más información, consulte [Control de acceso de IAM](#) en la Guía para desarrolladores de Amazon Managed Streaming para Apache Kafka.

- `SASL_SSL_SCRAM_SHA512`: puede usar este tipo de autenticación para controlar el acceso a los clústeres de Amazon MSK. Este método almacena el nombre de usuario y la contraseña en AWS Secrets Manager. El secreto debe estar asociado al clúster de Amazon MSK. Para obtener más información, consulte [Configurar la autenticación SASL/SCRAM para un clúster de Amazon MKS](#) en la Guía para desarrolladores de Amazon Managed Streaming para Apache Kafka.
- `SSL`: la autenticación SSL utiliza archivos del almacén de claves y el almacén de confianza para conectarse con el clúster de Amazon MSK. Debe generar los archivos del almacén de confianza y el almacén de claves, cargarlos en un bucket de Amazon S3 y proporcionar la referencia a Amazon S3 cuando implemente el conector. El almacén de claves, el almacén de confianza y la clave SSL se almacenan en el AWS Secrets Manager. El cliente debe proporcionar la clave secreta de AWS cuando se implemente el conector. Para obtener más información, consulte [Autenticación TLS mutua](#) en la Guía para desarrolladores de Amazon Managed Streaming para Apache Kafka.

Para obtener más información, consulte [Configuración de la autenticación del conector de Athena MSK](#).

- `certificates_s3_reference`: la ubicación de Amazon S3 que contiene los certificados (los archivos del almacén de claves y del almacén de confianza).
- `disable_spill_encryption`: (opcional) cuando se establece en `True`, desactiva el cifrado del vertido. El valor predeterminado es `False`, de modo que los datos que se vierten a S3 se cifran mediante AES-GCM, ya sea mediante una clave generada aleatoriamente o KMS para generar claves. La desactivación del cifrado de vertido puede mejorar el rendimiento, especialmente si su ubicación de vertido usa [cifrado del servidor](#).
- `kafka_endpoint`: los detalles del punto de conexión que se van a proporcionar a Kafka. Por ejemplo, para un clúster de Amazon MSK, debe proporcionar una [URL de arranque](#) para el clúster.
- `secrets_manager_secret`: el nombre del secreto de AWS en el que se guardan las credenciales. Este parámetro no es obligatorio para la autenticación de IAM.
- **Parámetros de vertido**: las funciones de Lambda almacenan (“vierten”) temporalmente los datos que no caben en la memoria de Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación. Use los parámetros de la siguiente tabla para especificar la ubicación de vertido.

Parámetro	Descripción
<code>spill_bucket</code>	Obligatorio. El nombre del bucket de Amazon S3 en el que la función de Lambda puede verter datos.
<code>spill_prefix</code>	Obligatorio. El prefijo del bucket de vertido donde la función de Lambda puede verter datos.
<code>spill_put_request_headers</code>	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud <code>putObject</code> de Amazon S3 que se usa para el vertido (por ejemplo, <code>{"x-amz-server-side-encryption" : "AES256"}</code> ). Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.

### Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos correspondientes compatibles con Kafka y Apache Arrow.

Kafka	Arrow
CHAR	VARCHAR
VARCHAR	VARCHAR
MARCA DE TIEMPO	MILLISECOND
FECHA	DAY
BOOLEAN	BOOL
SMALLINT	SMALLINT
INTEGER	INT
BIGINT	BIGINT

Kafka	Arrow
DECIMAL	FLOAT8
DOBLE	FLOAT8

## Particiones y divisiones

Los temas de Kafka se dividen en particiones. Cada partición está ordenada. Cada mensaje de una partición tiene un ID incremental denominado desplazamiento. Cada partición de Kafka se divide a su vez en múltiples divisiones para su procesamiento en paralelo. Los datos están disponibles durante el periodo de retención configurado en los clústeres de Kafka.

## Prácticas recomendadas

Como práctica recomendada, use la inserción de predicados cuando realice consultas en Athena, como en los siguientes ejemplos.

```
SELECT *
FROM "msk_catalog_name"."glue_schema_registry_name"."glue_schema_name"
WHERE integercol = 2147483647
```

```
SELECT *
FROM "msk_catalog_name"."glue_schema_registry_name"."glue_schema_name"
WHERE timestampcol >= TIMESTAMP '2018-03-25 07:30:58.878'
```

## Configuración del conector para MSK

Antes de poder usar el conector, debe configurar el clúster de Amazon MSK, usar [AWS Glue Schema Registry](#) para definir el esquema y configurar la autenticación del conector.

### Note

Si se implementa el conector en una VPC con el fin de acceder a recursos privados y también se desea conectar a un servicio de acceso público como Confluent, se debe asociar el conector a una subred privada que tenga una puerta de enlace NAT. Para obtener más información, consulte [Puerta de enlace NAT](#) en la Guía del usuario de Amazon VPC.

Cuando trabaje con AWS Glue Schema Registry, tenga en cuenta los siguientes puntos:

- Asegúrese de que el texto del campo Description (Descripción) de AWS Glue Schema Registry incluya la cadena {AthenaFederationMSK}. Esta cadena de marcadores es obligatoria para los registros AWS Glue que use con el conector para MSK de Amazon Athena.
- Para obtener el mejor rendimiento, use solo minúsculas en los nombres de las bases de datos y tablas. El uso combinado de mayúsculas y minúsculas hace que el conector realice una búsqueda que no distinga mayúsculas de minúsculas, lo que requiere un mayor esfuerzo computacional.

Para configurar el entorno de Amazon MSK y AWS Glue Schema Registry


1. Configure el entorno de Amazon MSK. Para obtener más información y conocer los pasos que se deben seguir, consulte [Configuración de Amazon MSK](#) y [Empezar a utilizar Amazon MSK](#) en la Guía para desarrolladores de Amazon Managed Streaming para Apache Kafka.
2. Cargue el archivo de descripción del tema de Kafka (es decir, su esquema) en formato JSON en AWS Glue Schema Registry. Para obtener más información, consulte [Integración con AWS Glue Schema Registry](#) en la Guía para desarrolladores de AWS Glue. Para obtener más esquemas de ejemplo, consulte la siguiente sección.

Ejemplos de esquemas para AWS Glue Schema Registry

Use el formato de los ejemplos de esta sección cuando cargue el esquema en [AWS Glue Schema Registry](#).

Ejemplo de esquema de tipo JSON

En el siguiente ejemplo, el esquema que se va a crear en AWS Glue Schema Registry especifica json como el valor de dataFormat y usa datatypejson para topicName.

 Note

El valor de topicName debe usar la misma distinción entre mayúsculas y minúsculas que el nombre del tema en Kafka.

```
{
  "topicName": "datatypejson",
  "message": {
```

```
"dataFormat": "json",
"fields": [
  {
    "name": "intcol",
    "mapping": "intcol",
    "type": "INTEGER"
  },
  {
    "name": "varcharcol",
    "mapping": "varcharcol",
    "type": "VARCHAR"
  },
  {
    "name": "booleancol",
    "mapping": "booleancol",
    "type": "BOOLEAN"
  },
  {
    "name": "bigintcol",
    "mapping": "bigintcol",
    "type": "BIGINT"
  },
  {
    "name": "doublecol",
    "mapping": "doublecol",
    "type": "DOUBLE"
  },
  {
    "name": "smallintcol",
    "mapping": "smallintcol",
    "type": "SMALLINT"
  },
  {
    "name": "tinyintcol",
    "mapping": "tinyintcol",
    "type": "TINYINT"
  },
  {
    "name": "datecol",
    "mapping": "datecol",
    "type": "DATE",
    "formatHint": "yyyy-MM-dd"
  },
  {
```

```
        "name": "timestampcol",
        "mapping": "timestampcol",
        "type": "TIMESTAMP",
        "formatHint": "yyyy-MM-dd HH:mm:ss.SSS"
    }
]
}
}
```

## Ejemplo de esquema de tipo CSV

En el siguiente ejemplo, el esquema que se va a crear en AWS Glue Schema Registry especifica `csv` como el valor de `dataFormat` y usa `datatypecsvbulk` para `topicName`. El valor de `topicName` debe usar la misma distinción entre mayúsculas y minúsculas que el nombre del tema en Kafka.

```
{
  "topicName": "datatypecsvbulk",
  "message": {
    "dataFormat": "csv",
    "fields": [
      {
        "name": "intcol",
        "type": "INTEGER",
        "mapping": "0"
      },
      {
        "name": "varcharcol",
        "type": "VARCHAR",
        "mapping": "1"
      },
      {
        "name": "booleancol",
        "type": "BOOLEAN",
        "mapping": "2"
      },
      {
        "name": "bigintcol",
        "type": "BIGINT",
        "mapping": "3"
      },
      {
        "name": "doublecol",
```

```

    "type": "DOUBLE",
    "mapping": "4"
  },
  {
    "name": "smallintcol",
    "type": "SMALLINT",
    "mapping": "5"
  },
  {
    "name": "tinyintcol",
    "type": "TINYINT",
    "mapping": "6"
  },
  {
    "name": "floatcol",
    "type": "DOUBLE",
    "mapping": "7"
  }
]
}
}

```

## Configuración de la autenticación del conector de Athena MSK


Puede usar diversos métodos para autenticarse en su clúster de Amazon MSK, entre los que se incluyen IAM, SSL, SCRAM y Kafka independiente.

En la siguiente tabla, se muestran los tipos de autenticación del conector, el protocolo de seguridad y el mecanismo SASL de cada uno. Para obtener más información, consulte [Autenticación y autorización para las API de Apache Kafka](#) en la Guía para desarrolladores de Amazon Managed Streaming para Apache Kafka.

auth_type	security.protocol	sasl.mechanism
SASL_SSL_PLAIN	SASL_SSL	PLAIN
SASL_PLAINTEXT_PLAIN	SASL_PLAINTEXT	PLAIN
SASL_SSL_AWS_MSK_IAM	SASL_SSL	AWS_MSK_IAM
SASL_SSL_SCRAM_SHA512	SASL_SSL	SCRAM-SHA-512



auth_type	security.protocol	sasl.mechanism
SSL	SSL	N/A

 Note

Apache Kafka admite los tipos de autenticación SASL\_SSL\_PLAIN y SASL\_PLAINTEXT\_PLAIN; sin embargo, Amazon MSK no los admite.

## SASL/IAM

Si el clúster usa la autenticación de IAM, debe configurar la política de IAM para el usuario al configurar el clúster. Para obtener más información, consulte [Control de acceso de IAM](#) en la Guía para desarrolladores de Amazon Managed Streaming para Apache Kafka.

Para usar este tipo de autenticación, establezca la variable de entorno auth\_type de Lambda del conector como SASL\_SSL\_AWS\_MSK\_IAM.

## SSL

Si el clúster se ha autenticado con SSL, debe generar los archivos del almacén de confianza y del almacén de claves y cargarlos en el bucket de Amazon S3. Debe proporcionar esta referencia de Amazon S3 al implementar el conector. El almacén de claves, el almacén de confianza y la clave SSL se almacenan en el AWS Secrets Manager. Debe proporcionar la clave secreta de AWS al implementar el conector.

Para obtener información sobre cómo crear un secreto en Secrets Manager, consulte [Creación de un secreto de AWS Secrets Manager](#).

Para usar este tipo de autenticación, establezca las variables de entorno como se muestra en la siguiente tabla.

Parámetro	Valor
auth_type	SSL
certificates_s3_reference	La ubicación de Amazon S3 que contiene los certificados.

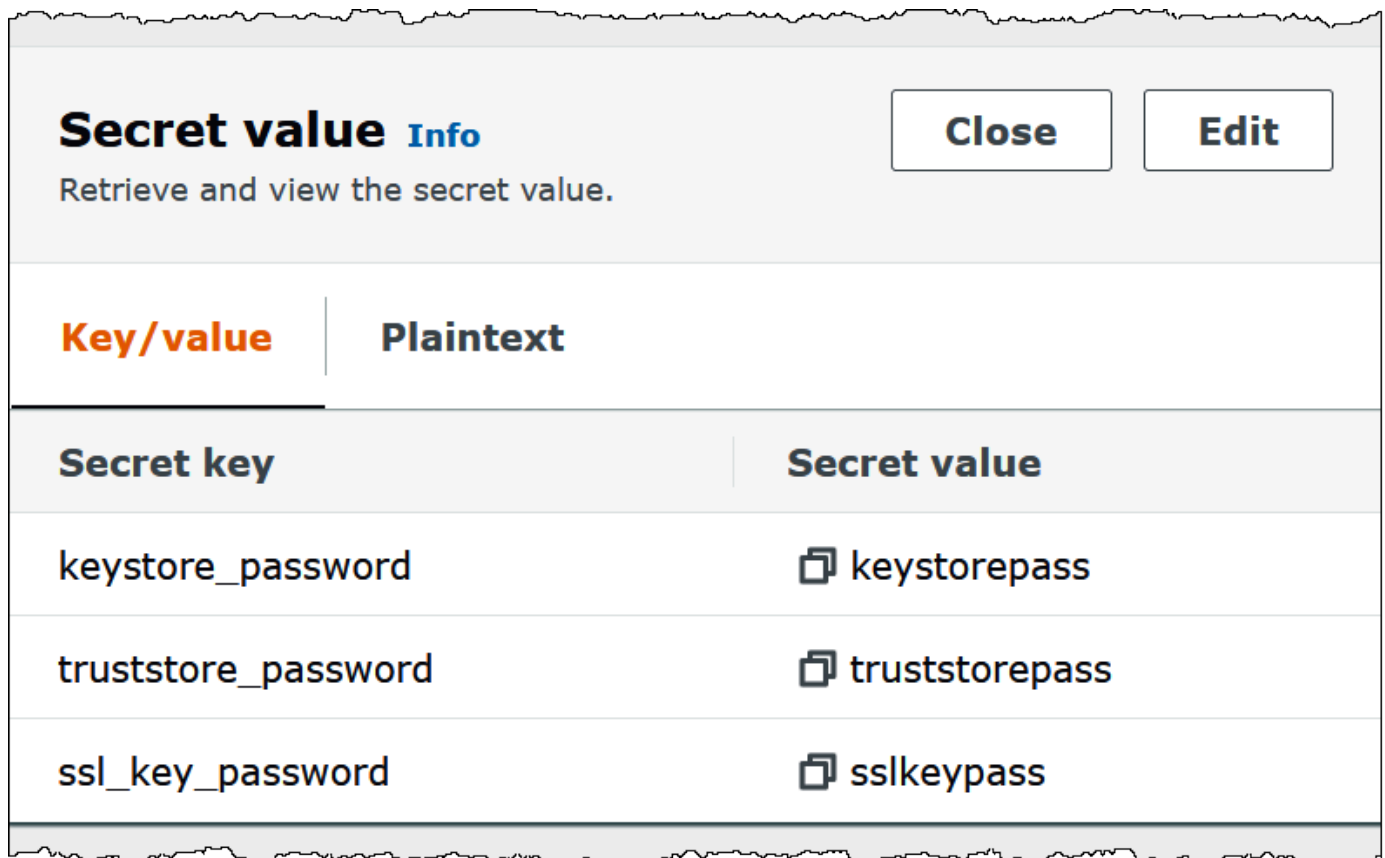
Parámetro	Valor
secrets_manager_secret	El nombre de su clave secreta de AWS.

Después de crear un secreto en Secrets Manager, puede verlo en la consola de Secrets Manager.

Para ver el secreto en Secrets Manager

1. Abra la consola de Secrets Manager en <https://console.aws.amazon.com/secretsmanager/>.
2. En el panel de navegación, elija Secretos.
3. En la página Secretos, elija el vínculo al secreto.
4. En la página de detalles del secreto, elija Retrieve secret value (Recuperar valor del secreto).

La siguiente imagen muestra un secreto de ejemplo con tres pares de clave y valor: keystore\_password, truststore\_password y ssl\_key\_password.



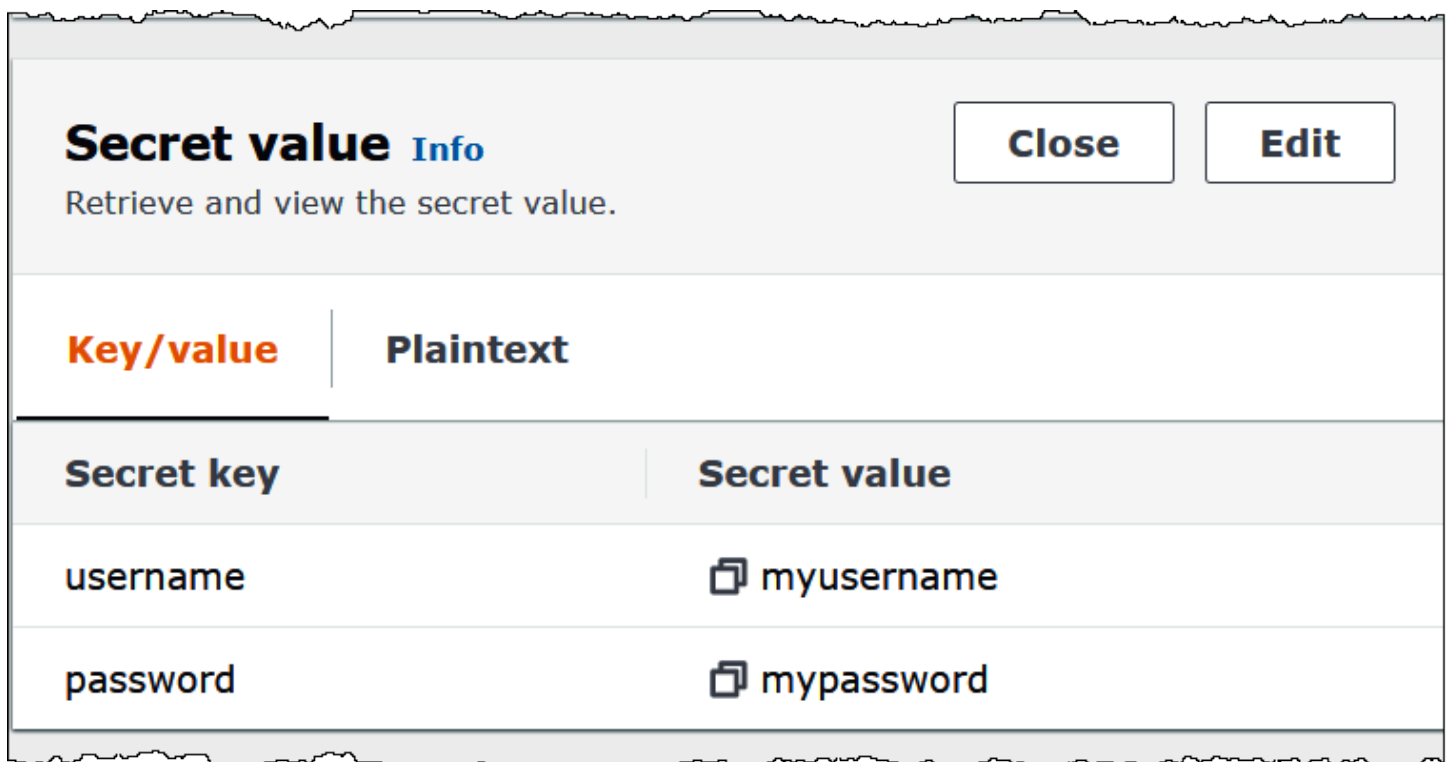
## SASL/SCRAM

Si su clúster usa la autenticación con SCRAM, proporcione la clave de Secrets Manager que está asociada al clúster cuando implemente el conector. Las credenciales de AWS del usuario (clave secreta y clave de acceso) se usan para autenticarse en el clúster.

Establezca las variables de entorno como se muestra en la siguiente tabla.

Parámetro	Valor
auth_type	SASL_SSL_SCRAM_SHA512
secrets_manager_secret	El nombre de su clave secreta de AWS.

La siguiente imagen muestra un secreto de ejemplo en la consola de Secrets Manager con dos pares de clave y valor: uno para username y otro para password.



### Información sobre licencias

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

### Conector para MySQL de Amazon Athena

El conector Lambda MySQL de Amazon Athena permite a Amazon Athena acceder a las bases de datos de MySQL.

### Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).
- Antes de usar este conector, debe configurar una VPC y un grupo de seguridad. Para obtener más información, consulte [Creación de una VPC para un conector de origen de datos](#).

### Limitaciones

- Las operaciones de escritura de DDL no son compatibles.
- En una configuración de multiplexor, el bucket de vertido y el prefijo se comparten en todas las instancias de base de datos.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.
- Como Athena convierte las consultas a minúsculas, los nombres de las tablas de MySQL deben estar en minúsculas. Por ejemplo, las consultas de Athena en una tabla llamada `myTable` producirán errores.

### Términos

Los siguientes términos hacen referencia al conector de MySQL.

- Instancia de base de datos: cualquier instancia de base de datos implementada en las instalaciones, en Amazon EC2 o en Amazon RDS.
- Controlador: un controlador de Lambda que accede a la instancia de base de datos. Un controlador puede ser para los metadatos o para los registros de datos.

- Controlador de metadatos: un controlador de Lambda que recupera los metadatos de la instancia de base de datos.
- Controlador de registros: un controlador de Lambda que recupera registros de datos de la instancia de base de datos.
- Controlador compuesto: un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.
- Propiedad o parámetro: una propiedad de base de datos que usan los controladores para extraer información de la base de datos. Estas propiedades se configuran como variables de entorno de Lambda.
- Cadena de conexión: una cadena de texto que se usa para establecer una conexión con una instancia de base de datos.
- Catálogo: un catálogo ajeno a AWS Glue registrado en Athena que es un prefijo obligatorio para la propiedad `connection_string`.
- Controlador de multiplexación: un controlador de Lambda que puede aceptar y usar varias conexiones a bases de datos.

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de MySQL.

## Cadena de conexión

Use una cadena de conexión JDBC con el siguiente formato para conectarse a una instancia de base de datos.

```
mysql://${jdbc_connection_string}
```

### Note

Si recibe el error `java.sql.SQLException: Zero date value prohibited (java.sql.SQLException: valor de fecha cero prohibido)` al hacer una consulta `SELECT` en una tabla de MySQL, agregue el siguiente parámetro a la cadena de conexión:

```
zeroDateTimeBehavior=convertToNull
```

Para obtener más información, consulte [Error “Valor de fecha cero prohibido” al intentar seleccionar de la tabla de MySQL](#) en GitHub.com.

## Uso de un controlador de multiplexación

Puede usar un multiplexor para conectarse a varias instancias de base de datos con una sola función de Lambda. Las solicitudes se enrutan por nombre de catálogo. Use las siguientes clases en Lambda.

Controlador	Clase
Controlador compuesto	<code>MySQLMuxCompositeHandler</code>
Controlador de metadatos	<code>MySQLMuxMetadataHandler</code>
Controlador de registros	<code>MySQLMuxRecordHandler</code>

## Parámetros del controlador de multiplexación

Parámetro	Descripción
<code><i>\$catalog_connection_string</i></code>	Obligatorio. Una cadena de conexión de instancia de base de datos. Agregue el prefijo a la variable de entorno con el nombre del catálogo usado en Athena. Por ejemplo, si el catálogo registrado en Athena es <code>mymysqlcatalog</code> , el nombre de la variable de entorno será <code>mymysqlcatalog_connection_string</code> .
<code>default</code>	Obligatorio. La cadena de conexión predeterminada. Esta cadena se usa cuando el catálogo es <code>lambda:\${AWS_LAMBDA_FUNCTION_NAME}</code> .

Las siguientes propiedades de ejemplo son para una función de Lambda de MySQL MUX que admite dos instancias de base de datos: `mysql1` (la predeterminada) y `mysql2`.

Propiedad	Valor
default	mysql://jdbc:mysql://mysql2 .host:3333/default? user=samp le2&password=sample2
mysql_catalog1_connection_string	mysql://jdbc:mysql://mysql1 .host:3306/default?\${Test/RDS/ MySql1}
mysql_catalog2_connection_string	mysql://jdbc:mysql://mysql2 .host:3333/default? user=samp le2&password=sample2

## Proporcionar credenciales

Para proporcionar un nombre de usuario y una contraseña para la base de datos en la cadena de conexión JDBC, puede usar las propiedades de la cadena de conexión o AWS Secrets Manager.

- Cadena de conexión: se pueden especificar un nombre de usuario y una contraseña como propiedades en la cadena de conexión JDBC.

### Important

Como práctica recomendada en materia de seguridad, no utilice credenciales codificadas en las variables de entorno ni en las cadenas de conexión. Para obtener información sobre cómo transferir los secretos codificados a AWS Secrets Manager, consulte [Mover secretos codificados a AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

- AWS Secrets Manager: para utilizar la característica Consulta federada de Athena con AWS Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Puede poner el nombre de un secreto en AWS Secrets Manager, en la cadena de conexión JDBC. El conector reemplaza el nombre del secreto por los valores `username` y `password` de Secrets Manager.

Para las instancias de bases de datos de Amazon RDS, esta compatibilidad está estrechamente integrada. Si usa Amazon RDS, le recomendamos encarecidamente que use AWS Secrets Manager y la rotación de credenciales. Si la base de datos no usa Amazon RDS, guarde las credenciales como archivos JSON con el siguiente formato:

```
{"username": "${username}", "password": "${password}"}
```

### Ejemplo de cadena de conexión con nombre secreto

La siguiente cadena tiene el nombre secreto `${Test/RDS/MySQL1}`.

```
mysql://jdbc:mysql://mysql1.host:3306/default?...&${Test/RDS/MySQL1}&...
```

El conector usa el nombre secreto para recuperar los secretos y proporcionar el nombre de usuario y la contraseña, como en el siguiente ejemplo.

```
mysql://jdbc:mysql://mysql1host:3306/default?...&user=sample2&password=sample2&...
```

Actualmente, el conector de MySQL reconoce las propiedades `user` y `password` de JDBC.

### Uso de un controlador de conexión única

Puede usar los siguientes controladores de registros y metadatos de conexión única para conectarse a una sola instancia de MySQL.

Tipo de controlador	Clase
Controlador compuesto	<code>MySQLCompositeHandler</code>
Controlador de metadatos	<code>MySQLMetadataHandler</code>
Controlador de registros	<code>MySQLRecordHandler</code>



## Parámetros de controlador de conexión única

Parámetro	Descripción
default	Obligatorio. La cadena de conexión predeterminada.

Los controladores de conexión única admiten una instancia de base de datos y deben proporcionar un parámetro de cadena de conexión default. Se omiten todas las demás cadenas de conexión.

La siguiente propiedad de ejemplo es para una sola instancia de MySQL compatible con una función de Lambda.

Propiedad	Valor
default	mysql://mysql1.host:3306/default?secret=Test/RDS/ MySQL1

## Parámetros de vertido

El SDK de Lambda puede verter datos en Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación.

Parámetro	Descripción
spill_bucket	Obligatorio. Nombre del bucket de vertido.
spill_prefix	Obligatorio. Prefijo de la clave del bucket de vertido.
spill_put_request_headers	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud putObject de Amazon S3 que se usa para el vertido (por ejemplo, {"x-amz-server-side-encryption" : "AES256"} ). Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.

## Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos correspondientes para JDBC y Arrow.

JDBC	Arrow
Booleano	Bit
Entero	Pequeño
Short	Smallint
Entero	Int
Largo	Bigint
float	Float4
Doble	Float8
Date	DateDay
Marca de tiempo	DateMilli
Cadena	Varchar
Bytes	Varbinary
BigDecimal	Decimal
ARRAY	Enumeración

## Particiones y divisiones

Las particiones se usan para determinar cómo generar divisiones para el conector. Athena crea una columna sintética de tipo `varchar` que representa el esquema de partición de la tabla para ayudar al conector a generar divisiones. El conector no modifica la definición de la tabla real.

## Rendimiento

MySQL admite particiones nativas. El conector MySQL de Athena puede recuperar datos de estas particiones en paralelo. Si quiere consultar conjuntos de datos muy grandes con una distribución uniforme de particiones, se recomienda encarecidamente la partición nativa.

El conector MySQL de Athena inserta predicados para reducir los datos analizados en la consulta. Los predicados simples, las expresiones complejas y las cláusulas LIMIT se insertan en el conector para reducir la cantidad de datos analizados y disminuir el tiempo de ejecución de las consultas.

## Cláusulas LIMIT

Una instrucción LIMIT N reduce los datos analizados en la consulta. Con la inserción LIMIT N, el conector devuelve solo las filas N a Athena.

## Predicados

Un predicado es una expresión de la cláusula WHERE de una consulta SQL que da como resultado un valor booleano y filtra las filas en función de varias condiciones. El conector MySQL de Athena puede combinar estas expresiones e insertarlas directamente en MySQL para mejorar la funcionalidad y reducir la cantidad de datos analizados.

Los siguientes operadores del conector MySQL de Athena admiten la inserción de predicados:

- Booleano: AND, OR, NOT.
- Igualdad: EQUAL, NOT\_EQUAL, LESS\_THAN, LESS\_THAN\_OR\_EQUAL, GREATER\_THAN, GREATER\_THAN\_OR\_EQUAL, IS\_DISTINCT\_FROM, NULL\_IF, IS\_NULL.
- Aritmética: ADD, SUBTRACT, MULTIPLY, DIVIDE, MODULUS, NEGATE.
- Otros: LIKE\_PATTERN, IN.

## Ejemplo de inserción combinada

Para mejorar las capacidades de consulta, combine los tipos de inserciones, como en el siguiente ejemplo:

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
LIMIT 10;
```

Para obtener información sobre cómo usar la inserción de predicados a fin de mejorar el rendimiento en las consultas federadas, incluida MySQL, consulte el artículo [Mejorar las consultas federadas con la inserción de predicados en Amazon Athena](#) en el Blog de macrodatos de AWS.

## Consultas de acceso directo

El conector MySQL admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.

Para usar consultas de acceso directo con MySQL, puede utilizar la siguiente sintaxis:

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'query string'  
    ))
```

El siguiente ejemplo de consulta envía una consulta a un origen de datos en MySQL. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## Información sobre licencias

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

Para obtener la información más reciente sobre la versión del controlador JDBC, consulte el archivo [pom.xml](#) para el conector de MySQL en GitHub.com.

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector para Neptune de Amazon Athena

Amazon Neptune es un servicio de base de datos de gráficos rápido, fiable y completamente administrado que le permite crear y ejecutar fácilmente aplicaciones que funcionen con conjuntos de datos altamente conectados. El motor de base de datos de gráficos de alto rendimiento diseñado específicamente de Neptune almacena de manera óptima miles de millones de relaciones y consulta gráficos con una latencia de milisegundos. Para obtener más información, consulte la [Guía del usuario de Neptune](#).

El conector Amazon Athena Neptune permite a Athena comunicarse con la instancia de base de datos de gráficos de Neptune, lo que hace que sus datos de gráficos de Neptune sean accesibles mediante consultas SQL.

Si Lake Formation está habilitado en la cuenta, el rol de IAM del conector de Lambda federado de Athena que haya implementado en AWS Serverless Application Repository debe tener acceso de lectura en Lake Formation para AWS Glue Data Catalog.

### Requisitos previos

El uso del conector de Neptune requiere los tres pasos siguientes.

- Configuración de un clúster de Neptune
- Configuración de un AWS Glue Data Catalog
- Implementación del conector en su Cuenta de AWS. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#). Para obtener más información específica sobre la implementación del conector de Neptune, consulte [Implementación del conector de Neptune de Amazon Athena](#) en GitHub.com.

### Limitaciones

Actualmente, el conector de Neptune tiene la siguiente limitación.

- No se admite la proyección de columnas, ni siquiera la clave principal (ID).

### Configuración de un clúster de Neptune

Si no tiene un clúster existente de Amazon Neptune y un conjunto de datos de gráficos de propiedades que le gustaría usar, debe configurar uno.

Asegúrese de tener una puerta de enlace de Internet y una puerta de enlace NAT en la VPC que aloja el clúster de Neptune. Las subredes privadas que utiliza la función de Lambda del conector de Neptune deben tener una ruta a Internet a través de esta puerta de enlace NAT. La función de Lambda del conector de Neptune utiliza la puerta de enlace NAT para comunicarse con AWS Glue.

Para obtener instrucciones sobre cómo configurar un nuevo clúster de Neptune y cargarlo con un conjunto de datos de ejemplo, consulte [Configuración del clúster de Neptune de ejemplo](#) en GitHub.com.

## Configuración de un AWS Glue Data Catalog

A diferencia de los almacenes de datos relacionales tradicionales, los nodos y periféricas de la base de datos de gráficos Neptune no utilizan ningún esquema establecido. Cada entrada puede tener diferentes campos y tipos de datos. Sin embargo, dado que el conector de Neptune recupera los metadatos de AWS Glue Data Catalog, debe crear una base de datos de AWS Glue que tenga tablas con el esquema requerido. Después de crear la base de datos y tablas de AWS Glue, el conector puede rellenar la lista de tablas disponibles para consultarla desde Athena.

### Habilitación de la coincidencia de columnas sin distinción entre mayúsculas y minúsculas

Si desea resolver los nombres de las columnas de la tabla de Neptune con las mayúsculas y minúsculas correctas, incluso cuando todos los nombres de columnas aparezcan en minúsculas en AWS Glue, puede configurar el conector Neptune para que no distinga entre mayúsculas y minúsculas.

Para habilitar esta característica, establezca la variable de entorno `enable_caseinsensitivematch` como `true` en la función de Lambda del conector Neptune.

### Especificación del parámetro de tabla `glabel` de AWS Glue para los nombres de las tablas en mayúsculas

Dado que AWS Glue solo admite nombres de tablas en minúsculas, es importante especificar el parámetro de tabla `glabel` de AWS Glue si se crea una tabla de AWS Glue para Neptune y el nombre de la tabla de Neptune incluye mayúsculas.

En la definición de la tabla de AWS Glue, incluya el parámetro `glabel` y establezca su valor en el nombre de la tabla con las mayúsculas originales. Esto garantiza que se conserven las mayúsculas correctas cuando AWS Glue interactúe con la tabla de Neptune. En el siguiente ejemplo, se establece el valor de `glabel` como el nombre de la tabla `Airport`.

```
glabel = Airport
```

Table properties (3)	
Key	Value
<code>separatorChar</code>	<code>,</code>
<code>componenttype</code>	<code>vertex</code>
<code>glabel</code>	<code>Airport</code>

Para obtener más información sobre cómo configurar AWS Glue Data Catalog para utilizarlo con Neptune, consulte [Configurar un catálogo de AWS Glue](#) en GitHub.com.

## Rendimiento

El conector Neptune de Athena inserta predicados para reducir los datos analizados en la consulta. Sin embargo, los predicados que usan la clave principal provocan un error en la consulta. Las cláusulas LIMIT reducen la cantidad de datos analizados; sin embargo, si no proporciona un predicado, es probable que las consultas SELECT con una cláusula LIMIT analicen al menos 16 MB de datos. El conector para Neptune resiste las limitaciones debidas a la simultaneidad.

## Consultas de acceso directo

El conector de Neptune admite [consultas de acceso directo](#). Puede utilizar esta característica para ejecutar consultas Gremlin en gráficos de propiedades y para ejecutar consultas SPARQL en datos RDF.

Para crear consultas de acceso directo con Neptune, puede utilizar la siguiente sintaxis:

```
SELECT * FROM TABLE(  
  system.query(  
    DATABASE => 'database_name',  
    COLLECTION => 'collection_name',  
    QUERY => 'query_string'  
  ))
```

En el siguiente ejemplo, se filtra la consulta de acceso directo de Neptune para los aeropuertos con el código ATL.

```
SELECT * FROM TABLE(  
  system.query(  
    DATABASE => 'graph-database',  
    COLLECTION => 'airport',  
    QUERY => 'g.V().has(\"airport\", \"code\", \"ATL\").valueMap()',  
  ))
```

## Recursos adicionales de

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector para OpenSearch de Amazon Athena

### OpenSearch Service

El conector Amazon Athena OpenSearch permite a Amazon Athena comunicarse con sus instancias de OpenSearch para que pueda utilizar SQL para consultar los datos de OpenSearch.

#### Note

Debido a un problema conocido, el conector OpenSearch no se puede utilizar con una VPC.

Si Lake Formation está habilitado en la cuenta, el rol de IAM del conector de Lambda federado de Athena que haya implementado en AWS Serverless Application Repository debe tener acceso de lectura en Lake Formation para AWS Glue Data Catalog.

### Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

### Términos

Los siguientes términos hacen referencia al conector de OpenSearch.

- **Dominio:** un nombre que este conector asocia con el punto de conexión de la instancia de OpenSearch. El dominio también se usa como nombre de base de datos. Para las instancias de OpenSearch definidas en Amazon OpenSearch Service, el dominio se puede detectar automáticamente. Para otras instancias, debe proporcionar una asignación entre el nombre de dominio y el punto de conexión.
- **Índice:** una tabla de base de datos definida en la instancia de OpenSearch.
- **Asignación:** si un índice es una tabla de base de datos, la asignación es su esquema (es decir, la definición de sus campos y atributos).

Este conector admite tanto la recuperación de metadatos de la instancia de OpenSearch como de AWS Glue Data Catalog. Si el conector encuentra una base de datos y una tabla de AWS Glue que coincidan con los nombres de dominio e índice de OpenSearch, el conector intenta utilizarlos



para la definición del esquema. Le recomendamos que cree su tabla de AWS Glue para que sea un superconjunto de todos los campos del índice de OpenSearch.

- Documento: un registro dentro de una tabla de base de datos.
- Flujo de datos: datos basados en el tiempo que se componen de varios índices de respaldo. Para obtener más información, consulte [Flujos de datos](#) en la documentación de OpenSearch y [Introducción a los flujos de datos](#) en la Guía para desarrolladores de Amazon OpenSearch Service.

#### Note

Como los índices de flujos de datos se crean y administran internamente mediante una búsqueda abierta, el conector elige la asignación de esquemas del primer índice disponible. Por este motivo, recomendamos encarecidamente configurar una tabla de AWS Glue como origen de metadatos complementario. Para obtener más información, consulte [Configuración de bases de datos y tablas en AWS Glue](#).

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de OpenSearch.

- `spill_bucket`: especifica el bucket de Amazon S3 para los datos que superen los límites de la función de Lambda.
- `spill_prefix`: (opcional) de forma predeterminada, se establece una subcarpeta en la carpeta especificada `spill_bucket` llamada `athena-federation-spill`. Le recomendamos configurar un [ciclo de vida de almacenamiento](#) de Amazon S3 en esta ubicación para eliminar vertidos de más de un número predeterminado de días u horas.
- `spill_put_request_headers`: (opcional) un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud `putObject` de Amazon S3 que se usa para el vertidos (por ejemplo, `{"x-amz-server-side-encryption" : "AES256"}`). Para ver otros encabezados posibles, consulte [PutObject](#) en la referencia de la API de Amazon Simple Storage Service.
- `kms_key_id`: (opcional) de forma predeterminada, los datos que se vierten a Amazon S3 se cifran mediante el modo de cifrado autenticado AES-GCM y una clave generada aleatoriamente. Para que la función de Lambda use claves de cifrado más seguras generadas por KMS, como `a7e63k4b-81oc-40db-a2a1-4d0en2cd8331`, puede especificar un ID de clave de KMS.
- `disable_spill_encryption`: (opcional) cuando se establece en `True`, desactiva el cifrado del vertido. El valor predeterminado es `False`, de modo que los datos que se vierten a S3 se cifran mediante

AES-GCM, ya sea mediante una clave generada aleatoriamente o KMS para generar claves. La desactivación del cifrado de vertido puede mejorar el rendimiento, especialmente si su ubicación de vertido usa [cifrado del servidor](#).

- `disable_glue`: (opcional) si está presente y se establece en `true` (verdadero), el conector no intentará recuperar metadatos complementarios de AWS Glue.
- `query_timeout_cluster`: el tiempo de espera, en segundos, para las consultas de estado del clúster utilizadas en la generación de análisis paralelos.
- `query_timeout_search`: el tiempo de espera, en segundos, de las consultas de búsqueda utilizadas en la recuperación de documentos de un índice.
- `auto_discover_endpoint`: booleano. El valor predeterminado es `true`. Cuando utiliza Amazon OpenSearch Service y establece este parámetro en `true` (verdadero), el conector puede detectar automáticamente los dominios y puntos de conexión al llamar a las operaciones de API de descripción o lista correspondientes en el servicio OpenSearch Service. Para cualquier otro tipo de instancia de OpenSearch (por ejemplo, autoalojada), debe especificar los puntos de conexión del dominio asociados en la variable `domain_mapping`. Si `auto_discover_endpoint=true`, el conector usa las credenciales de AWS para autenticarse en OpenSearch Service. De lo contrario, el conector recupera las credenciales del nombre de usuario y contraseña de AWS Secrets Manager a través de la variable `domain_mapping`.
- `domain_mapping`: se usa solo cuando `auto_discover_endpoint` se establece en `false` (falso) y define la asignación entre los nombres de dominio y sus puntos de conexión asociados. La variable `domain_mapping` puede admitir varios puntos de conexión de OpenSearch con el siguiente formato:

```
domain1=endpoint1,domain2=endpoint2,domain3=endpoint3,...
```

Con el fin de autenticarse en un punto de conexión de OpenSearch, el conector admite cadenas de sustitución inyectadas con el formato `${SecretName}`: y con nombre de usuario y contraseña obtenidos de AWS Secrets Manager. Los dos puntos (:) al final de la expresión sirven como separador del resto del punto de conexión.

#### Important

Como práctica recomendada en materia de seguridad, no utilice credenciales codificadas en las variables de entorno ni en las cadenas de conexión. Para obtener información sobre

cómo transferir los secretos codificados a AWS Secrets Manager, consulte [Mover secretos codificados a AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

En el siguiente ejemplo se usa el secreto `opensearch-creds`.

```
movies=https://{opensearch-creds}:search-movies-ne...qu.us-east-1.es.amazonaws.com
```

En tiempo de ejecución, `{opensearch-creds}` se representa como nombre de usuario y contraseña, como en el siguiente ejemplo.

```
movies=https://myusername@mypassword:search-movies-ne...qu.us-east-1.es.amazonaws.com
```

En el parámetro `domain_mapping`, cada par dominio-punto de conexión puede usar un secreto diferente. El secreto en sí se debe especificar en el formato `user_name@password`. Aunque la contraseña puede contener signos `@` incrustados, la primera `@` sirve como separador de `user_name`.

También es importante tener en cuenta que este conector utiliza la coma (,) y el signo igual (=) como separadores para los pares dominio-punto de conexión. Por este motivo, no debe utilizarlos en ningún lugar dentro del secreto almacenado.

## Configuración de bases de datos y tablas en AWS Glue

El conector obtiene información de metadatos mediante AWS Glue u OpenSearch. Puede configurar una tabla de AWS Glue como origen de la definición de metadatos suplementaria. Para habilitar esta característica, defina una base de datos y una tabla de AWS Glue que coincidan con el dominio y el índice del origen que va a complementar. El conector también puede aprovechar las definiciones de metadatos almacenadas en la instancia de OpenSearch al recuperar la asignación para el índice especificado.

## Definición de metadatos para matrices en OpenSearch

OpenSearch no tiene un tipo de datos de matrices dedicado. Cualquier campo puede contener cero o más valores siempre que sean del mismo tipo de datos. Si desea utilizar OpenSearch como origen de definición de metadatos, debe definir una propiedad `_meta` para todos los índices utilizados con Athena en los campos que se van a considerar una lista o matriz. Si no completa este paso, las

consultas devuelven solo el primer elemento del campo de la lista. Al especificar la propiedad `_meta`, los nombres del campo deben estar totalmente cualificados para las estructuras JSON anidadas (por ejemplo, `address.street`, en la que `street` es un campo anidado dentro de una estructura de `address`).

En el siguiente ejemplo, se definen las listas `actor` y `genre` en la tabla `movies`.

```
PUT movies/_mapping
{
  "_meta": {
    "actor": "list",
    "genre": "list"
  }
}
```

## Tipos de datos

El conector de OpenSearch puede extraer definiciones de metadatos de AWS Glue o de la instancia de OpenSearch. El conector utiliza la asignación de la siguiente tabla para convertir las definiciones en tipos de datos de Apache Arrow, incluidos los puntos que se indican en la sección siguiente.

OpenSearch	Apache Arrow	AWS Glue
text, keyword, binary	VARCHAR	cadena
long	BIGINT	bigint
scaled_float	BIGINT	SCALED_FLOAT(...)
integer	INT	int
short	SMALLINT	smallint
byte	TINYINT	tinyint
double	FLOAT8	double
float, half_float	FLOAT4	float
boolean	BIT	boolean

OpenSearch	Apache Arrow	AWS Glue
date, date_nanos	DATEMILLI	Marca de tiempo
Estructura JSON	STRUCT	STRUCT
_meta (para obtener información, consulte la sección <a href="#">Definición de metadatos para matrices en OpenSearch</a> ).	LIST	ARRAY

### Notas sobre los tipos de datos

- Actualmente, el conector solo admite OpenSearch y los tipos de datos de AWS Glue incluidos en la tabla anterior.
- `scaled_float` es un número de coma flotante escalado por un factor de escala doble fijo y representado como `BIGINT` en Apache Arrow. Por ejemplo, 0,756 con un factor de escala de 100 se redondea a 76.
- Para definir un `scaled_float` en AWS Glue, debe seleccionar la columna de `array` y declarar el campo con el formato `SCALED_FLOAT` (*scaling\_factor*).

Los ejemplos siguientes son válidos:

```
SCALED_FLOAT(10.51)
SCALED_FLOAT(100)
SCALED_FLOAT(100.0)
```

Los ejemplos siguientes no son válidos:

```
SCALED_FLOAT(10.)
SCALED_FLOAT(.5)
```

- Al hacer la conversión de `date_nanos` a `DATEMILLI`, los nanosegundos se redondean al milisegundo más próximo. Los valores válidos de `date` y `date_nanos` incluyen los siguientes formatos, entre otros:

```
"2020-05-18T10:15:30.123456789"
"2020-05-15T06:50:01.123Z"
```

```
"2020-05-15T06:49:30.123-05:00"  
1589525370001 (epoch milliseconds)
```

- Un elemento binary de OpenSearch es una representación de cadena de un valor binario codificado mediante Base64 y se convierte a VARCHAR.

## Ejecución de consultas SQL

Los siguientes son ejemplos de consultas DDL que puede usar con este conector. En los ejemplos, *function\_name* corresponde al nombre de la función de Lambda, *domain* es el nombre del dominio que se desea consultar e *index* es el nombre del índice.

```
SHOW DATABASES in `lambda:function_name`
```

```
SHOW TABLES in `lambda:function_name`.`domain`
```

```
DESCRIBE `lambda:function_name`.`domain`.`index`
```

## Rendimiento

El conector de OpenSearch de Athena admite análisis paralelos basados en particiones. El conector utiliza la información de estado del clúster recuperada de la instancia de OpenSearch para generar varias solicitudes de consulta de búsqueda de documentos. Las solicitudes se dividen para cada partición y se ejecutan de forma simultánea.

El conector también inserta los predicados como parte de sus consultas de búsqueda de documentos. El siguiente ejemplo de consulta y predicado muestra cómo utiliza el conector la inserción de predicados.

## Query

```
SELECT * FROM "lambda:elasticsearch".movies.movies  
WHERE year >= 1955 AND year <= 1962 OR year = 1996
```

## Predicado

```
(_exists_:year) AND year:([1955 TO 1962] OR 1996)
```

## Consultas de acceso directo

El conector OpenSearch admite [consultas de acceso directo](#) y utiliza el lenguaje Query DSL. Para obtener más información sobre las consultas con Query DSL, consulte [Query DSL](#) en la documentación de Elasticsearch o [Query DSL](#) en la documentación de OpenSearch.

Para utilizar consultas de acceso directo con el conector OpenSearch, utilice la siguiente sintaxis:

```
SELECT * FROM TABLE(  
  system.query(  
    schema => 'schema_name',  
    index => 'index_name',  
    query => "{query_string}"  
  ))
```

En el siguiente ejemplo de OpenSearch, se filtran las consultas de acceso directo para los empleados con un estado laboral activo en el índice `employee` del esquema `default`.

```
SELECT * FROM TABLE(  
  system.query(  
    schema => 'default',  
    index => 'employee',  
    query => "{ 'bool':{ 'filter':{ 'term':{ 'status': 'active' } } } }"  
  ))
```

## Recursos adicionales de

- Para obtener información sobre cómo usar el conector OpenSearch de Amazon Athena para consultar datos en Amazon OpenSearch Service y Amazon S3 en una sola consulta, consulte el artículo [Consulta de datos en Amazon OpenSearch Service mediante SQL desde Amazon Athena](#) en el Blog de macrodatos de AWS.
- Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector Oracle de Amazon Athena

El conector de Oracle de Amazon Athena permite que Amazon Athena ejecute consultas de SQL en los datos almacenados en Oracle que se ejecutan en las instalaciones, en Amazon EC2 o en Amazon RDS. También puede utilizar el conector para consultar datos en [Oracle Exadata](#).

## Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

## Limitaciones

- Las operaciones de escritura de DDL no son compatibles.
- En una configuración de multiplexor, el bucket de vertido y el prefijo se comparten en todas las instancias de base de datos.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.

## Términos

Los siguientes términos hacen referencia al conector de Oracle.

- Instancia de base de datos: cualquier instancia de base de datos implementada en las instalaciones, en Amazon EC2 o en Amazon RDS.
- Controlador: un controlador de Lambda que accede a la instancia de base de datos. Un controlador puede ser para los metadatos o para los registros de datos.
- Controlador de metadatos: un controlador de Lambda que recupera los metadatos de la instancia de base de datos.
- Controlador de registros: un controlador de Lambda que recupera registros de datos de la instancia de base de datos.
- Controlador compuesto: un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.
- Propiedad o parámetro: una propiedad de base de datos que usan los controladores para extraer información de la base de datos. Estas propiedades se configuran como variables de entorno de Lambda.
- Cadena de conexión: una cadena de texto que se usa para establecer una conexión con una instancia de base de datos.



- Catálogo: un catálogo ajeno a AWS Glue registrado en Athena que es un prefijo obligatorio para la propiedad `connection_string`.
- Controlador de multiplexación: un controlador de Lambda que puede aceptar y usar varias conexiones a bases de datos.

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de Oracle.

## Cadena de conexión

Use una cadena de conexión JDBC con el siguiente formato para conectarse a una instancia de base de datos.

```
oracle://${jdbc_connection_string}
```

### Note

Si la contraseña contiene caracteres especiales (por ejemplo, `some.password`), escríbala entre comillas dobles cuando la pase a la cadena de conexión (por ejemplo, `"some.password"`). Si no lo hace, puede producirse un error de URL de Oracle especificada no válida.

## Uso de un controlador de multiplexación

Puede usar un multiplexor para conectarse a varias instancias de base de datos con una sola función de Lambda. Las solicitudes se enrutan por nombre de catálogo. Use las siguientes clases en Lambda.

Controlador	Clase
Controlador compuesto	<code>OracleMuxCompositeHandler</code>
Controlador de metadatos	<code>OracleMuxMetadataHandler</code>
Controlador de registros	<code>OracleMuxRecordHandler</code>

## Parámetros del controlador de multiplexación

Parámetro	Descripción
<code>\$<i>catalog</i>_connection_string</code>	Obligatorio. Una cadena de conexión de instancia de base de datos. Agregue el prefijo a la variable de entorno con el nombre del catálogo usado en Athena. Por ejemplo, si el catálogo registrado en Athena es <code>myoraclecatalog</code> , el nombre de la variable de entorno será <code>myoraclecatalog_connection_string</code> .
<code>default</code>	Obligatorio. La cadena de conexión predeterminada. Esta cadena se usa cuando el catálogo es <code>lambda:\${AWS_LAMBDA_FUNCTION_NAME}</code> .

Las siguientes propiedades de ejemplo son para una función de Lambda de Oracle MUX que admite dos instancias de base de datos: `oracle1` (la predeterminada) y `oracle2`.

Propiedad	Valor
<code>default</code>	<code>oracle://jdbc:oracle:thin:\${Test/RDS/Oracle1}@//oracle1.hostname:port/serviceName</code>
<code>oracle_catalog1_connection_string</code>	<code>oracle://jdbc:oracle:thin:\${Test/RDS/Oracle1}@//oracle1.hostname:port/serviceName</code>
<code>oracle_catalog2_connection_string</code>	<code>oracle://jdbc:oracle:thin:\${Test/RDS/Oracle2}@//oracle2.hostname:port/serviceName</code>

### Proporcionar credenciales

Para proporcionar un nombre de usuario y una contraseña para la base de datos en la cadena de conexión JDBC, puede usar las propiedades de la cadena de conexión o AWS Secrets Manager.

- Cadena de conexión: se pueden especificar un nombre de usuario y una contraseña como propiedades en la cadena de conexión JDBC.

#### Important

Como práctica recomendada en materia de seguridad, no utilice credenciales codificadas en las variables de entorno ni en las cadenas de conexión. Para obtener información sobre cómo transferir los secretos codificados a AWS Secrets Manager, consulte [Mover secretos codificados a AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

- AWS Secrets Manager: para utilizar la característica Consulta federada de Athena con AWS Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Puede poner el nombre de un secreto en AWS Secrets Manager, en la cadena de conexión JDBC. El conector reemplaza el nombre del secreto por los valores `username` y `password` de Secrets Manager.

Para las instancias de bases de datos de Amazon RDS, esta compatibilidad está estrechamente integrada. Si usa Amazon RDS, le recomendamos encarecidamente que use AWS Secrets Manager y la rotación de credenciales. Si la base de datos no usa Amazon RDS, guarde las credenciales como archivos JSON con el siguiente formato:

```
{"username": "${username}", "password": "${password}"}
```

#### Note

Si la contraseña contiene caracteres especiales (por ejemplo, `some .password`), escríbala entre comillas dobles cuando la guarde en Secrets Manager (por ejemplo, `"some .password"`). Si no lo hace, puede producirse un error de URL de Oracle especificada no válida.

Ejemplo de cadena de conexión con nombre secreto

La siguiente cadena tiene el nombre secreto `${Test/RDS/Oracle}`.

```
oracle://jdbc:oracle:thin:${Test/RDS/Oracle}@//hostname:port/servicename
```

El conector usa el nombre secreto para recuperar los secretos y proporcionar el nombre de usuario y la contraseña, como en el siguiente ejemplo.

```
oracle://jdbc:oracle:thin:username/password@//hostname:port/servicename
```

Actualmente, el conector de Oracle reconoce las propiedades UID y PWD de JDBC.

### Uso de un controlador de conexión única

Puede usar los siguientes controladores de registros y metadatos de conexión única para conectarse a una sola instancia de Oracle.

Tipo de controlador	Clase
Controlador compuesto	<code>OracleCompositeHandler</code>
Controlador de metadatos	<code>OracleMetadataHandler</code>
Controlador de registros	<code>OracleRecordHandler</code>

### Parámetros de controlador de conexión única

Parámetro	Descripción
<code>default</code>	Obligatorio. La cadena de conexión predeterminada.
<code>IsFIPSEnabled</code>	Opcional. Se establece en <code>true</code> cuando el modo FIPS está habilitado. El valor predeterminado es <code>false</code> .

Los controladores de conexión única admiten una instancia de base de datos y deben proporcionar un parámetro de cadena de conexión `default`. Se omiten todas las demás cadenas de conexión.

El conector admite conexiones basadas en SSL con instancias de Amazon RDS. La compatibilidad se limita al protocolo de seguridad de la capa de transporte (TLS) y a la autenticación del servidor por

parte del cliente. La autenticación mutua no se admite en Amazon RDS. La segunda fila de la tabla siguiente muestra la sintaxis para usar SSL.

La siguiente propiedad de ejemplo es para una sola instancia de Oracle compatible con una función de Lambda.

Propiedad	Valor
default	oracle://jdbc:oracle:thin:\${Test/RDS/Oracle}@//hostname:port/servicename
	oracle://jdbc:oracle:thin:\${Test/RDS/Oracle}@((DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS) (HOST=<HOST_NAME>)(PORT=)))(CONNECT_DATA=(SID=))(SECURITY=(SSL_SERVER_CERT_DN=)))

### Parámetros de vertido

El SDK de Lambda puede verter datos en Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación.

Parámetro	Descripción
spill_bucket	Obligatorio. Nombre del bucket de vertido.
spill_prefix	Obligatorio. Prefijo de la clave del bucket de vertido.
spill_put_request_headers	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud <code>putObject</code> de Amazon S3 que se usa para el vertido (por ejemplo, <code>{"x-amz-server-side-encryption" : "AES256"}</code> ). Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.

### Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos correspondientes para JDBC, Oracle y Arrow.

JDBC	Oracle	Arrow
Booleano	boolean	Bit
Entero	N/A	Pequeño
Short	smallint	Smallint
Entero	integer	Int
Largo	bigint	Bigint
float	float4	Float4
Doble	float8	Float8
Date	date	DateDay
Timestamp	Marca de tiempo	DateMilli
Cadena	texto	Varchar
Bytes	bytes	Varbinary
BigDecimal	numeric(p,s)	Decimal
ARRAY	N/D (ver nota)	Enumeración

## Particiones y divisiones

Las particiones se usan para determinar cómo generar divisiones para el conector. Athena crea una columna sintética de tipo `varchar` que representa el esquema de partición de la tabla para ayudar al conector a generar divisiones. El conector no modifica la definición de la tabla real.

## Rendimiento

Oracle admite particiones nativas. El conector Oracle de Athena puede recuperar datos de estas particiones en paralelo. Si quiere consultar conjuntos de datos muy grandes con una distribución uniforme de particiones, se recomienda encarecidamente la partición nativa. La selección de un subconjunto de columnas acelera significativamente el tiempo de ejecución de las consultas y

reduce la cantidad de datos analizados. El conector Oracle resiste las limitaciones debidas a la simultaneidad. Sin embargo, los tiempos de ejecución de las consultas suelen ser prolongados.

El conector Oracle de Athena inserta predicados para reducir los datos analizados en la consulta. Los predicados simples y las expresiones complejas se insertan en el conector para reducir la cantidad de datos analizados y disminuir el tiempo de ejecución de las consultas.

## Predicados

Un predicado es una expresión de la cláusula WHERE de una consulta SQL que da como resultado un valor booleano y filtra las filas en función de varias condiciones. El conector Oracle de Athena puede combinar estas expresiones e insertarlas directamente en Oracle para mejorar la funcionalidad y reducir la cantidad de datos analizados.

Los siguientes operadores del conector Oracle de Athena admiten la inserción de predicados:

- Booleano: AND, OR, NOT.
- Igualdad: EQUAL, NOT\_EQUAL, LESS\_THAN, LESS\_THAN\_OR\_EQUAL, GREATER\_THAN, GREATER\_THAN\_OR\_EQUAL, IS\_NULL.
- Aritmética: ADD, SUBTRACT, MULTIPLY, DIVIDE, NEGATE.
- Otros: LIKE\_PATTERN, IN.

## Ejemplo de inserción combinada

Para mejorar las capacidades de consulta, combine los tipos de inserciones, como en el siguiente ejemplo:

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%');
```

## Consultas de acceso directo

El conector de Oracle admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.

Para usar consultas de transferencia de datos con Oracle, puede utilizar la siguiente sintaxis:

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'query string'  
    ))
```

El siguiente ejemplo de consulta envía una consulta a un origen de datos de Oracle. La consulta selecciona todas las columnas de la tabla `customer`.

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer'  
    ))
```

## Información sobre licencias

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

Para obtener la información más reciente sobre la versión del controlador JDBC, consulte el archivo [pom.xml](#) para el conector de Oracle en GitHub.com.

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector para PostgreSQL de Amazon Athena

El conector de PostgreSQL de Amazon Athena permite a Athena acceder a las bases de datos de PostgreSQL.

## Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).



## Limitaciones

- Las operaciones de escritura de DDL no son compatibles.
- En una configuración de multiplexor, el bucket de vertido y el prefijo se comparten en todas las instancias de base de datos.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.

## Términos

Los siguientes términos hacen referencia al conector de PostgreSQL.

- Instancia de base de datos: cualquier instancia de base de datos implementada en las instalaciones, en Amazon EC2 o en Amazon RDS.
- Controlador: un controlador de Lambda que accede a la instancia de base de datos. Un controlador puede ser para los metadatos o para los registros de datos.
- Controlador de metadatos: un controlador de Lambda que recupera los metadatos de la instancia de base de datos.
- Controlador de registros: un controlador de Lambda que recupera registros de datos de la instancia de base de datos.
- Controlador compuesto: un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.
- Propiedad o parámetro: una propiedad de base de datos que usan los controladores para extraer información de la base de datos. Estas propiedades se configuran como variables de entorno de Lambda.
- Cadena de conexión: una cadena de texto que se usa para establecer una conexión con una instancia de base de datos.
- Catálogo: un catálogo ajeno a AWS Glue registrado en Athena que es un prefijo obligatorio para la propiedad `connection_string`.
- Controlador de multiplexación: un controlador de Lambda que puede aceptar y usar varias conexiones a bases de datos.

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de PostgreSQL.

## Cadena de conexión

Use una cadena de conexión JDBC con el siguiente formato para conectarse a una instancia de base de datos.

```
postgres://${jdbc_connection_string}
```

## Uso de un controlador de multiplexación

Puede usar un multiplexor para conectarse a varias instancias de base de datos con una sola función de Lambda. Las solicitudes se enrutan por nombre de catálogo. Use las siguientes clases en Lambda.

Controlador	Clase
Controlador compuesto	PostGreSqlMuxCompositeHandler
Controlador de metadatos	PostGreSqlMuxMetadataHandler
Controlador de registros	PostGreSqlMuxRecordHandler

## Parámetros del controlador de multiplexación

Parámetro	Descripción
<code>catalog_connection_string</code>	Obligatorio. Una cadena de conexión de instancia de base de datos. Agregue el prefijo a la variable de entorno con el nombre del catálogo usado en Athena. Por ejemplo, si el catálogo registrado en Athena es <code>mypostgrescatalog</code> , el nombre de la variable de entorno será <code>mypostgrescatalog_connection_string</code> .
<code>default</code>	Obligatorio. La cadena de conexión predeterminada. Esta cadena se usa cuando el catálogo es <code>lambda:\${AWS_LAMBDA_FUNCTION_NAME}</code> .

Las siguientes propiedades de ejemplo son para una función de Lambda de PostgreSQL MUX que admite dos instancias de base de datos: `postgres1` (la predeterminada) y `postgres2`.

Propiedad	Valor
<code>default</code>	<code>postgres://jdbc:postgresql://postgres1.host:5432/default?\${Test/RDS/PostGres1}</code>
<code>postgres_catalog1_connection_string</code>	<code>postgres://jdbc:postgresql://postgres1.host:5432/default?\${Test/RDS/PostGres1}</code>
<code>postgres_catalog2_connection_string</code>	<code>postgres://jdbc:postgresql://postgres2.host:5432/default?user=sample&amp;password=sample</code>

### Proporcionar credenciales

Para proporcionar un nombre de usuario y una contraseña para la base de datos en la cadena de conexión JDBC, puede usar las propiedades de la cadena de conexión o AWS Secrets Manager.

- Cadena de conexión: se pueden especificar un nombre de usuario y una contraseña como propiedades en la cadena de conexión JDBC.

#### Important

Como práctica recomendada en materia de seguridad, no utilice credenciales codificadas en las variables de entorno ni en las cadenas de conexión. Para obtener información sobre cómo transferir los secretos codificados a AWS Secrets Manager, consulte [Mover secretos codificados a AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

- AWS Secrets Manager: para utilizar la característica Consulta federada de Athena con AWS Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Puede poner el nombre de un secreto en AWS Secrets Manager, en la cadena de conexión JDBC. El conector reemplaza el nombre del secreto por los valores `username` y `password` de Secrets Manager.

Para las instancias de bases de datos de Amazon RDS, esta compatibilidad está estrechamente integrada. Si usa Amazon RDS, le recomendamos encarecidamente que use AWS Secrets Manager y la rotación de credenciales. Si la base de datos no usa Amazon RDS, guarde las credenciales como archivos JSON con el siguiente formato:

```
{"username": "${username}", "password": "${password}"}
```

### Ejemplo de cadena de conexión con nombre secreto

La siguiente cadena tiene el nombre secreto `${Test/RDS/PostGres1}`.

```
postgres://jdbc:postgresql://postgres1.host:5432/default?...&${Test/RDS/PostGres1}&...
```

El conector usa el nombre secreto para recuperar los secretos y proporcionar el nombre de usuario y la contraseña, como en el siguiente ejemplo.

```
postgres://jdbc:postgresql://postgres1.host:5432/  
default?...&user=sample2&password=sample2&...
```

Actualmente, el conector de PostgreSQL reconoce las propiedades `user` y `password` de JDBC.

### Habilitar SSL

Para admitir SSL en su conexión PostgreSQL, agregue lo siguiente a la cadena de conexión:

```
&sslmode=verify-ca&sslfactory=org.postgresql.ssl.DefaultJavaSSLFactory
```

### Ejemplo

En el siguiente ejemplo de cadena de conexión no se utiliza SSL.

```
postgres://jdbc:postgresql://example-asdf-aurora-postgres-endpoint:5432/asdf?  
user=someuser&password=somepassword
```

Para habilitar SSL, modifique la cadena de la siguiente manera.

```
postgres://jdbc:postgresql://example-asdf-aurora-postgres-
endpoint:5432/asdf?user=someuser&password=somepassword&sslmode=verify-
ca&sslfactory=org.postgresql.ssl.DefaultJavaSSLFactory
```

## Uso de un controlador de conexión única

Puede usar los siguientes controladores de registros y metadatos de conexión única para conectarse a una sola instancia de PostgreSQL.

Tipo de controlador	Clase
Controlador compuesto	<code>PostGreSqlCompositeHandler</code>
Controlador de metadatos	<code>PostGreSqlMetadataHandler</code>
Controlador de registros	<code>PostGreSqlRecordHandler</code>

## Parámetros de controlador de conexión única

Parámetro	Descripción
<code>default</code>	Obligatorio. La cadena de conexión predeterminada.

Los controladores de conexión única admiten una instancia de base de datos y deben proporcionar un parámetro de cadena de conexión `default`. Se omiten todas las demás cadenas de conexión.

La siguiente propiedad de ejemplo es para una sola instancia de PostgreSQL compatible con una función de Lambda.

Propiedad	Valor
<code>default</code>	<code>postgres://jdbc:postgresql://postgres1.host:5432/default?secret=\${Test/RDS/PostgreSQL1}</code>

## Parámetros de vertido

El SDK de Lambda puede verter datos en Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación.

Parámetro	Descripción
<code>spill_bucket</code>	Obligatorio. Nombre del bucket de vertido.
<code>spill_prefix</code>	Obligatorio. Prefijo de la clave del bucket de vertido.
<code>spill_put_request_headers</code>	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud <code>putObject</code> de Amazon S3 que se usa para el vertido (por ejemplo, <code>{"x-amz-server-side-encryption" : "AES256"}</code> ). Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.

## Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos correspondientes para JDBC, PostgreSQL y Arrow.

JDBC	PostgreSQL	Arrow
Booleano	Booleano	Bit
Entero	N/A	Pequeño
Short	<code>smallint</code>	<code>Smallint</code>
Entero	<code>integer</code>	<code>Int</code>
Largo	<code>bigint</code>	<code>Bigint</code>
float	<code>float4</code>	<code>Float4</code>
Doble	<code>float8</code>	<code>Float8</code>

JDBC	PostgreSQL	Arrow
Date	date	DateDay
Timestamp	Marca de tiempo	DateMilli
Cadena	texto	Varchar
Bytes	bytes	Varbinary
BigDecimal	numeric(p,s)	Decimal
ARRAY	N/D (ver nota)	Enumeración

### Note

El tipo ARRAY se admite en el conector de PostgreSQL con las siguientes restricciones: no se admiten las matrices multidimensionales (`<data_type>[][]` o matrices anidadas). Las columnas con los tipos de datos ARRAY no admitidos se convierten a una matriz de elementos de cadena (`array<varchar>`).

## Particiones y divisiones

Las particiones se usan para determinar cómo generar divisiones para el conector. Athena crea una columna sintética de tipo `varchar` que representa el esquema de partición de la tabla para ayudar al conector a generar divisiones. El conector no modifica la definición de la tabla real.

## Rendimiento

PostgreSQL admite particiones nativas. El conector PostgreSQL de Athena puede recuperar datos de estas particiones en paralelo. Si quiere consultar conjuntos de datos muy grandes con una distribución uniforme de particiones, se recomienda encarecidamente la partición nativa.

El conector PostgreSQL de Athena inserta predicados para reducir los datos analizados en la consulta. Los predicados simples, las expresiones complejas y las cláusulas `LIMIT` se insertan en el conector para reducir la cantidad de datos analizados y disminuir el tiempo de ejecución de las consultas. Sin embargo, seleccionar un subconjunto de columnas da como resultado que en ocasiones se prolongue el tiempo de ejecución de las consultas.

## Cláusulas LIMIT

Una instrucción `LIMIT N` reduce los datos analizados en la consulta. Con la inserción `LIMIT N`, el conector devuelve solo las filas `N` a Athena.

## Predicados

Un predicado es una expresión de la cláusula `WHERE` de una consulta SQL que da como resultado un valor booleano y filtra las filas en función de varias condiciones. El conector PostgreSQL de Athena puede combinar estas expresiones e insertarlas directamente en PostgreSQL para mejorar la funcionalidad y reducir la cantidad de datos analizados.

Los siguientes operadores del conector PostgreSQL de Athena admiten la inserción de predicados:

- Booleano: `AND`, `OR`, `NOT`.
- Igualdad: `EQUAL`, `NOT_EQUAL`, `LESS_THAN`, `LESS_THAN_OR_EQUAL`, `GREATER_THAN`, `GREATER_THAN_OR_EQUAL`, `IS_DISTINCT_FROM`, `NULL_IF`, `IS_NULL`.
- Aritmética: `ADD`, `SUBTRACT`, `MULTIPLY`, `DIVIDE`, `MODULUS`, `NEGATE`.
- Otros: `LIKE_PATTERN`, `IN`.

## Ejemplo de inserción combinada

Para mejorar las capacidades de consulta, combine los tipos de inserciones, como en el siguiente ejemplo:

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
LIMIT 10;
```

## Consultas de acceso directo

El conector PostgreSQL admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.

Para usar consultas de acceso directo con PostgreSQL, puede utilizar la siguiente sintaxis:

```
SELECT * FROM TABLE(
```



```
system.query(  
    query => 'query string'  
))
```

El siguiente ejemplo de consulta envía una consulta a un origen de datos en PostgreSQL. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## Recursos adicionales de

Para obtener la información más reciente sobre la versión del controlador JDBC, consulte el archivo [pom.xml](#) para el conector de PostgreSQL en GitHub.com.

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector para Redis de Amazon Athena

El conector Amazon Athena Redis permite a Amazon Athena comunicarse con las instancias de Redis para que pueda consultar los datos de Redis con SQL. Puede usar AWS Glue Data Catalog para asignar sus pares clave-valor de Redis en tablas virtuales.

A diferencia de los conjuntos de datos relacionales tradicionales, Redis no tiene el concepto de tabla o columna. En cambio, Redis ofrece patrones de acceso clave-valor en los que la clave es esencialmente una `string` y el valor es una `string`, un `z-set` o un `hmap`.

Puede usar AWS Glue Data Catalog para crear esquemas y configurar tablas virtuales. Las propiedades especiales de la tabla indican al conector de Redis de Athena cómo asignar las claves y los valores de Redis a una tabla. Para obtener más información, consulte [Configuración de bases de datos y tablas en AWS Glue](#) más adelante en este documento.

Si Lake Formation está habilitado en la cuenta, el rol de IAM del conector de Lambda federado de Athena que haya implementado en AWS Serverless Application Repository debe tener acceso de lectura en Lake Formation para AWS Glue Data Catalog.

El conector Redis de Amazon Athena es compatible con Amazon MemoryDB para Redis y Amazon ElastiCache para Redis.

## Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).
- Antes de usar este conector, debe configurar una VPC y un grupo de seguridad. Para obtener más información, consulte [Creación de una VPC para un conector de origen de datos](#).

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de Redis.

- `spill_bucket`: especifica el bucket de Amazon S3 para los datos que superen los límites de la función de Lambda.
- `spill_prefix`: (opcional) de forma predeterminada, se establece una subcarpeta en la carpeta especificada `spill_bucket` llamada `athena-federation-spill`. Le recomendamos configurar un [ciclo de vida de almacenamiento](#) de Amazon S3 en esta ubicación para eliminar vertidos de más de un número predeterminado de días u horas.
- `spill_put_request_headers`: (opcional) un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud `putObject` de Amazon S3 que se usa para el vertido (por ejemplo, `{"x-amz-server-side-encryption" : "AES256"}`). Para ver otros encabezados posibles, consulte [PutObject](#) en la referencia de la API de Amazon Simple Storage Service.
- `kms_key_id`: (opcional) de forma predeterminada, los datos que se vierten a Amazon S3 se cifran mediante el modo de cifrado autenticado AES-GCM y una clave generada aleatoriamente. Para que la función de Lambda use claves de cifrado más seguras generadas por KMS, como `a7e63k4b-81oc-40db-a2a1-4d0en2cd8331`, puede especificar un ID de clave de KMS.
- `disable_spill_encryption`: (opcional) cuando se establece en `True`, desactiva el cifrado del vertido. El valor predeterminado es `False`, de modo que los datos que se vierten a S3 se cifran mediante AES-GCM, ya sea mediante una clave generada aleatoriamente o KMS para generar claves. La desactivación del cifrado de vertido puede mejorar el rendimiento, especialmente si su ubicación de vertido usa [cifrado del servidor](#).
- `glue_catalog`: (opcional) use esta opción para especificar un [catálogo de AWS Glue entre cuentas](#). De forma predeterminada, el conector intenta obtener los metadatos de su propia cuenta de AWS Glue.

## Configuración de bases de datos y tablas en AWS Glue

Para habilitar una tabla de AWS Glue con el fin de usarla con Redis, puede establecer en ella las siguientes propiedades: `redis-endpoint`, `redis-value-type` y ya sea `redis-keys-zset` o `redis-key-prefix`.

Además, cualquier base de datos de AWS Glue que contiene tablas de Redis debe tener una `redis-db-flag` en la propiedad del URI de la base de datos. Para establecer la propiedad `redis-db-flag` del URI, use la consola de AWS Glue para editar la base de datos.

En la siguiente lista se describen las propiedades de tabla.

- `redis-endpoint`: (obligatoria) *el nombre de host:el puerto:la contraseña* del servidor de Redis que contiene datos para esta tabla (por ejemplo, `athena-federation-demo.cache.amazonaws.com:6379`). De forma alternativa, puede almacenar el punto de conexión, o parte de este, en AWS Secrets Manager mediante `${Secret_Name}` como valor de propiedad de la tabla.

### Note

Para usar la función Consulta federada de Athena con AWS Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

- `redis-keys-zset`: (obligatoria si no se usa `redis-key-prefix`) una lista de claves separadas por comas, cuyo valor es un [zset](#) (por ejemplo, `active-orders,pending-orders`). Cada uno de los valores del zset se trata como una clave que forma parte de la tabla. Se debe establecer la propiedad `redis-keys-zset` o la propiedad `redis-key-prefix`.
- `redis-key-prefix`: (obligatoria si no se usa `redis-keys-zset`) una lista de prefijos de claves separados por comas para buscar valores en la tabla (por ejemplo, `accounts-*,acct-`). Se debe establecer la propiedad `redis-key-prefix` o la propiedad `redis-keys-zset`.
- `redis-value-type`: (obligatoria) determina cómo se asignan los valores de las claves definidas por `redis-key-prefix` o `redis-keys-zset` a la tabla. Se asigna un literal a una sola columna. Un zset también se asigna a una sola columna, pero cada clave puede almacenar varias filas. Un hash permite que cada clave sea una fila con varias columnas (por ejemplo, un hash, literal o zset).

- `redis-ssl-flag`: (opcional) cuando es `True`, se crea una conexión a Redis que usa SSL/TLS. El valor predeterminado es `False`.
- `redis-cluster-flag`: (opcional) cuando es `True`, se habilita compatibilidad con instancias de Redis en clúster. El valor predeterminado es `False`.
- `redis-db-number`: (opcional) se aplica solo para instancias independientes que no están en un clúster. Establezca este número (por ejemplo, 1, 2 o 3) para que se haga lectura de una base de datos de Redis no predeterminada. El valor predeterminado es la base de datos lógica 0 de Redis. Este número no hace referencia a una base de datos en Athena o AWS Glue, sino a una base de datos lógica de Redis. Para obtener más información, consulte [SELECT index](#) (Índice SELECT) en la documentación de Redis.

## Tipos de datos

El conector para Redis admite los siguientes tipos de datos. No se admiten los flujos de Redis.

- [Cadena](#)
- [Hash](#)
- Conjunto ordenado ([ZSet](#))

Todos los valores de Redis se recuperan como tipo de datos de la `string`. A continuación, se convierten en uno de los siguientes tipos de datos de Apache Arrow en función de cómo se definen las tablas en el AWS Glue Data Catalog.

Tipo de datos AWS Glue	Tipo de datos de Apache Arrow
<code>int</code>	<code>INT</code>
<code>cadena</code>	<code>VARCHAR</code>
<code>bigint</code>	<code>BIGINT</code>
<code>double</code>	<code>FLOAT8</code>
<code>float</code>	<code>FLOAT4</code>
<code>smallint</code>	<code>SMALLINT</code>

Tipo de datos AWS Glue	Tipo de datos de Apache Arrow
tinyint	TINYINT
boolean	BIT
binario	VARBINARY

## Permisos necesarios

Para obtener información completa sobre las políticas de IAM que requiere este conector, consulte la sección `Policies` del archivo [athena-redis.yaml](#). En la siguiente lista se resumen los permisos requeridos.

- **Acceso de escritura a Amazon S3:** el conector requiere acceso de escritura a una ubicación de Amazon S3 para volcar los resultados de consultas de gran tamaño.
- **Athena GetQueryExecution:** el conector usa este permiso para fallar rápidamente cuando finaliza la consulta ascendente de Athena.
- **AWS Glue Data Catalog:** el conector de Redis requiere acceso de solo lectura a AWS Glue Data Catalog para obtener información sobre el esquema.
- **Registros de CloudWatch:** el conector requiere acceso a Registros de CloudWatch para almacenar registros.
- **Acceso de lectura a AWS Secrets Manager:** si decide almacenar los detalles del punto de conexión de Redis en Secrets Manager, debe conceder al conector acceso a esos secretos.
- **Acceso a la VPC:** el conector requiere la capacidad de conectar y desconectar interfaces a la VPC para que pueda conectarse a ella y comunicarse con las instancias de Redis.

## Rendimiento

El conector de Redis de Athena intenta paralelizar las consultas con la instancia de Redis de acuerdo con el tipo de tabla que haya definido (por ejemplo, claves de zset o de prefijo).

El conector Redis de Athena inserta predicados para reducir los datos analizados en la consulta. Sin embargo, las consultas que contienen un predicado en la clave principal generan un error cuando se agota el tiempo de espera. Las cláusulas `LIMIT` reducen la cantidad de datos analizados; sin embargo, si no proporciona un predicado, es probable que las consultas `SELECT` con una cláusula

LIMIT analicen al menos 16 MB de datos. El conector para Redis resiste las limitaciones debidas a la simultaneidad.

### Información sobre licencias

El proyecto de conector de Redis de Amazon Athena está cubierto por la [Licencia de Apache-2.0](#).

### Recursos adicionales de

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

### Conector para Redshift de Amazon Athena

El conector de Amazon Athena Redshift permite a Amazon Athena acceder a las bases de datos de Amazon Redshift y Amazon Redshift sin servidor, incluidas las vistas de Redshift sin servidor. Se puede conectar a cualquiera de los servicios con los ajustes de configuración de la cadena de conexión JDBC que se describen en esta página.

### Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

### Limitaciones

- Las operaciones de escritura de DDL no son compatibles.
- En una configuración de multiplexor, el bucket de vertido y el prefijo se comparten en todas las instancias de base de datos.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.
- Dado que Redshift no admite particiones externas, todos los datos especificados por una consulta se recuperan cada vez.

### Términos

Los siguientes términos hacen referencia al conector de Redshift.

- Instancia de base de datos: cualquier instancia de base de datos implementada en las instalaciones, en Amazon EC2 o en Amazon RDS.
- Controlador: un controlador de Lambda que accede a la instancia de base de datos. Un controlador puede ser para los metadatos o para los registros de datos.
- Controlador de metadatos: un controlador de Lambda que recupera los metadatos de la instancia de base de datos.
- Controlador de registros: un controlador de Lambda que recupera registros de datos de la instancia de base de datos.
- Controlador compuesto: un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.
- Propiedad o parámetro: una propiedad de base de datos que usan los controladores para extraer información de la base de datos. Estas propiedades se configuran como variables de entorno de Lambda.
- Cadena de conexión: una cadena de texto que se usa para establecer una conexión con una instancia de base de datos.
- Catálogo: un catálogo ajeno a AWS Glue registrado en Athena que es un prefijo obligatorio para la propiedad `connection_string`.
- Controlador de multiplexación: un controlador de Lambda que puede aceptar y usar varias conexiones a bases de datos.

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de Redshift.

### Cadena de conexión

Use una cadena de conexión JDBC con el siguiente formato para conectarse a una instancia de base de datos.

```
redshift://${jdbc_connection_string}
```

### Uso de un controlador de multiplexación

Puede usar un multiplexor para conectarse a varias instancias de base de datos con una sola función de Lambda. Las solicitudes se enrutan por nombre de catálogo. Use las siguientes clases en Lambda.

Controlador	Clase
Controlador compuesto	RedshiftMuxCompositeHandler
Controlador de metadatos	RedshiftMuxMetadataHandler
Controlador de registros	RedshiftMuxRecordHandler

### Parámetros del controlador de multiplexación

Parámetro	Descripción
<code><i>\$catalog_</i>connection_string</code>	Obligatorio. Una cadena de conexión de instancia de base de datos. Agregue el prefijo a la variable de entorno con el nombre del catálogo usado en Athena. Por ejemplo, si el catálogo registrado en Athena es <code>myredshiftcatalog</code> , el nombre de la variable de entorno será <code>myredshiftcatalog_connection_string</code> .
default	Obligatorio. La cadena de conexión predeterminada. Esta cadena se usa cuando el catálogo es <code>lambda:\${AWS_LAMBDA_FUNCTION_NAME}</code> .

Las siguientes propiedades de ejemplo son para una función de Lambda de Redshift MUX que admite dos instancias de base de datos: `redshift1` (la predeterminada) y `redshift2`.

Propiedad	Valor
default	<code>redshift://jdbc:redshift://redshift1.host:5439/dev?user=sample2&amp;password=sample2</code>
<code>redshift_catalog1_connection_string</code>	<code>redshift://jdbc:redshift://redshift1.host:3306/default?\${Test/RDS/Redshift1}</code>



Propiedad	Valor
redshift_catalog2_connection_string	redshift://jdbc:redshift://redshift2.host:3333/default?user=sample2&password=sample2

## Proporcionar credenciales

Para proporcionar un nombre de usuario y una contraseña para la base de datos en la cadena de conexión JDBC, puede usar las propiedades de la cadena de conexión o AWS Secrets Manager.

- Cadena de conexión: se pueden especificar un nombre de usuario y una contraseña como propiedades en la cadena de conexión JDBC.

### Important

Como práctica recomendada en materia de seguridad, no utilice credenciales codificadas en las variables de entorno ni en las cadenas de conexión. Para obtener información sobre cómo transferir los secretos codificados a AWS Secrets Manager, consulte [Mover secretos codificados a AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

- AWS Secrets Manager: para utilizar la característica Consulta federada de Athena con AWS Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Puede poner el nombre de un secreto en AWS Secrets Manager, en la cadena de conexión JDBC. El conector reemplaza el nombre del secreto por los valores `username` y `password` de Secrets Manager.

Para las instancias de bases de datos de Amazon RDS, esta compatibilidad está estrechamente integrada. Si usa Amazon RDS, le recomendamos encarecidamente que use AWS Secrets Manager y la rotación de credenciales. Si la base de datos no usa Amazon RDS, guarde las credenciales como archivos JSON con el siguiente formato:

```
{"username": "${username}", "password": "${password}"}
```

## Ejemplo de cadena de conexión con nombre secreto

La siguiente cadena tiene el nombre secreto `${Test/RDS/Redshift1}`.

```
redshift://jdbc:redshift://redshift1.host:3306/default?...&${Test/RDS/Redshift1}&...
```

El conector usa el nombre secreto para recuperar los secretos y proporcionar el nombre de usuario y la contraseña, como en el siguiente ejemplo.

```
redshift://jdbc:redshift://redshift1.host:3306/
default?...&user=sample2&password=sample2&...
```

Actualmente, el conector de Redshift reconoce las propiedades `user` y `password` de JDBC.

### Parámetros de vertido

El SDK de Lambda puede verter datos en Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación.

Parámetro	Descripción
<code>spill_bucket</code>	Obligatorio. Nombre del bucket de vertido.
<code>spill_prefix</code>	Obligatorio. Prefijo de la clave del bucket de vertido.
<code>spill_put_request_headers</code>	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud <code>putObject</code> de Amazon S3 que se usa para el vertido (por ejemplo, <code>{"x-amz-server-side-encryption" : "AES256"}</code> ). Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.

### Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos correspondientes para JDBC y Apache Arrow.

JDBC	Arrow
Booleano	Bit

JDBC	Arrow
Entero	Pequeño
Short	Smallint
Entero	Int
Largo	Bigint
float	Float4
Doble	Float8
Date	DateDay
Marca de tiempo	DateMilli
Cadena	Varchar
Bytes	Varbinary
BigDecimal	Decimal
ARRAY	Enumeración

## Particiones y divisiones

Redshift no admite particiones externas. Para obtener información acerca de los problemas relacionados con el rendimiento, consulte [Rendimiento](#).

## Rendimiento

El conector Redshift de Athena inserta predicados para reducir los datos analizados en la consulta. Las cláusulas LIMIT, las cláusulas ORDER BY, los predicados simples y las expresiones complejas se insertan en el conector para reducir la cantidad de datos analizados y disminuir el tiempo de ejecución de las consultas. Sin embargo, seleccionar un subconjunto de columnas da como resultado que en ocasiones se prolongue el tiempo de ejecución de las consultas. Amazon Redshift es particularmente susceptible a la ralentización en la ejecución de consultas cuando se ejecutan varias de estas simultáneamente.

## Cláusulas LIMIT

Una instrucción `LIMIT N` reduce los datos analizados en la consulta. Con la inserción `LIMIT N`, el conector devuelve solo las filas `N` a Athena.

### Consultas `N` principales

Una consulta `N` principal especifica el orden del conjunto de resultados y un límite en la cantidad de filas devueltas. Puede usar este tipo de consulta para determinar los valores `N` principales máximos o `N` principales mínimos de sus conjuntos de datos. Con la inserción `N` principal, el conector devuelve solo las filas `N` ordenadas a Athena.

### Predicados

Un predicado es una expresión de la cláusula `WHERE` de una consulta SQL que da como resultado un valor booleano y filtra las filas en función de varias condiciones. El conector Redshift de Athena puede combinar estas expresiones e insertarlas directamente en Redshift para mejorar la funcionalidad y reducir la cantidad de datos analizados.

Los siguientes operadores del conector Redshift de Athena admiten la inserción de predicados:

- Booleano: `AND`, `OR`, `NOT`.
- Igualdad: `EQUAL`, `NOT_EQUAL`, `LESS_THAN`, `LESS_THAN_OR_EQUAL`, `GREATER_THAN`, `GREATER_THAN_OR_EQUAL`, `IS_DISTINCT_FROM`, `NULL_IF`, `IS_NULL`.
- Aritmética: `ADD`, `SUBTRACT`, `MULTIPLY`, `DIVIDE`, `MODULUS`, `NEGATE`.
- Otros: `LIKE_PATTERN`, `IN`.

### Ejemplo de inserción combinada

Para mejorar las capacidades de consulta, combine los tipos de inserciones, como en el siguiente ejemplo:

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
ORDER BY col_a DESC
LIMIT 10;
```

Para obtener información sobre cómo usar la inserción de predicados a fin de mejorar el rendimiento en las consultas federadas, incluido Amazon Redshift, consulte el artículo [Mejorar las consultas federadas con la inserción de predicados en Amazon Athena](#) en el Blog de macrodatos de AWS.

## Consultas de acceso directo

El conector Redshift admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.

Para usar consultas de acceso directo con Redshift, puede utilizar la siguiente sintaxis:

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'query string'  
    ))
```

El siguiente ejemplo de consulta envía una consulta a un origen de datos en Redshift. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## Recursos adicionales de

Para obtener la información más reciente sobre la versión del controlador JDBC, consulte el archivo [pom.xml](#) para el conector de Redshift en GitHub.com.

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector SAP HANA de Amazon Athena

### Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

## Limitaciones

- Las operaciones de escritura de DDL no son compatibles.
- En una configuración de multiplexor, el bucket de vertido y el prefijo se comparten en todas las instancias de base de datos.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.
- En SAP HANA, los nombres de los objetos se convierten a mayúsculas cuando se almacenan en la base de datos. Sin embargo, dado que los nombres entre comillas distinguen entre mayúsculas y minúsculas, es posible que dos tablas tengan el mismo nombre en minúsculas y mayúsculas (por ejemplo, EMPLOYEE y emp1oyee).

En Consulta federada de Athena, los nombres de las tablas de esquemas se proporcionan a la función de Lambda en minúsculas. Para evitar este problema, puede proporcionar sugerencias de consulta `@schemaCase` para recuperar los datos de las tablas que tienen nombres que distinguen entre mayúsculas y minúsculas. A continuación, se muestran dos consultas de ejemplo con sugerencias de consulta.

```
SELECT *  
FROM "lambda:saphanaconnector".SYSTEM."MY_TABLE@schemaCase=upper&tableCase=upper"
```

```
SELECT *  
FROM "lambda:saphanaconnector".SYSTEM."MY_TABLE@schemaCase=upper&tableCase=lower"
```

## Términos

Los siguientes términos hacen referencia al conector de SAP HANA.

- Instancia de base de datos: cualquier instancia de base de datos implementada en las instalaciones, en Amazon EC2 o en Amazon RDS.
- Controlador: un controlador de Lambda que accede a la instancia de base de datos. Un controlador puede ser para los metadatos o para los registros de datos.
- Controlador de metadatos: un controlador de Lambda que recupera los metadatos de la instancia de base de datos.

- **Controlador de registros:** un controlador de Lambda que recupera registros de datos de la instancia de base de datos.
- **Controlador compuesto:** un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.
- **Propiedad o parámetro:** una propiedad de base de datos que usan los controladores para extraer información de la base de datos. Estas propiedades se configuran como variables de entorno de Lambda.
- **Cadena de conexión:** una cadena de texto que se usa para establecer una conexión con una instancia de base de datos.
- **Catálogo:** un catálogo ajeno a AWS Glue registrado en Athena que es un prefijo obligatorio para la propiedad `connection_string`.
- **Controlador de multiplexación:** un controlador de Lambda que puede aceptar y usar varias conexiones a bases de datos.

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de SAP HANA.

## Cadena de conexión

Use una cadena de conexión JDBC con el siguiente formato para conectarse a una instancia de base de datos.

```
saphana://${jdbc_connection_string}
```

## Uso de un controlador de multiplexación

Puede usar un multiplexor para conectarse a varias instancias de base de datos con una sola función de Lambda. Las solicitudes se enrutan por nombre de catálogo. Use las siguientes clases en Lambda.

Controlador	Clase
Controlador compuesto	<code>SaphanaMuxCompositeHandler</code>
Controlador de metadatos	<code>SaphanaMuxMetadataHandler</code>
Controlador de registros	<code>SaphanaMuxRecordHandler</code>

## Parámetros del controlador de multiplexación

Parámetro	Descripción
<code>\$catalog_connection_string</code>	Obligatorio. Una cadena de conexión de instancia de base de datos. Agregue el prefijo a la variable de entorno con el nombre del catálogo usado en Athena. Por ejemplo, si el catálogo registrado en Athena es <code>mysaphanacatalog</code> , el nombre de la variable de entorno será <code>mysaphanacatalog_connection_string</code> .
<code>default</code>	Obligatorio. La cadena de conexión predeterminada. Esta cadena se usa cuando el catálogo es <code>lambda:\${AWS_LAMBDA_FUNCTION_NAME}</code> .

Las siguientes propiedades de ejemplo son para una función de Lambda de SAP HANA MUX que admite dos instancias de base de datos: `saphana1` (la predeterminada) y `saphana2`.

Propiedad	Valor
<code>default</code>	<code>saphana://jdbc:sap://saphana1.host:port/?\${Test/RDS/Saphana1}</code>
<code>saphana_catalog1_connection_string</code>	<code>saphana://jdbc:sap://saphana1.host:port/?\${Test/RDS/Saphana1}</code>
<code>saphana_catalog2_connection_string</code>	<code>saphana://jdbc:sap://saphana2.host:port/?user=sample2&amp;password=sample2</code>

### Proporcionar credenciales

Para proporcionar un nombre de usuario y una contraseña para la base de datos en la cadena de conexión JDBC, puede usar las propiedades de la cadena de conexión o AWS Secrets Manager.



- Cadena de conexión: se pueden especificar un nombre de usuario y una contraseña como propiedades en la cadena de conexión JDBC.

### Important

Como práctica recomendada en materia de seguridad, no utilice credenciales codificadas en las variables de entorno ni en las cadenas de conexión. Para obtener información sobre cómo transferir los secretos codificados a AWS Secrets Manager, consulte [Mover secretos codificados a AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

- AWS Secrets Manager: para utilizar la característica Consulta federada de Athena con AWS Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Puede poner el nombre de un secreto en AWS Secrets Manager, en la cadena de conexión JDBC. El conector reemplaza el nombre del secreto por los valores `username` y `password` de Secrets Manager.

Para las instancias de bases de datos de Amazon RDS, esta compatibilidad está estrechamente integrada. Si usa Amazon RDS, le recomendamos encarecidamente que use AWS Secrets Manager y la rotación de credenciales. Si la base de datos no usa Amazon RDS, guarde las credenciales como archivos JSON con el siguiente formato:

```
{"username": "${username}", "password": "${password}"}
```

### Ejemplo de cadena de conexión con nombre secreto

La siguiente cadena tiene el nombre secreto `${Test/RDS/Saphana1}`.

```
saphana://jdbc:sap://saphana1.host:port/?${Test/RDS/Saphana1}&...
```

El conector usa el nombre secreto para recuperar los secretos y proporcionar el nombre de usuario y la contraseña, como en el siguiente ejemplo.

```
saphana://jdbc:sap://saphana1.host:port/?user=sample2&password=sample2&...
```

Actualmente, el conector de SAP HANA reconoce las propiedades `user` y `password` de JDBC.

## Uso de un controlador de conexión única

Puede usar los siguientes controladores de registros y metadatos de conexión única para conectarse a una sola instancia de SAP HANA.

Tipo de controlador	Clase
Controlador compuesto	<code>SaphanaCompositeHandler</code>
Controlador de metadatos	<code>SaphanaMetadataHandler</code>
Controlador de registros	<code>SaphanaRecordHandler</code>

## Parámetros de controlador de conexión única

Parámetro	Descripción
<code>default</code>	Obligatorio. La cadena de conexión predeterminada.

Los controladores de conexión única admiten una instancia de base de datos y deben proporcionar un parámetro de cadena de conexión `default`. Se omiten todas las demás cadenas de conexión.

La siguiente propiedad de ejemplo es para una sola instancia de SAP HANA compatible con una función de Lambda.

Propiedad	Valor
<code>default</code>	<code>saphana://jdbc:sap://saphana1.host:port/?secret=Test/RDS/Saphana1</code>

## Parámetros de vertido

El SDK de Lambda puede verter datos en Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación.

Parámetro	Descripción
<code>spill_bucket</code>	Obligatorio. Nombre del bucket de vertido.
<code>spill_prefix</code>	Obligatorio. Prefijo de la clave del bucket de vertido.
<code>spill_put_request_headers</code>	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud <code>putObject</code> de Amazon S3 que se usa para el vertido (por ejemplo, <code>{"x-amz-server-side-encryption" : "AES256"}</code> ). Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.

### Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos correspondientes para JDBC y Apache Arrow.

JDBC	Arrow
Booleano	Bit
Entero	Pequeño
Short	Smallint
Entero	Int
Largo	Bigint
float	Float4
Doble	Float8
Date	DateDay
Marca de tiempo	DateMilli
Cadena	Varchar

JDBC	Arrow
Bytes	Varbinary
BigDecimal	Decimal
ARRAY	Enumeración

## Conversiones de tipos de datos

Además de las conversiones de JDBC a Arrow, el conector lleva a cabo otras conversiones para que los tipos de datos de origen de SAP HANA y Athena sean compatibles. Estas conversiones ayudan a garantizar que las consultas se ejecuten correctamente. En la siguiente tabla, se muestran estas conversiones.

Tipo de datos de origen (SAP HANA)	Tipo de datos convertido (Athena)
DECIMAL	BIGINT
INTEGER	INT
FECHA	DATEDAY
MARCA DE TIEMPO	DATEMILLI

Todos los demás tipos de datos no admitidos se convierten a VARCHAR.

## Particiones y divisiones

Una partición se representa mediante una sola columna de partición del tipo Integer. La columna contiene los nombres de las particiones definidas en una tabla de SAP HANA. Para una tabla que no tiene nombres de partición, se devuelve \*, que equivale a una sola partición. Una partición equivale a una división.

Nombre	Tipo	Descripción
PART_ID	Entero	Partición con nombre en SAP HANA.

## Rendimiento

SAP HANA admite particiones nativas. El conector SAP HANA de Athena puede recuperar datos de estas particiones en paralelo. Si quiere consultar conjuntos de datos muy grandes con una distribución uniforme de particiones, se recomienda encarecidamente la partición nativa. La selección de un subconjunto de columnas acelera significativamente el tiempo de ejecución de las consultas y reduce la cantidad de datos analizados. El conector presenta una limitación significativa y, a veces, genera errores en las consultas debido a la simultaneidad.

El conector SAP HANA de Athena inserta predicados para reducir los datos analizados en la consulta. Los predicados simples, las expresiones complejas y las cláusulas LIMIT se insertan en el conector para reducir la cantidad de datos analizados y disminuir el tiempo de ejecución de las consultas.

### Cláusulas LIMIT

Una instrucción LIMIT N reduce los datos analizados en la consulta. Con la inserción LIMIT N, el conector devuelve solo las filas N a Athena.

### Predicados

Un predicado es una expresión de la cláusula WHERE de una consulta SQL que da como resultado un valor booleano y filtra las filas en función de varias condiciones. El conector SAP HANA de Athena puede combinar estas expresiones e insertarlas directamente en SAP HANA para mejorar la funcionalidad y reducir la cantidad de datos analizados.

Los siguientes operadores del conector SAP HANA de Athena admiten la inserción de predicados:

- Booleano: AND, OR, NOT.
- Igualdad: EQUAL, NOT\_EQUAL, LESS\_THAN, LESS\_THAN\_OR\_EQUAL, GREATER\_THAN, GREATER\_THAN\_OR\_EQUAL, IS\_DISTINCT\_FROM, NULL\_IF, IS\_NULL.
- Aritmética: ADD, SUBTRACT, MULTIPLY, DIVIDE, MODULUS, NEGATE.
- Otros: LIKE\_PATTERN, IN.

### Ejemplo de inserción combinada

Para mejorar las capacidades de consulta, combine los tipos de inserciones, como en el siguiente ejemplo:

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
LIMIT 10;
```

## Consultas de acceso directo

El conector SAP HANA admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.

Para realizar consultas de acceso directo con SAP HANA, puede usar la siguiente sintaxis:

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  ))
```

El siguiente ejemplo de consulta envía una consulta a un origen de datos de SAP HANA. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.

```
SELECT * FROM TABLE(
  system.query(
    query => 'SELECT * FROM customer LIMIT 10'
  ))
```

## Información sobre licencias

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

Para obtener la información más reciente sobre la versión del controlador JDBC, consulte el archivo [pom.xml](#) para el conector de SAP HANA en GitHub.com.

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector Snowflake de Amazon Athena

El conector de [Snowflake](#) de Amazon Athena permite que Amazon Athena ejecute consultas SQL en los datos almacenados en la base de datos SQL de Snowflake o en instancias de RDS mediante JDBC.

### Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

### Limitaciones

- Las operaciones de escritura de DDL no son compatibles.
- En una configuración de multiplexor, el bucket de vertido y el prefijo se comparten en todas las instancias de base de datos.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.
- Actualmente, se admiten vistas de Snowflake con una sola división.
- En Snowflake, dado que los nombres de los objetos distinguen entre mayúsculas y minúsculas, dos tablas pueden tener el mismo nombre en minúsculas y mayúsculas (por ejemplo, EMPLOYEE y employee). En Consulta federada de Athena, los nombres de las tablas de esquemas se proporcionan a la función de Lambda en minúsculas. Para evitar este problema, puede proporcionar sugerencias de consulta @schemaCase para recuperar los datos de las tablas que tienen nombres que distinguen entre mayúsculas y minúsculas. A continuación, se muestran dos consultas de ejemplo con sugerencias de consulta.

```
SELECT *  
FROM "lambda:snowflakeconnector".SYSTEM."MY_TABLE@schemaCase=upper&tableCase=upper"
```

```
SELECT *  
FROM "lambda:snowflakeconnector".SYSTEM."MY_TABLE@schemaCase=upper&tableCase=lower"
```

## Términos

Los siguientes términos hacen referencia al conector de Snowflake.

- **Instancia de base de datos:** cualquier instancia de base de datos implementada en las instalaciones, en Amazon EC2 o en Amazon RDS.
- **Controlador:** un controlador de Lambda que accede a la instancia de base de datos. Un controlador puede ser para los metadatos o para los registros de datos.
- **Controlador de metadatos:** un controlador de Lambda que recupera los metadatos de la instancia de base de datos.
- **Controlador de registros:** un controlador de Lambda que recupera registros de datos de la instancia de base de datos.
- **Controlador compuesto:** un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.
- **Propiedad o parámetro:** una propiedad de base de datos que usan los controladores para extraer información de la base de datos. Estas propiedades se configuran como variables de entorno de Lambda.
- **Cadena de conexión:** una cadena de texto que se usa para establecer una conexión con una instancia de base de datos.
- **Catálogo:** un catálogo ajeno a AWS Glue registrado en Athena que es un prefijo obligatorio para la propiedad `connection_string`.
- **Controlador de multiplexación:** un controlador de Lambda que puede aceptar y usar varias conexiones a bases de datos.

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de Snowflake.

### Cadena de conexión

Use una cadena de conexión JDBC con el siguiente formato para conectarse a una instancia de base de datos.

```
snowflake://${jdbc_connection_string}
```



## Uso de un controlador de multiplexación

Puede usar un multiplexor para conectarse a varias instancias de base de datos con una sola función de Lambda. Las solicitudes se enrutan por nombre de catálogo. Use las siguientes clases en Lambda.

Controlador	Clase
Controlador compuesto	<code>SnowflakeMuxCompositeHandler</code>
Controlador de metadatos	<code>SnowflakeMuxMetadataHandler</code>
Controlador de registros	<code>SnowflakeMuxRecordHandler</code>

## Parámetros del controlador de multiplexación

Parámetro	Descripción
<code><i>\$catalog_connection_string</i></code>	Obligatorio. Una cadena de conexión de instancia de base de datos. Agregue el prefijo a la variable de entorno con el nombre del catálogo usado en Athena. Por ejemplo, si el catálogo registrado en Athena es <code>mysnowflakecatalog</code> , el nombre de la variable de entorno será <code>mysnowflakecatalog_connection_string</code> .
<code>default</code>	Obligatorio. La cadena de conexión predeterminada. Esta cadena se usa cuando el catálogo es <code>lambda:\${AWS_LAMBDA_FUNCTION_NAME}</code> .

Las siguientes propiedades de ejemplo son para una función de Lambda de Snowflake MUX que admite dos instancias de base de datos: `snowflake1` (la predeterminada) y `snowflake2`.

Propiedad	Valor
<code>default</code>	<code>snowflake://jdbc:snowflake://snowflake1.host:port/?warehouse=warehousename&amp;db=db1&amp;schema=schema1&amp;\${Test/RDS/Snowflake1}</code>

Propiedad	Valor
snowflake _catalog1 _connecti on_string	snowflake://jdbc:snowflake://snowflake1.host: port/?warehouse=warehousename&db=db1&schema=s chema1\${Test/RDS/Snowflake1}
snowflake _catalog2 _connecti on_string	snowflake://jdbc:snowflake://snowflake2.host: port/?warehouse=warehousename&db=db1&schema=s chema1&user=sample2&password=sample2

## Proporcionar credenciales

Para proporcionar un nombre de usuario y una contraseña para la base de datos en la cadena de conexión JDBC, puede usar las propiedades de la cadena de conexión o AWS Secrets Manager.

- Cadena de conexión: se pueden especificar un nombre de usuario y una contraseña como propiedades en la cadena de conexión JDBC.

### Important

Como práctica recomendada en materia de seguridad, no utilice credenciales codificadas en las variables de entorno ni en las cadenas de conexión. Para obtener información sobre cómo transferir los secretos codificados a AWS Secrets Manager, consulte [Mover secretos codificados a AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

- AWS Secrets Manager: para utilizar la característica Consulta federada de Athena con AWS Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Puede poner el nombre de un secreto en AWS Secrets Manager, en la cadena de conexión JDBC. El conector reemplaza el nombre del secreto por los valores `username` y `password` de Secrets Manager.

Para las instancias de bases de datos de Amazon RDS, esta compatibilidad está estrechamente integrada. Si usa Amazon RDS, le recomendamos encarecidamente que use AWS Secrets

Manager y la rotación de credenciales. Si la base de datos no usa Amazon RDS, guarde las credenciales como archivos JSON con el siguiente formato:

```
{"username": "${username}", "password": "${password}"}
```

### Ejemplo de cadena de conexión con nombre secreto

La siguiente cadena tiene el nombre secreto `${Test/RDS/Snowflake1}`.

```
snowflake://jdbc:snowflake://snowflake1.host:port/?
warehouse=warehousename&db=db1&schema=schema1${Test/RDS/Snowflake1}&...
```

El conector usa el nombre secreto para recuperar los secretos y proporcionar el nombre de usuario y la contraseña, como en el siguiente ejemplo.

```
snowflake://jdbc:snowflake://snowflake1.host:port/
warehouse=warehousename&db=db1&schema=schema1&user=sample2&password=sample2&...
```

Actualmente, Snowflake reconoce las propiedades `user` y `password` de JDBC. También acepta el nombre de usuario y la contraseña con el formato *nombre de usuario/contraseña* sin las claves `user` o `password`.

### Uso de un controlador de conexión única

Puede usar los siguientes controladores de registros y metadatos de conexión única para conectarse a una sola instancia de Snowflake.

Tipo de controlador	Clase
Controlador compuesto	<code>SnowflakeCompositeHandler</code>
Controlador de metadatos	<code>SnowflakeMetadataHandler</code>
Controlador de registros	<code>SnowflakeRecordHandler</code>

## Parámetros de controlador de conexión única

Parámetro	Descripción
<code>default</code>	Obligatorio. La cadena de conexión predeterminada.

Los controladores de conexión única admiten una instancia de base de datos y deben proporcionar un parámetro de cadena de conexión `default`. Se omiten todas las demás cadenas de conexión.

La siguiente propiedad de ejemplo es para una sola instancia de Snowflake compatible con una función de Lambda.

Propiedad	Valor
<code>default</code>	<code>snowflake://jdbc:snowflake://snowflake1.host:port/?secret=Test/RDS/Snowflake1</code>

## Parámetros de vertido

El SDK de Lambda puede verter datos en Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación.

Parámetro	Descripción
<code>spill_bucket</code>	Obligatorio. Nombre del bucket de vertido.
<code>spill_prefix</code>	Obligatorio. Prefijo de la clave del bucket de vertido.
<code>spill_put_request_headers</code>	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud <code>putObject</code> de Amazon S3 que se usa para el vertido (por ejemplo, <code>{"x-amz-server-side-encryption" : "AES256"}</code> ). Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.

## Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos correspondientes para JDBC y Apache Arrow.

JDBC	Arrow
Booleano	Bit
Entero	Pequeño
Short	Smallint
Entero	Int
Largo	Bigint
float	Float4
Doble	Float8
Date	DateDay
Marca de tiempo	DateMilli
Cadena	Varchar
Bytes	Varbinary
BigDecimal	Decimal
ARRAY	Enumeración

## Conversiones de tipos de datos

Además de las conversiones de JDBC a Arrow, el conector lleva a cabo otras conversiones para que los tipos de datos de origen de Snowflake y Athena sean compatibles. Estas conversiones ayudan a garantizar que las consultas se ejecuten correctamente. En la siguiente tabla, se muestran estas conversiones.

Tipo de datos de origen (Snowflake)	Tipo de datos convertido (Athena)
MARCA DE TIEMPO	TIMESTAMPMILLI
FECHA	TIMESTAMPMILLI
INTEGER	INT
DECIMAL	BIGINT
TIMESTAMP_NTZ	TIMESTAMPMILLI

Todos los demás tipos de datos no admitidos se convierten a VARCHAR.

## Particiones y divisiones

Las particiones se usan para determinar cómo generar divisiones para el conector. Athena crea una columna sintética de tipo `varchar` que representa el esquema de partición de la tabla para ayudar al conector a generar divisiones. El conector no modifica la definición de la tabla real.

Para crear esta columna sintética y las particiones, Athena requiere que se defina una clave principal. Sin embargo, dado que Snowflake no aplica restricciones de clave principal, deberá aplicar la unicidad por su cuenta. En caso contrario, Athena adoptará una única partición de forma predeterminada.

## Rendimiento

Para obtener un rendimiento óptimo, use filtros en las consultas siempre que sea posible. Además, recomendamos encarecidamente la creación de particiones nativas para recuperar enormes conjuntos de datos que tengan una distribución uniforme de particiones. La selección de un subconjunto de columnas acelera significativamente el tiempo de ejecución de las consultas y reduce la cantidad de datos analizados. El conector Snowflake resiste las limitaciones debidas a la simultaneidad.

El conector Snowflake de Athena inserta predicados para reducir los datos analizados en la consulta. Los predicados simples, las expresiones complejas y las cláusulas `LIMIT` se insertan en el conector para reducir la cantidad de datos analizados y disminuir el tiempo de ejecución de las consultas.

## Cláusulas LIMIT

Una instrucción `LIMIT N` reduce los datos analizados en la consulta. Con la inserción `LIMIT N`, el conector devuelve solo las filas `N` a Athena.

## Predicados

Un predicado es una expresión de la cláusula `WHERE` de una consulta SQL que da como resultado un valor booleano y filtra las filas en función de varias condiciones. El conector Snowflake de Athena puede combinar estas expresiones e insertarlas directamente en Snowflake para mejorar la funcionalidad y reducir la cantidad de datos analizados.

Los siguientes operadores del conector Snowflake de Athena admiten la inserción de predicados:

- Booleano: `AND`, `OR`, `NOT`.
- Igualdad: `EQUAL`, `NOT_EQUAL`, `LESS_THAN`, `LESS_THAN_OR_EQUAL`, `GREATER_THAN`, `GREATER_THAN_OR_EQUAL`, `IS_DISTINCT_FROM`, `NULL_IF`, `IS_NULL`.
- Aritmética: `ADD`, `SUBTRACT`, `MULTIPLY`, `DIVIDE`, `MODULUS`, `NEGATE`.
- Otros: `LIKE_PATTERN`, `IN`.

## Ejemplo de inserción combinada

Para mejorar las capacidades de consulta, combine los tipos de inserciones, como en el siguiente ejemplo:

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%')
LIMIT 10;
```

## Consultas de acceso directo

El conector Snowflake admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.

Para usar consultas de acceso directo con Snowflake, puede usar la siguiente sintaxis:

```
SELECT * FROM TABLE(
```

```
system.query(  
    query => 'query string'  
))
```

En el siguiente ejemplo de consulta, se envía una consulta a un origen de datos de Snowflake. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## Información sobre licencias

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

Para obtener la información más reciente sobre la versión del controlador JDBC, consulte el archivo [pom.xml](#) para el conector de Snowflake en GitHub.com.

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector Microsoft SQL Server de Amazon Athena

El conector de Amazon Athena de [Microsoft SQL Server](#) permite que Amazon Athena ejecute consultas SQL en los datos almacenados en Microsoft SQL Server mediante JDBC.

## Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

## Limitaciones

- Las operaciones de escritura de DDL no son compatibles.



- En una configuración de multiplexor, el bucket de vertido y el prefijo se comparten en todas las instancias de base de datos.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.
- En condiciones de filtro, debe enviar los tipos de datos Date y Timestamp al tipo de dato correspondiente.
- Para buscar valores negativos de tipo Real y Float, use el operador `<=` o `>=`.
- Los tipos de datos binary, varbinary, image y rowversion no son compatibles.

## Términos

Los siguientes términos hacen referencia al conector de SQL Server.

- Instancia de base de datos: cualquier instancia de base de datos implementada en las instalaciones, en Amazon EC2 o en Amazon RDS.
- Controlador: un controlador de Lambda que accede a la instancia de base de datos. Un controlador puede ser para los metadatos o para los registros de datos.
- Controlador de metadatos: un controlador de Lambda que recupera los metadatos de la instancia de base de datos.
- Controlador de registros: un controlador de Lambda que recupera registros de datos de la instancia de base de datos.
- Controlador compuesto: un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.
- Propiedad o parámetro: una propiedad de base de datos que usan los controladores para extraer información de la base de datos. Estas propiedades se configuran como variables de entorno de Lambda.
- Cadena de conexión: una cadena de texto que se usa para establecer una conexión con una instancia de base de datos.
- Catálogo: un catálogo ajeno a AWS Glue registrado en Athena que es un prefijo obligatorio para la propiedad `connection_string`.
- Controlador de multiplexación: un controlador de Lambda que puede aceptar y usar varias conexiones a bases de datos.

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de SQL Server.

### Cadena de conexión

Use una cadena de conexión JDBC con el siguiente formato para conectarse a una instancia de base de datos.

```
sqlserver://${jdbc_connection_string}
```

### Uso de un controlador de multiplexación

Puede usar un multiplexor para conectarse a varias instancias de base de datos con una sola función de Lambda. Las solicitudes se enrutan por nombre de catálogo. Use las siguientes clases en Lambda.

Controlador	Clase
Controlador compuesto	SqlServerMuxCompositeHandler
Controlador de metadatos	SqlServerMuxMetadataHandler
Controlador de registros	SqlServerMuxRecordHandler

### Parámetros del controlador de multiplexación

Parámetro	Descripción
<code>catalog_connection_string</code>	Obligatorio. Una cadena de conexión de instancia de base de datos. Agregue el prefijo a la variable de entorno con el nombre del catálogo usado en Athena. Por ejemplo, si el catálogo registrado en Athena es <code>mysqlservercatalog</code> , el nombre de la variable de entorno será <code>mysqlservercatalog_connection_string</code> .
<code>default</code>	Obligatorio. La cadena de conexión predeterminada. Esta cadena se usa cuando el catálogo es <code>lambda:\${AWS_LAMBDA_FUNCTION_NAME}</code> .

Las siguientes propiedades de ejemplo son para una función de Lambda de SQL Server MUX que admite dos instancias de base de datos: `sqlserver1` (la predeterminada) y `sqlserver2`.

Propiedad	Valor
<code>default</code>	<code>sqlserver://jdbc:sqlserver://sqlserver1. <i>hostname:port</i>;databaseName= &lt;database_name&gt; ;\${secret1_name }</code>
<code>sqlserver_catalog1_connection_string</code>	<code>sqlserver://jdbc:sqlserver://sqlserver1. <i>hostname:port</i>;databaseName= &lt;database_name&gt; ;\${secret1_name }</code>
<code>sqlserver_catalog2_connection_string</code>	<code>sqlserver://jdbc:sqlserver://sqlserver2. <i>hostname:port</i>;databaseName= &lt;database_name&gt; ;\${secret2_name }</code>

### Proporcionar credenciales

Para proporcionar un nombre de usuario y una contraseña para la base de datos en la cadena de conexión JDBC, puede usar las propiedades de la cadena de conexión o AWS Secrets Manager.

- Cadena de conexión: se pueden especificar un nombre de usuario y una contraseña como propiedades en la cadena de conexión JDBC.

#### Important

Como práctica recomendada en materia de seguridad, no utilice credenciales codificadas en las variables de entorno ni en las cadenas de conexión. Para obtener información sobre cómo transferir los secretos codificados a AWS Secrets Manager, consulte [Mover secretos codificados a AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

- AWS Secrets Manager: para utilizar la característica Consulta federada de Athena con AWS Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Puede poner el nombre de un secreto en AWS Secrets Manager, en la cadena de conexión JDBC. El conector reemplaza el nombre del secreto por los valores `username` y `password` de Secrets Manager.

Para las instancias de bases de datos de Amazon RDS, esta compatibilidad está estrechamente integrada. Si usa Amazon RDS, le recomendamos encarecidamente que use AWS Secrets Manager y la rotación de credenciales. Si la base de datos no usa Amazon RDS, guarde las credenciales como archivos JSON con el siguiente formato:

```
{"username": "${username}", "password": "${password}"}
```

### Ejemplo de cadena de conexión con nombre secreto

La siguiente cadena tiene el nombre secreto `${secret_name}`.

```
sqlserver://jdbc:sqlserver://hostname:port;databaseName=<database_name>;${secret_name}
```

El conector usa el nombre secreto para recuperar los secretos y proporcionar el nombre de usuario y la contraseña, como en el siguiente ejemplo.

```
sqlserver://
jdbc:sqlserver://
hostname:port;databaseName=<database_name>;user=<user>;password=<password>
```

### Uso de un controlador de conexión única

Puede usar los siguientes controladores de registros y metadatos de conexión única para conectarse a una sola instancia de SQL Server.

Tipo de controlador	Clase
Controlador compuesto	<code>SqlServerCompositeHandler</code>
Controlador de metadatos	<code>SqlServerMetadataHandler</code>
Controlador de registros	<code>SqlServerRecordHandler</code>

## Parámetros de controlador de conexión única

Parámetro	Descripción
default	Obligatorio. La cadena de conexión predeterminada.

Los controladores de conexión única admiten una instancia de base de datos y deben proporcionar un parámetro de cadena de conexión default. Se omiten todas las demás cadenas de conexión.

La siguiente propiedad de ejemplo es para una sola instancia de SQL Server compatible con una función de Lambda.

Propiedad	Valor
default	sqlserver://jdbc:sqlserver:// <i>hostname:port</i> ;database Name= <i>&lt;database_name&gt;</i> ;\${ <i>secret_name</i> }

## Parámetros de vertido

El SDK de Lambda puede verter datos en Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación.

Parámetro	Descripción
spill_bucket	Obligatorio. Nombre del bucket de vertido.
spill_prefix	Obligatorio. Prefijo de la clave del bucket de vertido.
spill_put_request_headers	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud putObject de Amazon S3 que se usa para el vertido (por ejemplo, {"x-amz-server-side-encryption" : "AES256"} ). Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.

## Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos correspondientes para SQL Server y Apache Arrow.

SQL Server	Arrow
bit	TINYINT
tinyint	SMALLINT
smallint	SMALLINT
int	INT
bigint	BIGINT
decimal	DECIMAL
numérico	FLOAT8
smallmoney	FLOAT8
money	DECIMAL
float[24]	FLOAT4
float[53]	FLOAT8
real	FLOAT4
datetime	Date(MILLISECOND)
datetime2	Date(MILLISECOND)
smalldatetime	Date(MILLISECOND)
date	Date(DAY)
hora	VARCHAR
datetimeoffset	Date(MILLISECOND)

SQL Server	Arrow
char[n]	VARCHAR
varchar[n/max]	VARCHAR
nchar[n]	VARCHAR
nvarchar[n/max]	VARCHAR
texto	VARCHAR
ntext	VARCHAR

## Particiones y divisiones

Una partición se representa mediante una sola columna de partición del tipo `varchar`. En el caso del conector de SQL Server, una función de partición determina cómo se aplican las particiones en la tabla. La información sobre la función de partición y el nombre de la columna se recuperan de la tabla de metadatos de SQL Server. A continuación, una consulta personalizada obtiene la partición. Las divisiones se crean en función del número de particiones distintas recibidas.

## Rendimiento

La selección de un subconjunto de columnas acelera significativamente el tiempo de ejecución de las consultas y reduce la cantidad de datos analizados. El conector SQL Server resiste las limitaciones debidas a la simultaneidad.

El conector SQL Server de Athena inserta predicados para reducir los datos analizados en la consulta. Los predicados simples y las expresiones complejas se insertan en el conector para reducir la cantidad de datos analizados y disminuir el tiempo de ejecución de las consultas.

## Predicados

Un predicado es una expresión de la cláusula `WHERE` de una consulta SQL que da como resultado un valor booleano y filtra las filas en función de varias condiciones. El conector SQL Server de Athena puede combinar estas expresiones e insertarlas directamente en SQL Server para mejorar la funcionalidad y reducir la cantidad de datos analizados.

Los siguientes operadores del conector SQL Server de Athena admiten la inserción de predicados:

- Booleano: AND, OR, NOT.
- Igualdad: EQUAL, NOT\_EQUAL, LESS\_THAN, LESS\_THAN\_OR\_EQUAL, GREATER\_THAN, GREATER\_THAN\_OR\_EQUAL, IS\_DISTINCT\_FROM, NULL\_IF, IS\_NULL.
- Aritmética: ADD, SUBTRACT, MULTIPLY, DIVIDE, MODULUS, NEGATE.
- Otros: LIKE\_PATTERN, IN.

### Ejemplo de inserción combinada

Para mejorar las capacidades de consulta, combine los tipos de inserciones, como en el siguiente ejemplo:

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%');
```

### Consultas de acceso directo

El conector de SQL Server admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.

Para realizar consultas directas con SQL Server, puede utilizar la siguiente sintaxis:

```
SELECT * FROM TABLE(
  system.query(
    query => 'query string'
  ))
```

El siguiente ejemplo de consulta envía una consulta a un origen de datos de SQL Server. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.

```
SELECT * FROM TABLE(
  system.query(
    query => 'SELECT * FROM customer LIMIT 10'
  ))
```



## Información sobre licencias

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

Para obtener la información más reciente sobre la versión del controlador JDBC, consulte el archivo [pom.xml](#) para el conector de SQL Server en GitHub.com.

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector Teradata de Amazon Athena

El conector de Teradata de Amazon Athena permite que este servicio ejecute consultas SQL en los datos almacenados en las bases de datos de Teradata.

## Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

## Limitaciones

- Las operaciones de escritura de DDL no son compatibles.
- En una configuración de multiplexor, el bucket de vertido y el prefijo se comparten en todas las instancias de base de datos.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.

## Términos

Los siguientes términos hacen referencia al conector de Teradata.

- Instancia de base de datos: cualquier instancia de base de datos implementada en las instalaciones, en Amazon EC2 o en Amazon RDS.

- **Controlador:** un controlador de Lambda que accede a la instancia de base de datos. Un controlador puede ser para los metadatos o para los registros de datos.
- **Controlador de metadatos:** un controlador de Lambda que recupera los metadatos de la instancia de base de datos.
- **Controlador de registros:** un controlador de Lambda que recupera registros de datos de la instancia de base de datos.
- **Controlador compuesto:** un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.
- **Propiedad o parámetro:** una propiedad de base de datos que usan los controladores para extraer información de la base de datos. Estas propiedades se configuran como variables de entorno de Lambda.
- **Cadena de conexión:** una cadena de texto que se usa para establecer una conexión con una instancia de base de datos.
- **Catálogo:** un catálogo ajeno a AWS Glue registrado en Athena que es un prefijo obligatorio para la propiedad `connection_string`.
- **Controlador de multiplexación:** un controlador de Lambda que puede aceptar y usar varias conexiones a bases de datos.

### Requisito previo de la capa de Lambda

Para utilizar el conector Teradata con Athena, debe crear una capa de Lambda que incluya el controlador JDBC de Teradata. Una capa de Lambda es un archivo `.zip` que contiene un código adicional para una función de Lambda. Cuando implementa el conector Teradata en su cuenta, debe especificar el ARN de la capa. Esto adjunta la capa de Lambda junto con el controlador JDBC de Teradata al conector Teradata para que pueda utilizarse con Athena.

Para obtener más información sobre las capas de Lambda, consulte [Creación y uso compartido de capas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.

Para crear una capa de Lambda para el conector Teradata

1. Diríjase a la página de descarga del controlador JDBC de Teradata en <https://downloads.teradata.com/download/connectivity/jdbc-driver>.
2. Descargue el controlador JDBC de Teradata. El sitio web requiere que cree una cuenta y acepte un acuerdo de licencia para descargar el archivo.
3. Extraiga el archivo `terajdbc4.jar` desde el archivo que descargó.

4. Cree la siguiente estructura de carpetas y coloque el archivo `.jar` en ella.  

```
java\lib\terajdbc4.jar
```
5. Cree un archivo `.zip` de toda la estructura de carpetas que contiene el archivo `terajdbc4.jar`.
6. Inicie sesión en la AWS Management Console y abra la consola AWS Lambda en <https://console.aws.amazon.com/lambda/>.
7. En el panel de navegación, elija Capas y luego, elija Crear capa.
8. En Nombre, ingrese un nombre para la capa (por ejemplo, `TeradataJava11LambdaLayer`).
9. Asegúrese de que la opción Cargar un archivo `.zip` esté seleccionada.
10. Elija Cargar y, luego, cargue la carpeta comprimida que contiene el controlador JDBC de Teradata.
11. Seleccione Crear.
12. En la página de detalles de la capa, copie el ARN de la capa mediante la elección del icono del portapapeles en la parte superior de la página.
13. Guarde el ARN como referencia.

## Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de Teradata.

## Cadena de conexión

Use una cadena de conexión JDBC con el siguiente formato para conectarse a una instancia de base de datos.

```
teradata://${jdbc_connection_string}
```

## Uso de un controlador de multiplexación

Puede usar un multiplexor para conectarse a varias instancias de base de datos con una sola función de Lambda. Las solicitudes se enrutan por nombre de catálogo. Use las siguientes clases en Lambda.

Controlador	Clase
Controlador compuesto	<code>TeradataMuxCompositeHandler</code>

Controlador	Clase
Controlador de metadatos	TeradataMuxMetadataHandler
Controlador de registros	TeradataMuxRecordHandler

### Parámetros del controlador de multiplexación

Parámetro	Descripción
<code>\$<i>catalog</i>_connection_string</code>	Obligatorio. Una cadena de conexión de instancia de base de datos. Agregue el prefijo a la variable de entorno con el nombre del catálogo usado en Athena. Por ejemplo, si el catálogo registrado en Athena es <code>myteradatalog</code> , el nombre de la variable de entorno será <code>myteradatalog_connection_string</code> .
<code>default</code>	Obligatorio. La cadena de conexión predeterminada. Esta cadena se usa cuando el catálogo es <code>lambda:\${ AWS_LAMBDA_FUNCTION_NAME }</code> .

Las siguientes propiedades de ejemplo son para una función de Lambda de Teradata MUX que admite dos instancias de base de datos: `teradata1` (la predeterminada) y `teradata2`.

Propiedad	Valor
<code>default</code>	<code>teradata://jdbc:teradata://teradata2.host/TMODE=ANSI,CHARSET=UTF8,DATABASE=TEST,user=sample2&amp;password=sample2</code>
<code>teradata_catalog1_connection_string</code>	<code>teradata://jdbc:teradata://teradata1.host/TMODE=ANSI,CHARSET=UTF8,DATABASE=TEST,\${Test/RDS/Teradata1}</code>

Propiedad	Valor
teradata_catalog2_connection_string	teradata://jdbc:teradata://teradata2.host/TMODE=ANSI,C HARSET=UTF8,DATABASE=TEST,u ser=sample2&password=sample2

## Proporcionar credenciales

Para proporcionar un nombre de usuario y una contraseña para la base de datos en la cadena de conexión JDBC, puede usar las propiedades de la cadena de conexión o AWS Secrets Manager.

- Cadena de conexión: se pueden especificar un nombre de usuario y una contraseña como propiedades en la cadena de conexión JDBC.

### Important

Como práctica recomendada en materia de seguridad, no utilice credenciales codificadas en las variables de entorno ni en las cadenas de conexión. Para obtener información sobre cómo transferir los secretos codificados a AWS Secrets Manager, consulte [Mover secretos codificados a AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

- AWS Secrets Manager: para utilizar la característica Consulta federada de Athena con AWS Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Puede poner el nombre de un secreto en AWS Secrets Manager, en la cadena de conexión JDBC. El conector reemplaza el nombre del secreto por los valores `username` y `password` de Secrets Manager.

Para las instancias de bases de datos de Amazon RDS, esta compatibilidad está estrechamente integrada. Si usa Amazon RDS, le recomendamos encarecidamente que use AWS Secrets Manager y la rotación de credenciales. Si la base de datos no usa Amazon RDS, guarde las credenciales como archivos JSON con el siguiente formato:

```
{"username": "${username}", "password": "${password}"}
```

## Ejemplo de cadena de conexión con nombre secreto

La siguiente cadena tiene el nombre secreto `${Test/RDS/Teradata1}`.

```
teradata://jdbc:teradata1.host/TMODE=ANSI,CHARSET=UTF8,DATABASE=TEST,${Test/RDS/
Teradata1}&...
```

El conector usa el nombre secreto para recuperar los secretos y proporcionar el nombre de usuario y la contraseña, como en el siguiente ejemplo.

```
teradata://jdbc:teradata://teradata1.host/
TMODE=ANSI,CHARSET=UTF8,DATABASE=TEST,...&user=sample2&password=sample2&...
```

Actualmente, Teradata reconoce las propiedades `user` y `password` de JDBC. También acepta el nombre de usuario y la contraseña con el formato *nombre de usuario/contraseña* sin las claves `user` o `password`.

### Uso de un controlador de conexión única

Puede usar los siguientes controladores de registros y metadatos de conexión única para conectarse a una sola instancia de Teradata.

Tipo de controlador	Clase
Controlador compuesto	<code>TeradataCompositeHandler</code>
Controlador de metadatos	<code>TeradataMetadataHandler</code>
Controlador de registros	<code>TeradataRecordHandler</code>

### Parámetros de controlador de conexión única

Parámetro	Descripción
<code>default</code>	Obligatorio. La cadena de conexión predeterminada.

Los controladores de conexión única admiten una instancia de base de datos y deben proporcionar un parámetro de cadena de conexión default. Se omiten todas las demás cadenas de conexión.

La siguiente propiedad de ejemplo es para una sola instancia de Teradata compatible con una función de Lambda.

Propiedad	Valor
default	teradata://jdbc:teradata://teradata1.host/TMODE=ANSI,CHARSET=UTF8,DATABASE=TEST,secret=Test/RDS/Teradata1

### Parámetros de vertido

El SDK de Lambda puede verter datos en Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación.

Parámetro	Descripción
spill_bucket	Obligatorio. Nombre del bucket de vertido.
spill_prefix	Obligatorio. Prefijo de la clave del bucket de vertido.
spill_put_request_headers	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud putObject de Amazon S3 que se usa para el vertido (por ejemplo, {"x-amz-server-side-encryption" : "AES256"} ). Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.

### Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos correspondientes para JDBC y Apache Arrow.

JDBC	Arrow
Booleano	Bit
Entero	Pequeño
Short	Smallint
Entero	Int
Largo	Bigint
float	Float4
Doble	Float8
Date	DateDay
Marca de tiempo	DateMilli
Cadena	Varchar
Bytes	Varbinary
BigDecimal	Decimal
ARRAY	Enumeración

## Particiones y divisiones

Una partición se representa mediante una sola columna de partición del tipo `Integer`. La columna contiene los nombres de las particiones definidas en una tabla de Teradata. Para una tabla que no tiene nombres de partición, se devuelve `*`, que equivale a una sola partición. Una partición equivale a una división.

Nombre	Tipo	Descripción
Partición	Entero	Partición con nombre en Teradata.



## Rendimiento

Teradata admite particiones nativas. El conector Teradata de Athena puede recuperar datos de estas particiones en paralelo. Si quiere consultar conjuntos de datos muy grandes con una distribución uniforme de particiones, se recomienda encarecidamente la partición nativa. La selección de un subconjunto de columnas ralentiza significativamente el tiempo de ejecución de las consultas. El conector presenta ciertas limitaciones debido a la simultaneidad.

El conector Teradata de Athena inserta predicados para reducir los datos analizados en la consulta. Los predicados simples y las expresiones complejas se insertan en el conector para reducir la cantidad de datos analizados y disminuir el tiempo de ejecución de las consultas.

## Predicados

Un predicado es una expresión de la cláusula WHERE de una consulta SQL que da como resultado un valor booleano y filtra las filas en función de varias condiciones. El conector Teradata de Athena puede combinar estas expresiones e insertarlas directamente en Teradata para mejorar la funcionalidad y reducir la cantidad de datos analizados.

Los siguientes operadores del conector Teradata de Athena admiten la inserción de predicados:

- Booleano: AND, OR, NOT.
- Igualdad: EQUAL, NOT\_EQUAL, LESS\_THAN, LESS\_THAN\_OR\_EQUAL, GREATER\_THAN, GREATER\_THAN\_OR\_EQUAL, NULL\_IF, IS\_NULL.
- Aritmética: ADD, SUBTRACT, MULTIPLY, DIVIDE, MODULUS, NEGATE.
- Otros: LIKE\_PATTERN, IN.

## Ejemplo de inserción combinada

Para mejorar las capacidades de consulta, combine los tipos de inserciones, como en el siguiente ejemplo:

```
SELECT *
FROM my_table
WHERE col_a > 10
      AND ((col_a + col_b) > (col_c % col_d))
      AND (col_e IN ('val1', 'val2', 'val3') OR col_f LIKE '%pattern%');
```

## Consultas de acceso directo

El conector Teradata admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.

Para usar consultas de acceso directo con Teradata, puede utilizar la siguiente sintaxis:

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'query string'  
    ))
```

En el siguiente ejemplo de consulta, se envía una consulta a un origen de datos de Teradata. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## Información sobre licencias

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

Para obtener la información más reciente sobre la versión del controlador JDBC, consulte el archivo [pom.xml](#) para el conector de Teradata en GitHub.com.

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector para Timestream de Amazon Athena

El conector de Timestream de Amazon Athena permite que este servicio se comuniquen con [Amazon Timestream](#), lo que permite el acceso a los datos de serie temporal a través de Amazon Athena. De manera opcional, puede utilizar AWS Glue Data Catalog como fuente de metadatos complementarios.

Amazon Timestream es una base de datos de serie temporal rápida, escalable, completamente administrada y diseñada específicamente que facilita el almacenamiento y el análisis de billones de puntos de datos de serie temporal por día. Timestream le ahorra tiempo y costo en la administración del ciclo de vida de los datos de serie temporal al mantener los datos recientes en la memoria y mover los datos históricos a un nivel de almacenamiento de información de costo optimizado en función de las políticas definidas por el usuario.

Si Lake Formation está habilitado en la cuenta, el rol de IAM del conector de Lambda federado de Athena que haya implementado en AWS Serverless Application Repository debe tener acceso de lectura en Lake Formation para AWS Glue Data Catalog.

### Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

### Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de Timestream.

- `spill_bucket`: especifica el bucket de Amazon S3 para los datos que superen los límites de la función de Lambda.
- `spill_prefix`: (opcional) de forma predeterminada, se establece una subcarpeta en la carpeta especificada `spill_bucket` llamada `athena-federation-spill`. Le recomendamos configurar un [ciclo de vida de almacenamiento](#) de Amazon S3 en esta ubicación para eliminar vertidos de más de un número predeterminado de días u horas.
- `spill_put_request_headers`: (opcional) un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud `putObject` de Amazon S3 que se usa para el vertidos (por ejemplo, `{"x-amz-server-side-encryption" : "AES256"}`). Para ver otros encabezados posibles, consulte [PutObject](#) en la referencia de la API de Amazon Simple Storage Service.
- `kms_key_id`: (opcional) de forma predeterminada, los datos que se vierten a Amazon S3 se cifran mediante el modo de cifrado autenticado AES-GCM y una clave generada aleatoriamente. Para que la función de Lambda use claves de cifrado más seguras generadas por KMS, como `a7e63k4b-81oc-40db-a2a1-4d0en2cd8331`, puede especificar un ID de clave de KMS.

- `disable_spill_encryption`: (opcional) cuando se establece en `True`, desactiva el cifrado del vertido. El valor predeterminado es `False`, de modo que los datos que se vierten a S3 se cifran mediante AES-GCM, ya sea mediante una clave generada aleatoriamente o KMS para generar claves. La desactivación del cifrado de vertido puede mejorar el rendimiento, especialmente si su ubicación de vertido usa [cifrado del servidor](#).
- `glue_catalog`: (opcional) use esta opción para especificar un [catálogo de AWS Glue entre cuentas](#). De forma predeterminada, el conector intenta obtener los metadatos de su propia cuenta de AWS Glue.

## Configuración de bases de datos y tablas en AWS Glue

De manera opcional, puede usar AWS Glue Data Catalog como fuente de metadatos complementarios. Para habilitar una tabla de AWS Glue con la intención de usarla con Timestream, debe disponer de una tabla y una base de datos de AWS Glue con nombres que coincidan con la base de datos y la tabla de Timestream a los que quiere proporcionar metadatos complementarios.

### Note

Para obtener el mejor rendimiento, use solo minúsculas en los nombres de las bases de datos y tablas. El uso combinado de mayúsculas y minúsculas hace que el conector realice una búsqueda que no distinga mayúsculas de minúsculas, lo que requiere un mayor esfuerzo computacional.

Para configurar una tabla de AWS Glue con la intención de usarla con Timestream, debe establecer las propiedades de la tabla en AWS Glue.

Para usar una tabla de AWS Glue para metadatos complementarios

1. Edite la tabla en la consola de AWS Glue para agregar las siguientes propiedades de tabla:
  - `timestream-metadata-flag`: esta propiedad indica al conector de Timestream que este puede usar la tabla para obtener metadatos complementarios. Puede proporcionar cualquier valor para `timestream-metadata-flag` siempre y cuando la `timestream-metadata-flag` esté presente en la lista de propiedades de la tabla.
  - `_view_template`: cuando use AWS Glue para obtener metadatos complementarios, puede usar esta propiedad de tabla y especificar cualquier SQL de Timestream como vista. El conector de Timestream de Athena usa el SQL de la vista junto con el SQL de Athena para ejecutar la

consulta. Esta opción es útil si quiere usar una característica de SQL de Timestream que no está disponible en Athena de otro modo.

2. Asegúrese de usar los tipos de datos adecuados para AWS Glue, como se indica en este documento.

## Tipos de datos

Actualmente, el conector de Timestream solo admite un subconjunto de los tipos de datos disponibles en Timestream, específicamente: los valores escalares `varchar`, `double` y `timestamp`.

Para consultar el tipo de datos `timeseries`, debe configurar una vista en las propiedades de la tabla de AWS Glue que use la función `CREATE_TIME_SERIES` de Timestream. También debe proporcionar un esquema para la vista que use la sintaxis `ARRAY<STRUCT<time:timestamp,measure_value::double:double>>` como el tipo de cualquiera de las columnas de la serie temporal. Asegúrese de reemplazar `double` por el tipo escalar adecuado para la tabla.

En la siguiente imagen, se muestra un ejemplo de propiedades de tabla de AWS Glue configuradas para crear una vista en una serie temporal.

Tables > my\_timeseries Last updated 6 May 2020 Table Version (Current version) ▾

[Edit table](#) [Delete table](#) [View properties](#) [Compare versions](#) [Edit schema](#)

**Name** my\_timeseries  
**Description**  
**Database** virtuoso  
**Classification** parquet  
**Location** [s3://fake-path/](#)  
**Connection**  
**Deprecated** No  
**Last updated** Wed May 06 16:01:00 GMT-400 2020  
**Input format** org.apache.hadoop.hive.q1.io.parquet.MapredParquetInputFormat  
**Output format** org.apache.hadoop.hive.q1.io.parquet.MapredParquetOutputFormat  
**Serde serialization lib** org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe

**Serde parameters** serialization.format 1

**Table properties**

- timestream-metadata-flag timestream-metadata-flag
- \_view\_template

```
select az, hostname, region, CREATE_TIME_SERIES(time, measure_value::double) as cpu_utilization from
virtuoso.virtuoso WHERE measure_name = 'cpu_utilization' GROUP BY measure_name, az, hostname, region
```

## Permisos necesarios

Para obtener información completa sobre las políticas de IAM que requiere este conector, consulte la sección Políticas del archivo [athena-timestream.yaml](#). En la siguiente lista se resumen los permisos requeridos.

- Acceso de escritura a Amazon S3: el conector requiere acceso de escritura a una ubicación de Amazon S3 para volcar los resultados de consultas de gran tamaño.
- Athena GetQueryExecution: el conector usa este permiso para fallar rápidamente cuando finaliza la consulta ascendente de Athena.
- AWS Glue Data Catalog: el conector de Timestream requiere acceso de solo lectura a AWS Glue Data Catalog para obtener información sobre el esquema.
- Registros de CloudWatch: el conector requiere acceso a Registros de CloudWatch para almacenar registros.
- Acceso a Timestream: para ejecutar consultas de Timestream.

## Rendimiento

Le recomendamos usar la cláusula LIMIT para limitar los datos devueltos (no los datos analizados) a menos de 256 MB para garantizar que las consultas interactivas tengan un buen rendimiento.

El conector Timestream de Athena inserta predicados para reducir los datos analizados en la consulta. Las cláusulas LIMIT reducen la cantidad de datos analizados; sin embargo, si no proporciona un predicado, debe tener en cuenta que las consultas SELECT con una cláusula LIMIT analizan al menos 16 MB de datos. La selección de un subconjunto de columnas acelera significativamente el tiempo de ejecución de las consultas y reduce la cantidad de datos analizados. El conector para Timestream resiste las limitaciones debidas a la simultaneidad.

## Consultas de acceso directo

El conector para Timestream admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.

Para usar consultas de acceso directo con Timestream, puede utilizar la siguiente sintaxis:

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'query string'  
    ))
```

En el siguiente ejemplo de consulta, se envía una consulta a un origen de datos de Timestream. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## Información sobre licencias

El proyecto de conector de Timestream de Amazon Athena está cubierto por la [Licencia de Apache-2.0](#).

## Recursos adicionales de

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector para TPC Benchmark DS (TPC-DS) de Amazon Athena

El conector de TPC-DS de Amazon Athena permite que este servicio se comuniquen con un origen de datos de TPC Benchmark DS generados aleatoriamente para usarlo en evaluaciones comparativas y pruebas de funcionamiento de la federación de Athena. El conector Athena TPC-DS genera una base de datos compatible con TPC-DS en uno de cuatro factores de escala. No recomendamos usar este conector como alternativa a las pruebas de rendimiento de lagos de datos basados en Amazon S3.

### Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).

### Parámetros

Use las variables de entorno de Lambda de esta sección para configurar el conector de TPC-DS.

- `spill_bucket`: especifica el bucket de Amazon S3 para los datos que superen los límites de la función de Lambda.
- `spill_prefix`: (opcional) de forma predeterminada, se establece una subcarpeta en la carpeta especificada `spill_bucket` llamada `athena-federation-spill`. Le recomendamos configurar un [ciclo de vida de almacenamiento](#) de Amazon S3 en esta ubicación para eliminar vertidos de más de un número predeterminado de días u horas.
- `spill_put_request_headers`: (opcional) un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud `putObject` de Amazon S3 que se usa para el vertidos (por ejemplo, `{"x-amz-server-side-encryption" : "AES256"}`). Para ver otros encabezados posibles, consulte [PutObject](#) en la referencia de la API de Amazon Simple Storage Service.
- `kms_key_id`: (opcional) de forma predeterminada, los datos que se vierten a Amazon S3 se cifran mediante el modo de cifrado autenticado AES-GCM y una clave generada aleatoriamente. Para que la función de Lambda use claves de cifrado más seguras generadas por KMS, como `a7e63k4b-81oc-40db-a2a1-4d0en2cd8331`, puede especificar un ID de clave de KMS.
- `disable_spill_encryption`: (opcional) cuando se establece en `True`, desactiva el cifrado del vertido. El valor predeterminado es `False`, de modo que los datos que se vierten a S3 se cifran mediante AES-GCM, ya sea mediante una clave generada aleatoriamente o KMS para generar claves. La



desactivación del cifrado de vertido puede mejorar el rendimiento, especialmente si su ubicación de vertido usa [cifrado del servidor](#).

## Prueba de bases de datos y tablas

El conector de TPC-DS de Athena genera una base de datos compatible con TPC-DS en uno de cuatro factores de escala: `tpcds1`, `tpcds10`, `tpcds100`, `tpcds250` o `tpcds1000`.

## Resumen de tablas

Para obtener una lista completa de las tablas y columnas de datos de prueba, ejecute las consultas `SHOW TABLES` o `DESCRIBE TABLE`. Se proporciona el siguiente resumen de tablas para mayor comodidad.

1. `call_center`
2. `catalog_page`
3. `catalog_returns`
4. `catalog_sales`
5. `cliente`
6. `customer_address`
7. `customer_demographics`
8. `date_dim`
9. `dbgen_version`
10. `household_demographics`
11. `income_band`
12. `inventario`
13. `elemento`
14. `promotion`
15. `reason`
16. `ship_mode`
17. `almacén`
18. `store_returns`

19.store\_sales

20.time\_dim

21.almacén

22.web\_page

23.web\_returns

24.web\_sales

25.web\_site

Para las consultas de TPC-DS que sean compatibles con este esquema y datos generados, consulte el directorio [athena-tpcds/src/main/resources/queries/](https://github.com/awslabs/athena-tpcds/src/main/resources/queries/) en GitHub.

### Consulta de ejemplo

El siguiente ejemplo de consulta SELECT revisa en el catálogo de tpcds los datos demográficos de clientes de condados específicos.

```
SELECT
  cd_gender,
  cd_marital_status,
  cd_education_status,
  count(*) cnt1,
  cd_purchase_estimate,
  count(*) cnt2,
  cd_credit_rating,
  count(*) cnt3,
  cd_dep_count,
  count(*) cnt4,
  cd_dep_employed_count,
  count(*) cnt5,
  cd_dep_college_count,
  count(*) cnt6
FROM
  "lambda:tpcds".tpcds1.customer c, "lambda:tpcds".tpcds1.customer_address ca,
  "lambda:tpcds".tpcds1.customer_demographics
WHERE
  c.c_current_addr_sk = ca.ca_address_sk AND
  ca_county IN ('Rush County', 'Toole County', 'Jefferson County',
               'Dona Ana County', 'La Porte County') AND
  cd_demo_sk = c.c_current_cdemo_sk AND
  exists(SELECT *
```

```

FROM "lambda:tpcds".tpcds1.store_sales, "lambda:tpcds".tpcds1.date_dim
WHERE c.c_customer_sk = ss_customer_sk AND
      ss_sold_date_sk = d_date_sk AND
      d_year = 2002 AND
      d_moy BETWEEN 1 AND 1 + 3) AND
(exists(SELECT *
        FROM "lambda:tpcds".tpcds1.web_sales, "lambda:tpcds".tpcds1.date_dim
        WHERE c.c_customer_sk = ws_bill_customer_sk AND
              ws_sold_date_sk = d_date_sk AND
              d_year = 2002 AND
              d_moy BETWEEN 1 AND 1 + 3) OR
exists(SELECT *
        FROM "lambda:tpcds".tpcds1.catalog_sales, "lambda:tpcds".tpcds1.date_dim
        WHERE c.c_customer_sk = cs_ship_customer_sk AND
              cs_sold_date_sk = d_date_sk AND
              d_year = 2002 AND
              d_moy BETWEEN 1 AND 1 + 3))
GROUP BY cd_gender,
         cd_marital_status,
         cd_education_status,
         cd_purchase_estimate,
         cd_credit_rating,
         cd_dep_count,
         cd_dep_employed_count,
         cd_dep_college_count
ORDER BY cd_gender,
         cd_marital_status,
         cd_education_status,
         cd_purchase_estimate,
         cd_credit_rating,
         cd_dep_count,
         cd_dep_employed_count,
         cd_dep_college_count
LIMIT 100

```

## Permisos necesarios

Para obtener información completa sobre las políticas de IAM que requiere este conector, consulte la sección **Políticas** del archivo [athena-tpcds.yaml](#). En la siguiente lista se resumen los permisos requeridos.

- **Acceso de escritura a Amazon S3:** el conector requiere acceso de escritura a una ubicación de Amazon S3 para volcar los resultados de consultas de gran tamaño.

- Athena GetQueryExecution: el conector usa este permiso para fallar rápidamente cuando finaliza la consulta ascendente de Athena.

## Rendimiento

El conector de TPC-DS de Athena intenta paralelizar las consultas en función del factor de escala que elija. La inserción de predicados se lleva a cabo dentro de la función de Lambda.

## Información sobre licencias

El proyecto de conector de TPC-DS de Amazon Athena está cubierto por la [Licencia de Apache-2.0](#).

## Recursos adicionales de

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com.

## Conector para Vertica de Amazon Athena

Vertica es una plataforma de bases de datos en columnas que se puede implementar en la nube o en las instalaciones que admite almacenamiento de datos a escala de exabyte. Puede utilizar el conector Amazon Athena Vertica en consultas federadas para consultar orígenes de datos Vertica desde Athena. Por ejemplo, puede ejecutar consultas analíticas a través de un almacenamiento de datos en Vertica y un lago de datos en Amazon S3.

## Requisitos previos

- Implemente el conector en su Cuenta de AWS mediante la consola de Athena o AWS Serverless Application Repository. Para obtener más información, consulte [Implementación de un conector de origen de datos](#) o [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#).
- Antes de usar este conector, debe configurar una VPC y un grupo de seguridad. Para obtener más información, consulte [Creación de una VPC para un conector de origen de datos](#).

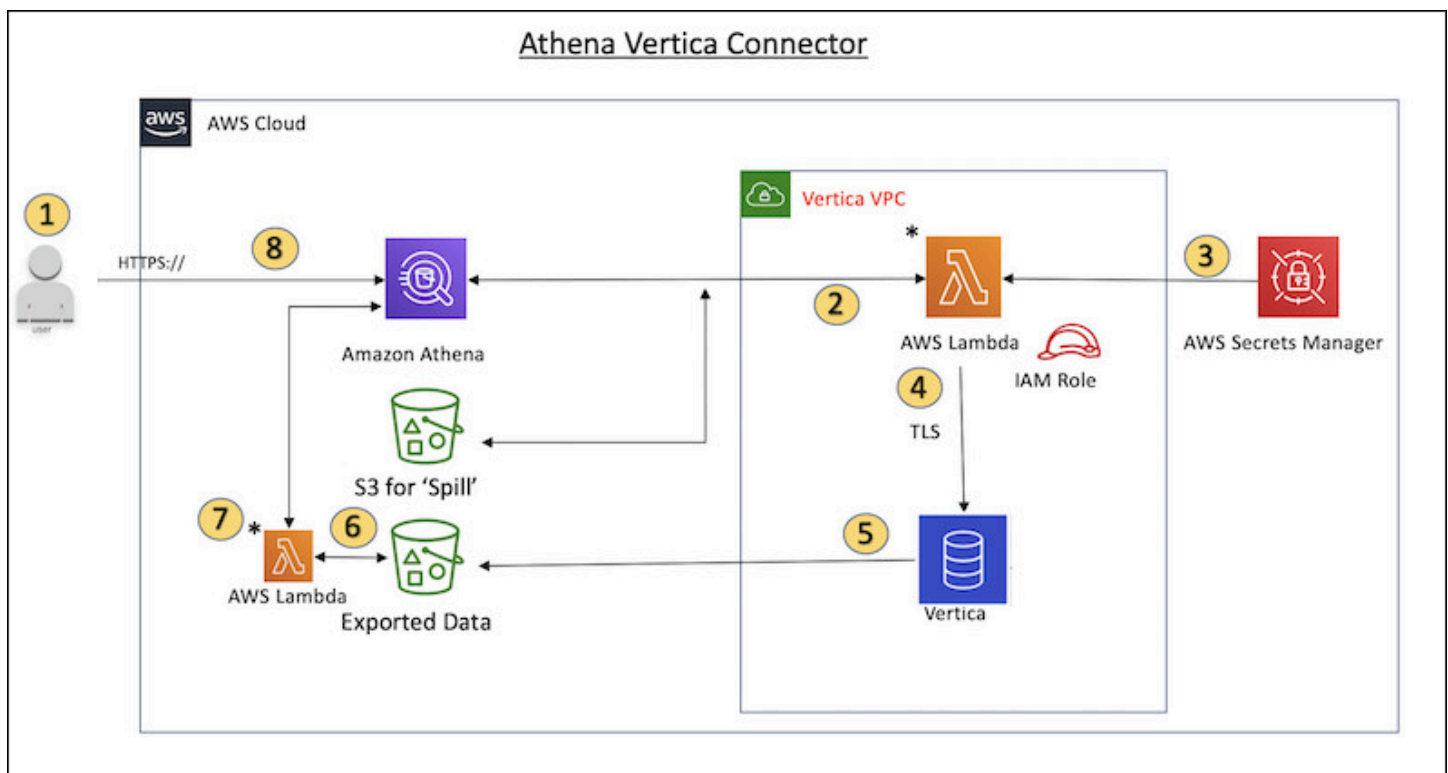
## Limitaciones

- Dado que el conector de Vertica de Athena usa [Amazon S3 Select](#) para leer archivos Parquet de Amazon S3, el rendimiento del conector puede ser lento. Cuando consulte tablas grandes, le recomendamos usar una consulta [CREAR TABLA COMO \(SELECCIONAR...\)](#) y predicados SQL.

- Actualmente, debido a un problema conocido en la consulta federada de Athena, el conector hace que Vertica exporte todas las columnas de la tabla consultada a Amazon S3, pero solo las columnas consultadas se pueden ver en los resultados de la consola de Athena.
- Las operaciones de escritura de DDL no son compatibles.
- Cualquier límite de Lambda relevante. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.

## Flujo de trabajo

En el siguiente diagrama, se muestra el flujo de trabajo de una consulta que usa el conector de Vertica.



1. Se emite una consulta SQL en una o más tablas de Vertica.
2. El conector analiza la consulta SQL para enviar la parte relevante a Vertica a través de la conexión JDBC.
3. Las cadenas de conexión usan el nombre de usuario y la contraseña almacenados en AWS Secrets Manager para acceder a Vertica.
4. El conector envuelve la consulta SQL con un comando EXPORT de Vertica, como se muestra en el siguiente ejemplo.

```
EXPORT TO PARQUET (directory = 's3://DOC-EXAMPLE-BUCKET/folder_name,
    Compression='Snappy', fileSizeMB=64) OVER() as
SELECT
PATH_ID,
...
SOURCE_ITEMIZED,
SOURCE_OVERRIDE
FROM DELETED_OBJECT_SCHEMA.FORM_USAGE_DATA
WHERE PATH_ID <= 5;
```

5. Vertica procesa la consulta SQL y envía el conjunto de resultados a un bucket de Amazon S3. Para obtener un mejor rendimiento, Vertica usa la opción EXPORT para paralelizar la operación de escritura de varios archivos Parquet.
6. Athena analiza el bucket de Amazon S3 para determinar el número de archivos que se van a leer para el conjunto de resultados.
7. Athena hace varias llamadas a la función de Lambda y usa [Amazon S3 Select](#) para leer los archivos Parquet del conjunto de resultados. Varias llamadas permiten a Athena paralelizar la lectura de los archivos de Amazon S3 y lograr un rendimiento de hasta 100 GB por segundo.
8. Athena procesa los datos que devuelve Vertica con los datos escaneados del lago de datos y devuelve el resultado.

## Términos

Los siguientes términos hacen referencia al conector de Vertica.

- Instancia de base de datos: cualquier instancia de una base de datos de Vertica implementada en Amazon EC2.
- Controlador: un controlador de Lambda que accede a la instancia de base de datos. Un controlador puede ser para los metadatos o para los registros de datos.
- Controlador de metadatos: un controlador de Lambda que recupera los metadatos de la instancia de base de datos.
- Controlador de registros: un controlador de Lambda que recupera registros de datos de la instancia de base de datos.
- Controlador compuesto: un controlador de Lambda que recupera tanto los metadatos como los registros de datos de la instancia de base de datos.

- Propiedad o parámetro: una propiedad de base de datos que usan los controladores para extraer información de la base de datos. Estas propiedades se configuran como variables de entorno de Lambda.
- Cadena de conexión: una cadena de texto que se usa para establecer una conexión con una instancia de base de datos.
- Catálogo: un catálogo ajeno a AWS Glue registrado en Athena que es un prefijo obligatorio para la propiedad `connection_string`.

## Parámetros

El conector de Vertica de Amazon Athena expone varias opciones de configuración a través de variables de entorno de Lambda. Puede usar las siguientes variables de entorno de Lambda para configurar el conector.

- `AthenaCatalogName`: nombre de la función de Lambda
- `ExportBucket`: el bucket de Amazon S3 en el que se exportan los resultados de consulta de Vertica.
- `SpillBucket`: nombre del bucket de Amazon S3 en el que esta función puede verter datos.
- `SpillPrefix`: el prefijo de la ubicación de `SpillBucket` en la que esta función puede verter datos.
- `SecurityGroupIds`: uno o más ID que corresponden al grupo de seguridad que se debe aplicar a la función de Lambda (por ejemplo, `sg1`, `sg2` o `sg3`).
- `SubnetIds`: uno o más ID de subred correspondientes a la subred que la función de Lambda puede usar para acceder al origen de datos (por ejemplo, `subnet1` o `subnet2`).
- `SecretNameOrPrefix`: el nombre o prefijo de un conjunto de nombres de Secrets Manager al que tiene acceso esta función (por ejemplo, `vertica-*`)
- `VerticaConnectionString`: los detalles de conexión de Vertica que se usarán de forma predeterminada si no se ha definido ninguna conexión específica del catálogo. Opcionalmente, la cadena puede usar la sintaxis de AWS Secrets Manager (por ejemplo, `${secret_name}`).
- ID de VPC: el ID de la VPC que se asocia a la función de Lambda.

## Cadena de conexión

Use una cadena de conexión JDBC con el siguiente formato para conectarse a una instancia de base de datos.

```
vertica://jdbc:vertica://host_name:port/database?user=vertica-username&password=vertica-password
```

## Uso de un controlador de conexión única

Puede usar los siguientes controladores de registros y metadatos de conexión única para conectarse a una sola instancia de Vertica.

Tipo de controlador	Clase
Controlador compuesto	VerticaCompositeHandler
Controlador de metadatos	VerticaMetadataHandler
Controlador de registros	VerticaRecordHandler

## Parámetros de controlador de conexión única

Parámetro	Descripción
default	Obligatorio. La cadena de conexión predeterminada.

Los controladores de conexión única admiten una instancia de base de datos y deben proporcionar un parámetro de cadena de conexión default. Se omiten todas las demás cadenas de conexión.

## Proporcionar credenciales

Para proporcionar un nombre de usuario y una contraseña para la base de datos en la cadena de conexión JDBC, puede usar las propiedades de la cadena de conexión o AWS Secrets Manager.

- Cadena de conexión: se pueden especificar un nombre de usuario y una contraseña como propiedades en la cadena de conexión JDBC.



**⚠ Important**

Como práctica recomendada en materia de seguridad, no utilice credenciales codificadas en las variables de entorno ni en las cadenas de conexión. Para obtener información sobre cómo transferir los secretos codificados a AWS Secrets Manager, consulte [Mover secretos codificados a AWS Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

- AWS Secrets Manager: para utilizar la característica Consulta federada de Athena con AWS Secrets Manager, la VPC conectada a la función de Lambda debe tener [acceso a Internet](#) o un [punto de conexión de VPC](#) para conectarse a Secrets Manager.

Puede poner el nombre de un secreto en AWS Secrets Manager, en la cadena de conexión JDBC. El conector reemplaza el nombre del secreto por los valores `username` y `password` de Secrets Manager.

Para las instancias de bases de datos de Amazon RDS, esta compatibilidad está estrechamente integrada. Si usa Amazon RDS, le recomendamos encarecidamente que use AWS Secrets Manager y la rotación de credenciales. Si la base de datos no usa Amazon RDS, guarde las credenciales como archivos JSON con el siguiente formato:

```
{"username": "${username}", "password": "${password}"}
```

### Ejemplo de cadena de conexión con nombres secretos

La siguiente cadena tiene los nombres secretos `${vertica-username}` y `${vertica-password}`.

```
vertica://jdbc:vertica://host_name:port/database?user=${vertica-username}&password=${vertica-password}
```

El conector usa el nombre secreto para recuperar los secretos y proporcionar el nombre de usuario y la contraseña, como en el siguiente ejemplo.

```
vertica://jdbc:vertica://host_name:port/database?user=sample-user&password=sample-password
```

Actualmente, el conector de Vertica reconoce las propiedades `vertica-username` y `vertica-password` de JDBC.

## Parámetros de vertido

El SDK de Lambda puede verter datos en Amazon S3. Todas las instancias de bases de datos a las que se accede mediante la misma función de Lambda se vierten en la misma ubicación.

Parámetro	Descripción
<code>spill_bucket</code>	Obligatorio. Nombre del bucket de vertido.
<code>spill_prefix</code>	Obligatorio. Prefijo de la clave del bucket de vertido.
<code>spill_put_request_headers</code>	(Opcional) Un mapa codificado en JSON de encabezados y valores de solicitudes para la solicitud <code>putObject</code> de Amazon S3 que se usa para el vertido (por ejemplo, <code>{"x-amz-server-side-encryption" : "AES256"}</code> ). Para ver otros encabezados posibles, consulte <a href="#">PutObject</a> en la referencia de la API de Amazon Simple Storage Service.

## Compatibilidad con tipos de datos

En la siguiente tabla, se muestran los tipos de datos admitidos para el conector de Vertica.

Booleano
BigInt
Short
Entero
Largo
Flotante
Doble
Date
Varchar

Booleano

Bytes

BigDecimal

TimeStamp como Varchar

## Rendimiento

La función de Lambda inserta proyecciones para reducir los datos analizados por la consulta. Las cláusulas LIMIT reducen la cantidad de datos analizados; sin embargo, si no proporciona un predicado, debe tener en cuenta que las consultas SELECT con una cláusula LIMIT analizan al menos 16 MB de datos. El conector para Vertica resiste las limitaciones debidas a la simultaneidad.

## Consultas de acceso directo

El conector para Vertica admite [consultas de acceso directo](#). Las consultas de acceso directo utilizan una función de tabla para enviar la consulta completa al origen de datos para su ejecución.

Para usar consultas de acceso directo con Vertica, puede utilizar la siguiente sintaxis:

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'query string'  
    ))
```

En el siguiente ejemplo de consulta, se envía una consulta a un origen de datos de Vertica. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.

```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10'  
    ))
```

## Información sobre licencias

Al usar este conector, reconoce la inclusión de componentes de terceros, cuya lista se puede encontrar en el archivo [pom.xml](#) para este conector y acepta los términos de las licencias de terceros correspondientes que se proporcionan en el archivo [LICENSE.txt](#) de GitHub.com.

## Recursos adicionales de

Para obtener la información más reciente sobre la versión del controlador JDBC, consulte el archivo [pom.xml](#) para el conector de Vertica en GitHub.com.

Para obtener más información acerca de este conector, consulte [el sitio correspondiente](#) en GitHub.com y [Consulta de un origen de datos de Vertica en Amazon Athena mediante el SDK de consulta federada de Athena](#) en el Blog de macrodatos de AWS.

## Implementación de un conector de origen de datos

La preparación para la creación de consultas federadas es un proceso de dos partes: implementación de un conector de origen de datos de función Lambda y conexión de la función Lambda a un origen de datos. En este proceso, proporciona a la función Lambda un nombre que luego puede elegir en la consola de Athena y asignar al conector un nombre al que puede referenciar en las consultas SQL.

### Note

Para utilizar la característica de consulta federada de Athena con AWS Secrets Manager, debe configurar un punto de conexión privado de Amazon VPC para Secrets Manager. Para obtener más información, consulte [Creación de un punto de conexión privado de VPC de Secrets Manager](#) en la Guía del usuario de AWS Secrets Manager.

## Temas

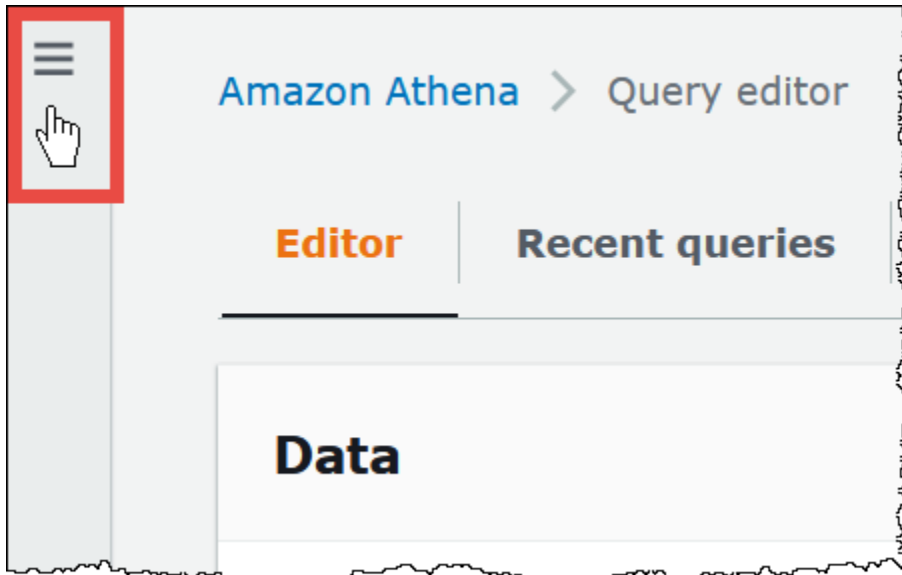
- [Uso de la consola de Athena](#)
- [Uso de AWS Serverless Application Repository para implementar un conector de origen de datos](#)
- [Creación de una VPC para un conector de origen de datos](#)
- [Habilitación de las consultas federadas entre cuentas](#)
- [Actualización de un conector de origen de datos](#)

## Uso de la consola de Athena

Para elegir, asignar un nombre e implementar un conector de origen de datos, utilice las consolas de Athena y Lambda en un proceso integrado.

Para implementar un conector de origen de datos, realice el siguiente procedimiento:

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.



3. En el panel de navegación, elija Data source (Orígenes de datos).
4. En la página Data sources (Orígenes de datos), elija Create data source (Crear origen de datos).
5. En Choose a data source (Elegir origen de datos), elija el origen de datos que quiera que Athena consulte, tenga en cuenta las siguientes recomendaciones:
  - Elija una opción de consulta federada que corresponda al origen de datos. Athena tiene conectores de orígenes de datos precreados que se pueden configurar para los orígenes, entre ellos, MySQL, Amazon DocumentDB y PostgreSQL.
  - Elija S3 -AWS Glue Data Catalog si quiere consultar datos en Amazon S3 y no utiliza un metastore de Apache Hive ni ninguna de las otras opciones de origen de datos de consultas federadas de esta página. Athena utiliza AWS Glue Data Catalog para almacenar metadatos e información de esquemas de orígenes de datos en Amazon S3. Esta es la opción predeterminada (no federada). Para obtener más información, consulte [Uso de AWS Glue para conectarse a orígenes de datos en Amazon S3](#).
  - Elija S3 - Apache Hive metastore (S3: metastore de Apache Hive) para consultar conjuntos de datos en Amazon S3 que utilicen un metastore de Apache Hive. Para obtener más información

acerca de esta opción, consulte [Conexión de Athena al almacén de metadatos de Apache Hive](#).

- Elija Custom or shared connector (Conector personalizado o compartido) si quiere crear su propio conector de origen de datos para usarlo con Athena. Para obtener información sobre cómo la escritura de un conector de origen de datos, consulte [Desarrollo de un conector de origen de datos mediante el SDK de Athena Query Federation](#).

En este tutorial se elige Amazon CloudWatch Logs como origen de datos federado.

6. Elija Siguiente.
7. En la página Enter data source details (Ingresar detalles del origen de datos), en Data source name (Nombre del origen de datos), ingrese el nombre que quiera utilizar en las instrucciones SQL cuando consulte el origen de datos desde Athena (por ejemplo, CloudWatchLogs). El nombre puede tener hasta 127 caracteres y debe ser único dentro de su cuenta. No se puede cambiar después de crearlo. Los caracteres válidos son a-z, A-Z, 0-9, \_ (guion bajo), @ (arroba) y - (guion). Los nombres awsdatacatalog, hive, jmx y system están reservados por Athena y no se pueden utilizar para nombres de orígenes de datos.
8. Para Lambda function (Función de Lambda), elija Create Lambda function (Crear una función de Lambda). La página de funciones del conector elegido se abre en la consola AWS Lambda. La página incluye información detallada sobre el conector.
9. En Application settings (Configuración de aplicación), lea detenidamente la descripción de cada configuración de aplicación y, a continuación, ingrese los valores que correspondan a los requisitos.

La configuración de aplicación que ve varía según el conector del origen de datos. La configuración mínima requerida incluye lo siguiente:

- AthenaCatalogName: un nombre para la función Lambda en minúsculas que indica el origen de datos de destino, como cloudwatchlogs.
- SpillBucket: un bucket de Amazon S3 en la cuenta para almacenar datos que superen los límites de tamaño de respuesta de la función de Lambda.

#### Note

Los datos vertidos no se reutilizan en ejecuciones posteriores y se pueden eliminar de forma segura después de 12 horas. Athena no elimina estos datos por usted. Para administrar estos objetos, considere agregar una política del ciclo de vida de

los objetos que elimine los datos antiguos del bucket de vertido de Simple Storage Service (Amazon S3). Para obtener más información, consulte [Administración del ciclo de vida de almacenamiento](#) en la Guía del usuario de Amazon S3.

10. Seleccione Confirmando que esta aplicación puede crear roles de IAM y políticas de recursos personalizados. Para obtener más información, elija el enlace Info.
11. Seleccione Implementar. Una vez finalizada la implementación, aparece la función Lambda en la sección Resources (Recursos) en la consola de Lambda.

## Conexión al origen de datos

Después de implementar el conector de origen de datos en su cuenta, puede conectar Athena a este.

Para conectar Athena a un origen de datos mediante un conector que haya implementado en su cuenta, lleve a cabo el siguiente procedimiento:

1. Vuelva a la página Enter data source details (Ingresar detalles de orígenes de datos) de la consola de Athena.
2. En la sección Connection details (Detalles de la conexión), elija el icono de actualización situado junto al cuadro de búsqueda Select or enter a Lambda function (Seleccione o introduzca una función de Lambda).
3. Elija el nombre de la función que acaba de crear en la consola de Lambda. Se muestra el ARN de la función Lambda.
4. (Opcional) En Tags (Etiquetas), agregue pares clave-valor que asociar con este origen de datos. Para obtener más información acerca de las etiquetas, consulte [Etiquetado de recursos de Athena](#).
5. Elija Siguiente.
6. En la página Review and create (Revisar y crear), revise los detalles del origen de datos y, a continuación, elija Create data source (Crear origen de datos).
7. La sección Data source details (Detalles del origen de datos) de la página de la fuente de datos muestra información sobre el nuevo conector. Ahora puede usar el conector en sus consultas de Athena.

Para obtener información sobre cómo utilizar conectores de datos en las consultas, consulte [Ejecución de consultas federadas](#).

## Uso de AWS Serverless Application Repository para implementar un conector de origen de datos

Para implementar un conector de origen de datos, puede utilizar [AWS Serverless Application Repository](#) en lugar de empezar con la consola de Athena. Utilice AWS Serverless Application Repository para encontrar el conector que quiere utilizar, proporcione los parámetros que requiere el conector y, a continuación, implemente el conector en su cuenta. Luego, después de implementar el conector, utilice la consola de Athena para poner el origen de datos a disposición de Athena.

### Implementación del conector en su cuenta

Para usar AWS Serverless Application Repository para implementar un conector de origen de datos en su cuenta, realice el siguiente procedimiento:

1. Inicie sesión en la AWS Management Console y abra el repositorio de aplicaciones sin servidor.
2. En el panel de navegación, elija Aplicaciones disponibles.
3. Seleccione la opción Show apps that create custom IAM roles or resource policies (Mostrar aplicaciones que crean roles de IAM personalizados o políticas de recursos).
4. En el cuadro de búsqueda, escriba el nombre del conector. Para obtener una lista de conectores de datos de Athena preconstruidos, consulte [Conectores de orígenes de datos disponibles](#).
5. Elija el nombre del conector. Al elegir un conector, se abre la página Application details (Detalles de la aplicación) de la función Lambda en la consola de AWS Lambda.
6. A la derecha de la página de detalles, en Application settings (Configuración de aplicación), complete la información requerida. La configuración mínima requerida incluye lo siguiente. Para obtener información sobre las opciones configurables restantes de los conectores de datos creados por Athena, consulte el tema [Available connectors](#) (Conectores disponibles) correspondiente en GitHub.
  - AthenaCatalogName: un nombre para la función Lambda en minúsculas que indica el origen de datos de destino, como `cloudwatchlogs`.
  - SpillBucket: especifique un bucket de Amazon S3 en la cuenta para recibir datos de cualquier carga de respuesta grande que supere los límites de tamaño de respuesta de la función Lambda.
7. Seleccione I acknowledge that this app creates custom IAM roles and resource policies (Confirmando que esta aplicación puede crear roles de IAM y políticas de recursos personalizados). Para obtener más información, elija el enlace Info.



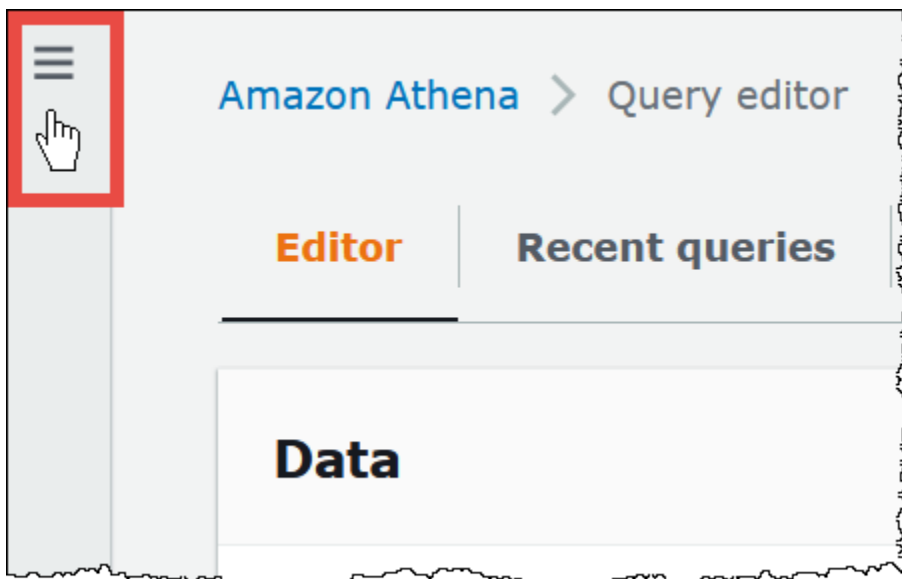
8. En la parte inferior derecha de la sección Configuración de aplicación, elija Implementar. Una vez finalizada la implementación, aparece la función Lambda en la sección Resources (Recursos) en la consola de Lambda.

Hacer que el conector esté disponible en Athena

Ahora está listo para usar la consola de Athena para poner el conector de origen de datos a disposición de Athena.

Para poner el origen de datos a disposición de Athena.

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.



3. En el panel de navegación, elija Data source (Orígenes de datos).
4. En la página Data sources (Orígenes de datos), elija Create data source (Crear origen de datos).
5. Para Choose a data source (Elegir un origen de datos), elija el origen de datos para el que ha creado un conector en el AWS Serverless Application Repository. Este tutorial usa Amazon CloudWatch Logs como origen de datos federado.
6. Elija Siguiente.
7. En la página Enter data source details (Ingresar detalles del origen de datos), en Data source name (Nombre del origen de datos), ingrese el nombre que quiera utilizar en las instrucciones SQL cuando consulte el origen de datos desde Athena (por ejemplo, CLOUDWATCHLOGS). El

- nombre puede tener hasta 127 caracteres y debe ser único dentro de su cuenta. No se puede cambiar después de crearlo. Los caracteres válidos son a-z, A-Z, 0-9, \_ (guion bajo), @ (arroba) y - (guion). Los nombres `awsdatacatalog`, `hive`, `jmx` y `system` están reservados por Athena y no se pueden utilizar para nombres de orígenes de datos.
- En la sección **Connection details** (Detalles de la conexión), utilice el cuadro **Select or enter a Lambda function** (Seleccione o introduzca una función de Lambda) para elegir el nombre de la función que acaba de crear. Se muestra el ARN de la función Lambda.
  - (Opcional) En **Tags** (Etiquetas), agregue pares clave-valor que asociar con este origen de datos. Para obtener más información acerca de las etiquetas, consulte [Etiquetado de recursos de Athena](#).
  - Elija **Siguiente**.
  - En la página **Review and create** (Revisar y crear), revise los detalles del origen de datos y, a continuación, elija **Create data source** (Crear origen de datos).
  - La sección **Data source details** (Detalles del origen de datos) de la página de la fuente de datos muestra información sobre el nuevo conector. Ahora puede usar el conector en sus consultas de Athena.

Para obtener información sobre cómo utilizar conectores de datos en las consultas, consulte [Ejecución de consultas federadas](#).

## Creación de una VPC para un conector de origen de datos

Algunos conectores de origen de datos de Athena deben disponer de una VPC y un grupo de seguridad. Este tema muestra cómo crear una VPC con una subred y un grupo de seguridad para la VPC. En este proceso, se recuperan los ID de la VPC, la subred y el grupo de seguridad que usted haya creado. Estos ID son necesarios cuando configura su conector para utilizarlo con Athena.

Para crear una VPC para un conector de origen de datos de Athena

- Inicie sesión en la AWS Management Console y abra la consola de Amazon VPC en <https://console.aws.amazon.com/vpc/>.
- En el panel de navegación, asegúrese de que se haya seleccionado la opción **New VPC Experience** (Nueva experiencia de VPC).
- Seleccione **Crear VPC**.
- En **VPC Settings** (Configuración de la VPC), en **Resources to create** (Recursos para crear), elija **VPC, etc.** (VPC, etc.).

5. En Auto-generate (Generación automática), ingrese un valor que se utilizará con el fin de generar etiquetas de nombre para todos los recursos de su VPC.
6. Seleccione Crear VPC.
7. Elija View VPC (Ver VPC).
8. En la sección Details (Detalles), en VPC ID (ID de la VPC), copie el ID de la VPC para su posterior referencia.

A continuación, ya podrá recuperar el ID de subred de la VPC que acaba de crear.

Para recuperar el ID de subred de la VPC

1. En el panel de navegación de la consola de la VPC, elija Subnets (Subredes).
2. Elija el nombre correspondiente a la subred que creó.
3. En la sección Details (Detalles), en Subnet ID (ID de la subred), copie el ID de la subred para su posterior referencia.

A continuación, cree un grupo de seguridad para la VPC.

Para crear un grupo de seguridad para la VPC

1. En el panel de navegación de la consola de la VPC, elija Security (Seguridad), Security Groups (Grupos de seguridad).
2. Elija Create Security Group (Crear grupo de seguridad).
3. En la página Create Security Group (Crear grupo de seguridad), ingrese la información que se indica a continuación:
  - En Security group name (Nombre del grupo de seguridad), ingrese un nombre para el grupo de seguridad.
  - En Description (Descripción), ingrese una descripción para el grupo de seguridad. Este campo es obligatorio.
  - En VPC:, ingrese el ID de la VPC de aquella que creó para el conector de origen de datos.
  - En Inbound rules (Reglas de entrada) y Outbound rules (Reglas de salida), agregue las reglas entrantes y salientes que necesite.
4. Elija Create Security Group (Crear grupo de seguridad).

5. En la página Details (Detalles) del grupo de seguridad, copie el ID de grupo de seguridad para su posterior referencia.

## Habilitación de las consultas federadas entre cuentas

La consulta federada permite consultar orígenes de datos distintos de Amazon S3 mediante conectores de origen de datos implementados en AWS Lambda. La característica de consulta federada entre cuentas permite que la función Lambda y los orígenes de datos que se van a consultar se encuentren en cuentas diferentes.

Como administrador de datos, puede habilitar las consultas federadas entre cuentas al compartir el conector de datos con la cuenta de un analista de datos o, como analista de datos, al utilizar un ARN Lambda compartido de un administrador de datos para agregarlo a su cuenta. Cuando se hacen cambios de configuración en un conector de la cuenta de origen, la configuración actualizada se aplica automáticamente a las instancias compartidas del conector en las cuentas de otro usuario.

## Consideraciones y limitaciones

- La característica de consulta federada entre cuentas está disponible para conectores de datos de metastore que no son de Hive y utilizan un origen de datos basado en Lambda.
- La característica no está disponible para el tipo de origen de datos AWS Glue Data Catalog. Para obtener información sobre el acceso entre cuentas a AWS Glue Data Catalog, consulte [Acceso entre cuentas a los catálogos de datos de AWS Glue](#).
- Si la respuesta de la función de Lambda del conector supera el límite de tamaño de respuesta de Lambda de 6 MB, Athena cifra, agrupa y distribuye de forma automática la respuesta en un bucket de Amazon S3 que configure. La entidad que ejecuta la consulta de Athena debe tener acceso a la ubicación de la distribución para que Athena pueda leer los datos distribuidos. Le recomendamos que establezca una política de ciclo de vida de Amazon S3 para eliminar los objetos de la ubicación de la distribución, ya que los datos no son necesarios una vez que finaliza la consulta.
- No se admite el uso de consultas federadas entre Regiones de AWS.

## Permisos necesarios

- Para que la cuenta A del administrador de datos comparta una función Lambda con la cuenta B de analista de datos, la cuenta B requiere la función de invocación de Lambda y el acceso al bucket de desbordamiento. En consecuencia, la cuenta A debería agregar una [política basada en](#)

[recursos](#) a la función Lambda y el acceso de la [entidad principal](#) al bucket de desbordamiento en Amazon S3.

1. La siguiente política otorga permisos de función de invocación de Lambda a la cuenta B en una función Lambda en la cuenta A.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CrossAccountInvocationStatement",
      "Effect": "Allow",
      "Principal": {
        "AWS": ["arn:aws:iam::account-B-id:user/username"]
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:aws-region:account-A-id:function:lambda-function-name"
    }
  ]
}
```

2. La siguiente política permite el acceso del bucket de desbordamiento a la entidad principal de la cuenta B.

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": ["arn:aws:iam::account-B-id:user/username"]
      },
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3::spill-bucket",
        "arn:aws:s3::spill-bucket/*"
      ]
    }
  ]
}
```

```
}

```

- Si la función Lambda cifra el bucket de desbordamiento con una clave AWS KMS en lugar del cifrado predeterminado que ofrece el SDK de federación, la política de la clave AWS KMS de la cuenta A debe conceder acceso al usuario de la cuenta B, como en el ejemplo siguiente.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal":
  {
    "AWS": ["arn:aws:iam::account-B-id:user/username"]
  },
  "Action": [ "kms:Decrypt" ],
  "Resource": "*" // Resource policy that gets placed on the KMS key.
}
```

- Para que la cuenta A comparta el conector con la cuenta B, la cuenta B debe crear un rol denominado AthenaCrossAccountCreate-*account-A-id* que la cuenta A asume al llamar a la acción de la API [AssumeRole](#) de AWS Security Token Service.

La siguiente política, que permite la acción CreateDataCatalog, debe crearse en la cuenta B y agregarse al rol AthenaCrossAccountCreate-*account-A-id* que crea la cuenta B para la cuenta A.

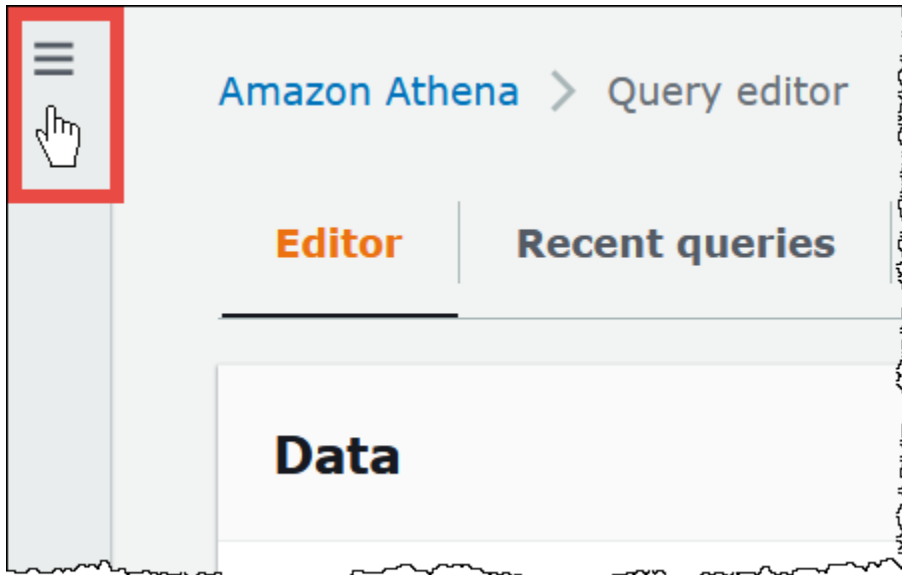
```
{
  "Effect": "Allow",
  "Action": "athena:CreateDataCatalog",
  "Resource": "arn:aws:athena:*:account-B-id:datacatalog/*"
}
```

## Uso compartido de un origen de datos de la cuenta A con la cuenta B

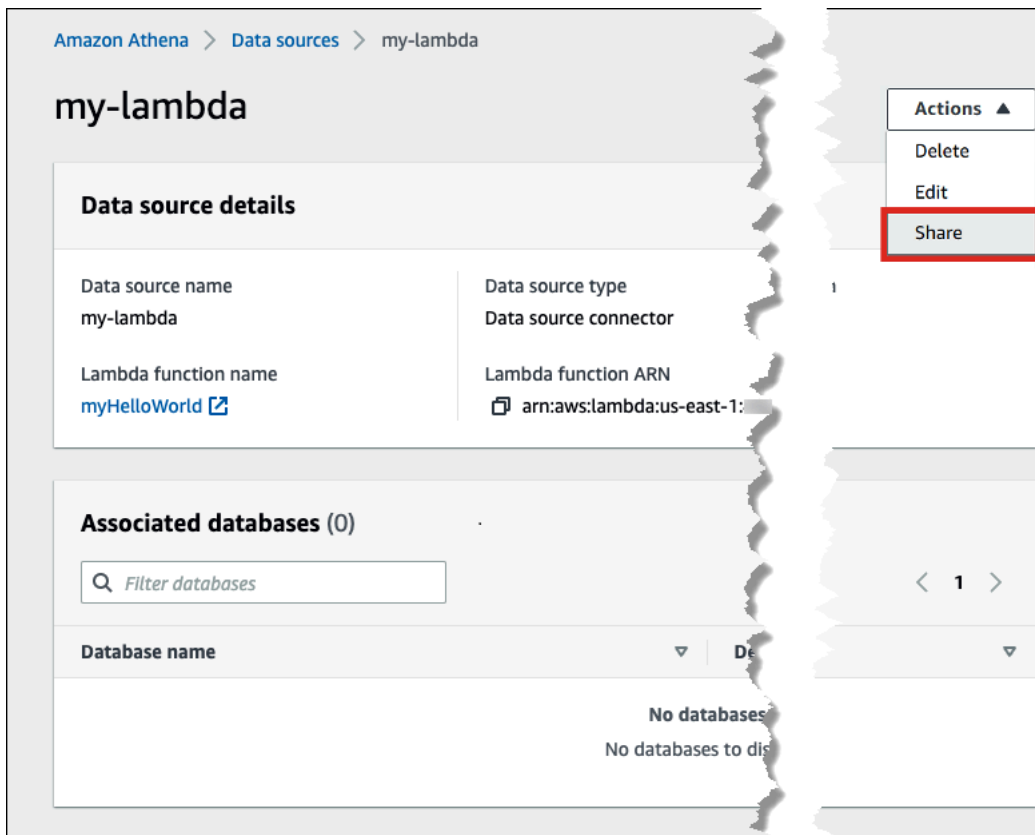
Una vez establecidos los permisos, puede utilizar la página Data sources (Orígenes de datos) en la consola de Athena para compartir un conector de datos de su cuenta (cuenta A) con otra cuenta (cuenta B). La cuenta A mantiene el control y la propiedad del conector. Cuando la cuenta A hace cambios de configuración en el conector, la configuración actualizada se aplica al conector compartido de la cuenta B.

Para compartir un origen de datos de Lambda de la cuenta A con la cuenta B

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.



3. Elija Edit Data Source (Editar origen de datos).
4. En la página Data sources (Orígenes de datos), elija el enlace del conector que quiera compartir.
5. En la página de detalles de un origen de datos de Lambda, elija la opción Share (Compartir) en la esquina superior derecha.



6. En el cuadro de diálogo Share **Lambda-name** with another account (Compartir nombre-Lambda con otra cuenta, ingrese la información requerida.
  - En Data source name (Nombre de origen de datos), ingrese el nombre del origen de datos copiado tal como quiera que aparezca en la otra cuenta.
  - En Account ID ID de cuenta, ingrese el ID de la cuenta con la que quiera compartir su origen de datos (en este caso, la cuenta B).



## Share my-lambda with another account? [Learn more](#)

**Data source name**  
Create a unique name to specify this data source within a SQL statement. For example, `SELECT * from <catalogName>.<database>.<table>`

The name cannot be changed after creation. It can be up to 127 characters. Valid characters are a-z, A-Z, 0-9, \_(underscore), @(at sign) and -(hyphen).

**Account ID**

Account ID can only be numbers (0-9) and 12 characters.

**Cancel** **Share**

7. Elija Compartir. El conector de datos compartidos que ha especificado se crea en la cuenta B. Los cambios de configuración del conector de la cuenta A se aplican al conector de la cuenta B.

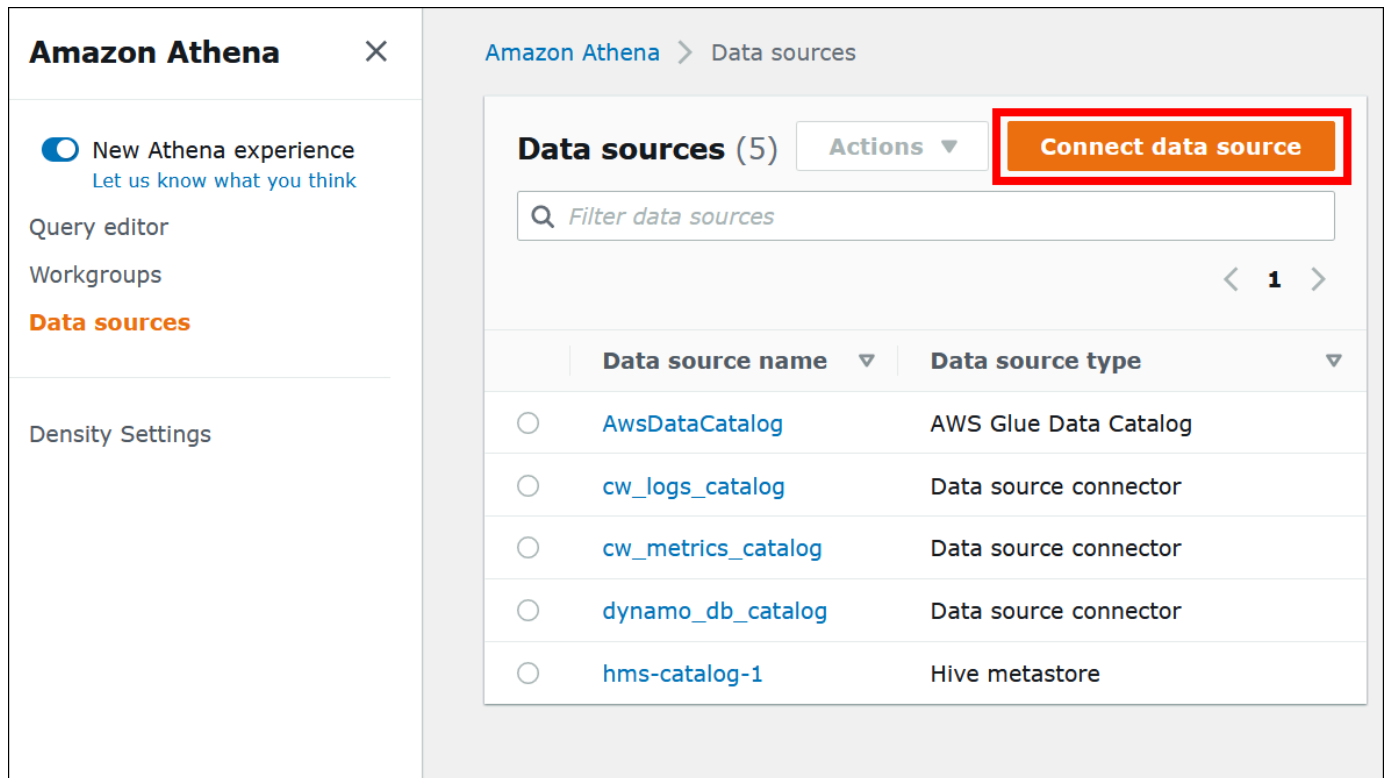
### Incorporación de un origen de datos compartido de la cuenta A a la cuenta B

Como analista de datos, es posible que se le proporcione el ARN de un conector para agregarlo a la cuenta desde un administrador de datos. Puede utilizar la página Data sources (Orígenes de datos) de la consola de Athena para agregar el ARN de Lambda proporcionado por el administrador de la cuenta.

Para agregar el ARN de Lambda de un conector de datos compartido a la cuenta

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si utiliza la nueva experiencia de la consola y el panel de navegación no está visible, elija el menú de expansión de la izquierda.
3. Elija Edit Data Source (Editar origen de datos).

- En la página Data sources (Orígenes de datos), elija Connect data source (Conectar origen de datos).



The screenshot shows the Amazon Athena console interface. On the left is a navigation sidebar with options like 'New Athena experience', 'Query editor', 'Workgroups', 'Data sources' (highlighted), and 'Density Settings'. The main content area is titled 'Data sources (5)' and includes a search bar, a pagination control showing page 1, and a table of existing data sources. A red rectangular box highlights the 'Connect data source' button in the top right corner of the main content area.


	Data source name ▾	Data source type ▾
<input type="radio"/>	<a href="#">AwsDataCatalog</a>	AWS Glue Data Catalog
<input type="radio"/>	<a href="#">cw_logs_catalog</a>	Data source connector
<input type="radio"/>	<a href="#">cw_metrics_catalog</a>	Data source connector
<input type="radio"/>	<a href="#">dynamo_db_catalog</a>	Data source connector
<input type="radio"/>	<a href="#">hms-catalog-1</a>	Hive metastore


- Elija Custom or shared connector (Conector personalizado o compartido).


Amazon Athena > Data sources > Connect data sources


## Connect data sources

**Data source selection** [Info](#)  
Choose the data source to query with Athena


 **S3 - AWS Glue Data Catalog**  
Queries data from S3.


 **S3 - Apache Hive metastore**  
Queries data from S3.

 **Redis**  
Queries data from Redis.

 **Custom or shared connector**  
Use a custom or another account's connector.

6. En la sección Lambda function (Función Lambda), asegúrese de que esté seleccionada la opción Use an existing Lambda function (Utilizar una función Lambda existente).

 **Redis**  
Queries data from Redis.

 **Custom or shared connector**  
Use a custom or another account's connector.

### Data source details

#### Lambda function [Info](#)

Choose or enter a Lambda function for your data source, or create and configure a Lambda function for the connection.

Choose an existing Lambda function or create a new one  
Select whether you want to access an existing Lambda function or create a new Lambda function to connect to the data source.

Use an existing Lambda function

Create a new Lambda function

Choose or enter a Lambda function  
Choose a Lambda function to connect to your data source, or enter the ARN for a cross-account Lambda data source function. To manage the Lambda function details, use the Lambda console. [Info](#)

7. En Choose or enter a Lambda function (Elegir o ingresar una función Lambda), ingrese el ARN de Lambda de la cuenta A.
8. Elija Connect data source (Conectar origen de datos).

### Solución de problemas

Si recibe un mensaje de error que indica que la cuenta A no tiene permisos para asumir un rol en la cuenta B, asegúrese de que el nombre del rol creado en la cuenta B esté escrito correctamente y de que tenga adjunta la política adecuada.

## Actualización de un conector de origen de datos

Athena recomienda actualizar periódicamente los conectores de orígenes de datos que utilice a la última versión para aprovechar las nuevas características y mejoras. Para empezar, debe buscar el número de la última versión.

### Búsqueda de la última versión de Athena Query Federation

El número de versión más reciente de los conectores de orígenes de datos de Athena corresponde a la última versión de Athena Query Federation. En ciertos casos, las versiones de GitHub pueden ser ligeramente más recientes que las que están disponibles en el AWS Serverless Application Repository (SAR).

### Para buscar la última versión de Athena Query Federation

1. Visite la URL de GitHub <https://github.com/awslabs/aws-athena-query-federation/releases/latest>.
2. Anote el número de la versión en el encabezado de la página principal en el siguiente formato:

Versión v *year.week\_of\_year.iteration\_of\_week* de Athena Query Federation

Por ejemplo, el número de versión de la Versión v2023.8.3 de Athena Query Federation es 2023.8.3.

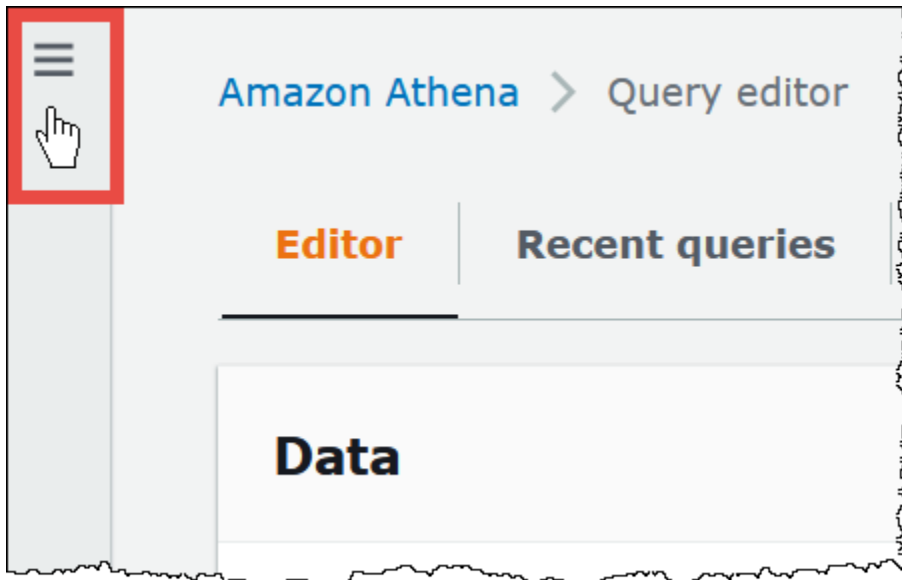
### Búsqueda y anotación de los nombres de los recursos

Como preparación para la actualización, debe buscar y anotar la siguiente información:



1. El nombre de la función de Lambda del conector.
2. Las variables de entorno para la función de Lambda.
3. El nombre de la aplicación de Lambda, que administra la función de Lambda del conector.

### Para buscar nombres de recursos en la consola de Athena

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.



3. En el panel de navegación, elija Data source (Orígenes de datos).
4. En la columna Nombre del origen de datos, elija el enlace al origen de datos del conector.
5. En la sección Detalles del origen de datos, en Función de Lambda, elija el enlace a su función de Lambda.

Data source details		
Data source name dynamo_db_catalog	Data source type Data source connector	Description DynamoDB Catalog
Lambda function <a href="#">dynamo_db_lambda</a> 	Lambda function ARN  arn:aws:lambda:us-west-2: [redacted] :function:dynamo_db_lambda	

6. En la página Funciones, en la columna Nombre de la función, anote el nombre de la función del conector.

The screenshot shows the AWS Lambda console interface. At the top, it says 'Lambda > Functions'. Below that, there's a section for 'Functions (5)' with a refresh button and a 'Create function' button. A search bar contains the text 'Filter by tags and attributes or search by keyword' and shows 'Matches: 1'. A filter tag 'dynamo' is active, and there's a 'Clear filters' button. Below the search bar is a table with columns: 'Function name', 'Description', 'Package type', 'Runtime', and 'Last modified'. The table contains one entry: 'dynamodbdatasource' (highlighted with a red box), 'Enables Amazon Athena to communicate with DynamoDB, making your tables accessible via SQL', 'Zip', 'Java 11 (Corretto)', and '1 hour ago'.

7. Seleccione el enlace del nombre de la función.
8. En la sección Información general de la función, seleccione la pestaña Configuración.
9. En el panel de la izquierda, elija Variables de entorno.
10. En la sección Variables de entorno, anote las claves y sus valores correspondientes.
11. Desplácese hasta la parte superior de la página.
12. En el mensaje Esta función pertenece a una aplicación. Haga clic aquí para administrarla, seleccione el enlace Haga clic aquí.
13. En la página serverlessrepo-*your\_application\_name*, anote el nombre de su aplicación sin serverlessrepo. Por ejemplo, si el nombre de la aplicación es serverlessrepo-DynamoDbTestApp, el nombre de su aplicación es DynamoDbTestApp.
14. Permanezca en la página de la consola de Lambda de su aplicación y, a continuación, siga con los pasos de Búsqueda de la versión del conector que está utilizando.

Búsqueda de la versión del conector que está utilizando

Siga estos pasos para encontrar la versión del conector que está utilizando.

Para buscar la versión del conector que está utilizando

1. En la página de la consola de Lambda correspondiente a la aplicación de Lambda, seleccione la pestaña Implementaciones.
2. En la pestaña Implementaciones, expanda la Plantilla de SAM.

3. Busque CodeUri.
4. En el campo Clave de CodeUri, busque la siguiente cadena:

```
applications-connector_name-  
versions-year.week_of_year.iteration_of_week/hash_number
```

En el siguiente ejemplo, se muestra una cadena para el conector de CloudWatch:

```
applications-AthenaCloudwatchConnector-versions-2021.42.1/15151159...
```

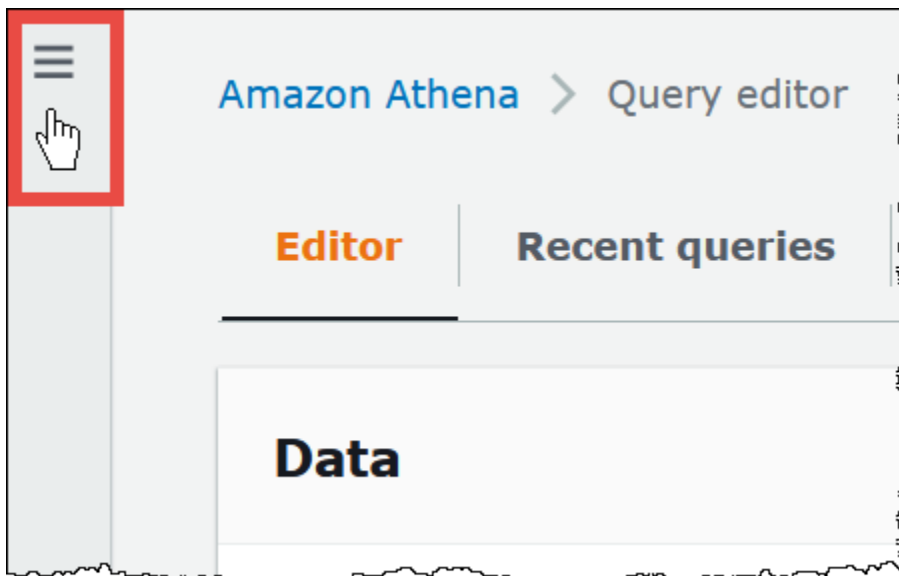
5. Registre el valor *year.week\_of\_year.iteration\_of\_week* (por ejemplo, 2021.42.1). Esta es la versión de su conector.

### Implementación de la nueva versión del conector

Siga estos pasos para implementar una nueva versión del conector.

Para implementar una nueva versión del conector

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.



3. En el panel de navegación, elija Data source (Orígenes de datos).
4. En la página Data sources (Orígenes de datos), elija Create data source (Crearo origen de datos).



5. Seleccione el origen de datos que desee actualizar y, a continuación, seleccione **Siguiente**.
6. En la sección **Detalles de la conexión**, elija **Crear función de Lambda**. Con esto, se abre la consola de Lambda, donde podrá implementar la aplicación actualizada.

Lambda > Applications > Review, configure and deploy

## AthenaDynamoDBConnector — version 2023.6.1

[Copy as SAM Resource](#)

Review, configure and deploy

### Application details

Author	Source code URL	Description	Report a vulnerability
<a href="#">Amazon Athena Federation</a> AWS verified author	<a href="https://github.com/aws-labs/aws-athena-query-federation">https://github.com/aws-labs/aws-athena-query-federation</a>	This connector enables Amazon Athena to communicate with DynamoDB, making your tables accessible via SQL.	If you believe this application poses a security risk, please file a vulnerability report.

▶ **Template**

▶ **Permissions**

▶ **License**

### Readme file

View on the [AWS Serverless Application Repository site](#).

### Application settings

**Application name**  
The stack name of this application created via AWS CloudFormation

7. Como en realidad no va a crear un nuevo origen de datos, puede cerrar la pestaña de la consola de Athena.
8. En la página de la consola de Lambda correspondiente al conector, proceda con los siguientes pasos:
  - a. Asegúrese de haber eliminado el prefijo `serverlessrepo-` del nombre de la aplicación y, a continuación, copie el nombre de la aplicación en el campo **Nombre de la aplicación**.

- b. Copie el nombre de la función de Lambda en el campo AthenaCatalogName. Algunos conectores llaman a este campo LambdaFunctionName.
  - c. Copie las variables de entorno que ha registrado en los campos correspondientes.
9. Seleccione la opción Confirmo que esta aplicación puede crear políticas de recursos y roles de IAM personalizados y, a continuación, elija Implementar.
  10. Para comprobar que la aplicación se ha actualizado, seleccione la pestaña Implementaciones.

En la sección Historial de implementaciones, se muestra que la actualización se ha completado.

The screenshot shows the AWS Lambda console for the application 'serverlessrepo-AthenaDynamoDBConnector'. The 'Deployments' tab is active. Below the 'SAM template' section, there is a 'Deployment history' table. The table has four columns: 'Deployment', 'Resource type', 'Last updated time', and 'Status'. The first row is highlighted with a red border and shows a deployment that occurred '2 minutes ago', is a 'Lambda application', and has a status of 'Update complete' with a green checkmark icon. Below it, there are two more rows for deployments from 'last year' and '3 years ago', both also showing 'Update complete' status.

Deployment	Resource type	Last updated time	Status
2 minutes ago	Lambda application	2 minutes ago	Update complete
last year	Lambda application	last year	Update complete
3 years ago	Lambda application	3 years ago	Update complete

11. Para confirmar el nuevo número de versión, puede expandir la Plantilla de SAM como antes, buscar CodeUri y comprobar el número de versión del conector en el campo Clave.

Ahora puede usar el conector actualizado para crear consultas federadas de Athena.

## Ejecución de consultas federadas

Una vez que haya configurado uno o más conectores de datos y los haya implementado en su cuenta, podrá usarlos en sus consultas de Athena.

### Consulta de un único origen de datos

En los ejemplos de esta sección se presupone que ha configurado e implementado los [Conector CloudWatch de Amazon Athena](#) en su cuenta. Utilice el mismo enfoque para realizar consultas cuando utilice otros conectores.

Para crear una consulta de Athena que utilice el conector CloudWatch

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. En el editor de consultas de Athena, cree una consulta SQL que utilice la siguiente sintaxis en la cláusula FROM.

```
MyCloudwatchCatalog.database_name.table_name
```

## Ejemplos

En el siguiente ejemplo, se utiliza el conector Athena CloudWatch para conectarse a la vista `all_log_streams` en el [Grupo de registros](#) de CloudWatch Logs `/var/ecommerce-engine/order-processor`. La vista `all_log_streams` es una vista de todas las secuencias de registro del grupo de registros. La consulta de ejemplo limita el número de filas devueltas a 100.

```
SELECT *
FROM "MyCloudwatchCatalog"."/var/ecommerce-engine/order-processor".all_log_streams
LIMIT 100;
```

En el siguiente ejemplo se analiza la información de la misma vista que el ejemplo anterior. En el ejemplo se extrae el ID de pedido y el nivel de registro y se filtra cualquier mensaje que tenga el nivel INFO.

```
SELECT
  log_stream as ec2_instance,
  Regexp_extract(message '.*orderId=(\d+) .*', 1) AS orderId,
  message AS order_processor_log,
  Regexp_extract(message, '(.):.*', 1) AS log_level
FROM MyCloudwatchCatalog."/var/ecommerce-engine/order-processor".all_log_streams
WHERE Regexp_extract(message, '(.):.*', 1) != 'INFO'
```

## Consulta de varios orígenes de datos

Como ejemplo más complejo, imagine una empresa de comercio electrónico que utiliza los siguientes orígenes de datos para almacenar datos relacionados con las compras de los clientes:

- [Amazon RDS para MySQL](#) para almacenar los datos del catálogo de productos.
- [Amazon DocumentDB](#) para almacenar datos de las cuenta de los cliente, como direcciones de correo electrónico y direcciones de envío.

- [Amazon DynamoDB](#) para almacenar datos de seguimiento y envío de pedidos.

Imagine que un analista de datos de esta aplicación de comercio electrónico descubre que el tiempo de envío en algunas regiones se ha visto afectado por las condiciones meteorológicas locales. El analista quiere saber cuántos pedidos están atrasados, dónde se encuentran los clientes afectados y qué productos están más comprometidos. En lugar de investigar las fuentes de información por separado, el analista utiliza Athena para unir los datos en una única consulta federada.

### Example

```
SELECT
    t2.product_name AS product,
    t2.product_category AS category,
    t3.customer_region AS region,
    count(t1.order_id) AS impacted_orders
FROM my_dynamodb.default.orders t1
JOIN my_mysql.products.catalog t2 ON t1.product_id = t2.product_id
JOIN my_documentdb.default.customers t3 ON t1.customer_id = t3.customer_id
WHERE
    t1.order_status = 'PENDING'
    AND t1.order_date between '2022-01-01' AND '2022-01-05'
GROUP BY 1, 2, 3
ORDER BY 4 DESC
```

### Consulta de vistas federadas

Al consultar orígenes federados, puede utilizar las vistas para ocultar los orígenes de datos subyacentes u ocultar las combinaciones complejas a otros analistas que consulten los datos.

### Consideraciones y limitaciones

- Las vistas federadas requieren la versión 3 del motor de Athena.
- Las vistas federadas se almacenan en el AWS Glue, no con el origen de datos subyacente.
- Las vistas creadas con catálogos federados deben utilizar una sintaxis de nombres calificada completa, como en el siguiente ejemplo:

```
"ddbcatalog"."default"."customers"
```

- Los usuarios que realicen consultas en orígenes federados deben tener permiso para consultar los orígenes federados.

- El permiso `athena:GetDataCatalog` es obligatorio para las vistas federadas. Para obtener más información, consulte [Ejemplos de políticas de permisos de IAM para permitir la consulta federada de Athena](#).

## Ejemplos

En el siguiente ejemplo, se crea una vista denominada `customers` en los datos almacenados en un origen de datos federado.

### Example

```
CREATE VIEW customers AS
SELECT *
FROM my_federated_source.default.table
```

En el siguiente ejemplo de consulta, se muestra una consulta que hace referencia a la vista `customers` en lugar de al origen de datos federado subyacente.

### Example

```
SELECT id, SUM(order_amount)
FROM customers
GROUP by 1
ORDER by 2 DESC
LIMIT 50
```

En el siguiente ejemplo, se crea una vista denominada `order_summary` que combina datos de un origen de datos federado y de un origen de datos de Amazon S3. Desde el origen federado, que ya se creó en Athena, la vista usa las tablas `person` y `profile`. Desde Amazon S3, la vista usa las tablas `purchase` y `payment`. Para hacer referencia a Amazon S3, la instrucción utiliza la palabra clave `awsdatacatalog`. Tenga en cuenta que el origen de datos federado utiliza la sintaxis de nombres calificada completa *`federated_source_name.federated_source_database.federated_source_table`*.

### Example

```
CREATE VIEW default.order_summary AS
SELECT *
FROM federated_source_name.federated_source_database."person" p
JOIN federated_source_name.federated_source_database."profile" pr ON pr.id = p.id
```

```
JOIN awsdatacatalog.default.purchase i ON p.id = i.id
JOIN awsdatacatalog.default.payment pay ON pay.id = p.id
```

## Recursos adicionales de

- Para ver un ejemplo de una vista federada que está desacoplada de su fuente de origen y que se encuentra disponible para su análisis bajo demanda en un modelo multiusuario, consulte [Amplíe la malla de datos con Amazon Athena y las vistas federadas](#) en el Blog sobre macrodatos de AWS.
- Para obtener más información sobre cómo trabajar con vistas en Athena, consulte [Uso de vistas](#).

## Ejecución de consultas de acceso directo federadas

En Athena, puede ejecutar consultas en orígenes de datos federados utilizando el lenguaje de consulta del propio origen de datos y enviar la consulta completa al origen de datos para su ejecución. Estas consultas se denominan consultas de acceso directo. Para ejecutar consultas de acceso directo, utilice una función de tabla en la consulta de Athena. Incluya la consulta de acceso directo que se ejecutará en el origen de datos en uno de los argumentos de la función de tabla. Las consultas de acceso directo devuelven una tabla que puede analizar con Athena SQL.

## Conectores admitidos

Los siguientes conectores de orígenes de datos de Athena admiten consultas de acceso directo.

- [Azure Data Lake Storage](#)
- [Azure Synapse](#)
- [Cloudera Hive](#)
- [Cloudera Impala](#)
- [CloudWatch](#)
- [Db2](#)
- [Db2 iSeries](#)
- [DocumentDB](#)
- [DynamoDB](#)
- [HBase](#)
- [Google BigQuery](#)
- [Hortonworks](#)
- [MySQL](#)

- [Neptune](#)
- [OpenSearch](#)
- [Oracle](#)
- [PostgreSQL](#)
- [Redshift](#)
- [SAP HANA](#)
- [Snowflake](#)
- [SQL Server](#)
- [Teradata](#)
- [Timestream](#)
- [Vertica](#)

## Consideraciones y limitaciones

Cuando utilice las consultas de acceso directo en Athena, tenga en cuenta los siguientes puntos:

- El acceso directo de consultas solo se admite para las instrucciones de SELECT u operaciones de lectura de Athena.
- Las consultas de acceso directo deben ejecutarse en el contexto del catálogo de la consulta externa (es decir, la consulta que llama a la función de tabla).
- El rendimiento de las consultas puede variar en función de la configuración del origen de datos.
- No se admiten las consultas de acceso directo en las vistas.

## Sintaxis

La sintaxis general del acceso directo de las consultas de Athena es la siguiente.

```
SELECT * FROM TABLE(system.function_name(arg1 => 'arg1Value'[, arg2 => 'arg2Value', ...]))
```

En la mayoría de los orígenes de datos, el primer y único argumento es query seguido del operador de flecha => y la cadena de consulta.

```
SELECT * FROM TABLE(system.query(query => 'query string'))
```

Para simplificar, puede omitir el argumento con nombre opcional `query` y el operador de flecha `=>`.

```
SELECT * FROM TABLE(system.query('query string'))
```

Si el origen de datos requiere más que la cadena de consulta, utilice los argumentos con nombre en el orden esperado por el origen de datos. Por ejemplo, la expresión `arg1 => 'arg1Value'` contiene el primer argumento y su valor. El nombre `arg1` es específico del origen de datos y puede variar de un conector a otro.

```
SELECT * FROM TABLE(  
    system.query(  
        arg1 => 'arg1Value',  
        arg2 => 'arg2Value',  
        arg3 => 'arg3Value'  
    ));
```

Para obtener información sobre la sintaxis exacta que se debe utilizar con un conector concreto, consulte la página del conector individual.

## Uso de comillas

Los valores de los argumentos, incluida la cadena de consulta que pase, deben escribirse entre comillas simples, como en el siguiente ejemplo.

```
SELECT * FROM TABLE(system.query(query => 'SELECT * FROM testdb.persons LIMIT 10'))
```

Cuando la cadena de consulta está entre comillas dobles, la consulta falla. Se produce un error en la siguiente consulta y aparece el mensaje de error `COLUMNA_NO_ENCONTRADA`: línea 1:43: Columna 'seleccionar \* de testdb.persons límite 10' no se puede resolver.

```
SELECT * FROM TABLE(system.query(query => "SELECT * FROM testdb.persons LIMIT 10"))
```

Para evitar una comilla simple, añada una comilla simple al original (por ejemplo, `terry 's_group` a `terry' 's_group`).

## Ejemplos

El siguiente ejemplo de consulta envía una consulta a un origen de datos. La consulta selecciona todas las columnas de la tabla `customer` y limita los resultados a 10.



```
SELECT * FROM TABLE(  
    system.query(  
        query => 'SELECT * FROM customer LIMIT 10;'  
    ))
```

La siguiente instrucción ejecuta la misma consulta, pero elimina el argumento con nombre opcional `query` y el operador de flecha `=>`.

```
SELECT * FROM TABLE(  
    system.query(  
        'SELECT * FROM customer LIMIT 10;'  
    ))
```

## Calificadores de nombres de tablas federadas y Athena

Athena utiliza los siguientes términos para referirse a las jerarquías de los objetos de datos:

- Origen de datos: un grupo de bases de datos.
- Base de datos: un grupo de tablas.
- Tabla: datos organizados como un grupo de filas o columnas.

En ocasiones, también se hace referencia a estos objetos con nombres alternativos pero equivalentes, como los siguientes:

- Un origen de datos es lo que a veces se denomina catálogo.
- Una base de datos es lo que a veces se denomina esquema.

En la siguiente consulta de ejemplo en la consola de Athena, se utiliza el origen de datos `awsdatacatalog`, la base de datos `default` y la tabla `some_table`.

The screenshot shows the Amazon Athena console interface. At the top, there are tabs for 'Editor', 'Recent queries', 'Saved queries', and 'Settings'. The 'Workgroup' is set to 'primary'. On the left, the 'Data' panel shows 'Data source' as 'AwsDataCatalog' and 'Database' as 'default'. Under 'Tables and views', 'some\_table' is selected. The main editor shows a SQL query: `SELECT * FROM "awsdatacatalog"."default"."some_table" limit 10;`. Below the query, there are buttons for 'Run', 'Explain', 'Cancel', 'Clear', and 'Create'. The 'Query results' section shows the query is 'Completed' with a run time of 6.535 sec and 0.91 KB of data scanned. The results are displayed in a table with 5 rows.

#	id	data	category
1	1	a	A
2	3	d	d1
3	4	e	e1
4	4	f	f1
5	2	b	b1

## Términos en orígenes de datos federados

Cuando consulte orígenes de datos federados, tenga en cuenta que es posible que el origen de datos subyacente no utilice la misma terminología que Athena. Tenga en cuenta esta distinción al escribir las consultas federadas. En las siguientes secciones se describe cómo los términos de objetos de datos de Athena se corresponden con los de los orígenes de datos federados.

## Amazon Redshift

Una base de datos de Amazon Redshift es un grupo de esquemas de Redshift que contiene un grupo de tablas de Redshift.

Athena	Redshift
Origen de datos de Redshift	Función de Lambda del conector de Redshift configurada para apuntar a una database de Redshift.
<code>data_source.database.table</code>	<code>database.schema.table</code>

### Consulta de ejemplo

```
SELECT * FROM
Athena_Redshift_connector_data_source.Redshift_schema_name.Redshift_table_name
```

Para obtener más información sobre este conector, consulte [Conector para Redshift de Amazon Athena](#).

### Cloudera Hive

Un servidor o clúster de Cloudera Hive es un grupo de bases de datos de Cloudera Hive que contiene un grupo de tablas de Cloudera Hive.

Athena	Hive
Origen de datos de Cloudera Hive	Función de Lambda del conector de Cloudera Hive configurada para apuntar a un server de Cloudera Hive.
<code>data_source.database.table</code>	<code>server.database.table</code>

### Consulta de ejemplo

```
SELECT * FROM
Athena_Cloudera_Hive_connector_data_source.Cloudera_Hive_database_name.Cloudera_Hive_table_name
```

Para obtener más información sobre este conector, consulte [Conector Cloudera Hive de Amazon Athena](#).

## Cloudera Impala

Un servidor o clúster de Impala es un grupo de bases de datos de Impala que contiene un grupo de tablas de Impala.

Athena	Impala
Origen de datos de Impala	Función de Lambda del conector de Impala configurada para apuntar a un server de Impala.
<code>data_source.database.table</code>	<code>server.database.table</code>

### Consulta de ejemplo

```
SELECT * FROM
Athena_Impala_connector_data_source.Impala_database_name.Impala_table_name
```

Para obtener más información sobre este conector, consulte [Conector Cloudera Impala de Amazon Athena](#).

## MySQL

Un servidor de MySQL es un grupo de bases de datos de MySQL que contiene un grupo de tablas de MySQL.

Athena	MySQL
Origen de datos de MySQL	Función de Lambda del conector de MySQL configurada para apuntar a un server de MySQL.
<code>data_source.database.table</code>	<code>server.database.table</code>

### Consulta de ejemplo

```
SELECT * FROM
Athena_MySQL_connector_data source.MySQL_database_name.MySQL_table_name
```

Para obtener más información sobre este conector, consulte [Conector para MySQL de Amazon Athena](#).

## Oracle

Un servidor (o base de datos) de Oracle es un grupo de esquemas de Oracle que contiene un grupo de tablas de Oracle.

Athena	Oracle
Origen de datos de Oracle	Función de Lambda del conector de Oracle configurada para apuntar a un server de Oracle.
<code>data_source.database.table</code>	<code>server.schema.table</code>

### Consulta de ejemplo

```
SELECT * FROM
Athena_Oracle_connector_data_source.Oracle_schema_name.Oracle_table_name
```

Para obtener más información sobre este conector, consulte [Conector Oracle de Amazon Athena](#).

## Postgres

Un servidor (o clúster) de Postgres es un grupo de bases de datos de Postgres. Una base de datos de Postgres es un grupo de esquemas de Postgres que contiene un grupo de tablas de Postgres.

Athena	Postgres
Origen de datos de Postgres	Función de Lambda del conector de Postgres configurada para apuntar a un server y una database de Postgres.
<code>data_source.database.table</code>	<code>server.database.schema.table</code>

### Consulta de ejemplo

```
SELECT * FROM
```

```
Athena_Postgres_connector_data_source.Postgres_schema_name.Postgres_table_name
```

Para obtener más información sobre este conector, consulte [Conector para PostgreSQL de Amazon Athena](#).

## Desarrollo de un conector de origen de datos mediante el SDK de Athena Query Federation

Para escribir sus propios [conectores de origen de datos](#), puede utilizar el [SDK de Athena Query Federation](#). El SDK de Athena Query Federation define un conjunto de interfaces y protocolos de cable que se pueden utilizar para permitir a Athena delegar partes de su plan de ejecución de consultas en el código que se escribe e implementa. El SDK incluye un conjunto de conectores y un conector de ejemplo.

También puede personalizar los [conectores prediseñados](#) de Amazon Athena para su propio uso. Puede modificar una copia del código de origen de GitHub y luego, utilizar la [herramienta de publicación de conectores](#) para crear su propio paquete de AWS Serverless Application Repository. Después de implementar el conector de esta manera, puede usarlo en sus consultas de Athena.

Para obtener información sobre cómo descargar el SDK e instrucciones detalladas para escribir su propio conector, consulte [Example Athena connector](#) (Conector de Athena de ejemplo) en GitHub.

## Conectores de orígenes de datos de Athena para Apache Spark

Algunos conectores de orígenes de datos de Athena están disponibles como conectores DSV2 de Spark. Los nombres de los conectores DSV2 de Spark tienen un sufijo -dsv2 (por ejemplo, athena-dynamodb-dsv2).

A continuación, se muestran los conectores DSV2 disponibles actualmente, su nombre de clase `.format()` de Spark y los enlaces a la documentación correspondiente de Amazon Athena Federated Query:

Conector DSV2	Nombre de clase <code>.format()</code> de Spark	Documentación
athena-cloudwatch-dsv2	<code>com.amazonaws.athena.connectors.dsv2.cloudwatch.CloudwatchTableProvider</code>	<a href="#">CloudWatch</a>

Conector DSV2	Nombre de clase .format() de Spark	Documentación
athena-cloudwatch-metrics-dsv2	com.amazonaws.athena.connectors.dsv2.cloudwatch.metrics.CloudwatchMetricsTableProvider	<a href="#">Métricas de CloudWatch</a>
athena-aws-cmdb-dsv2	com.amazonaws.athena.connectors.dsv2.aws.cmdb.AwsCmdbTableProvider	<a href="#">CMDB</a>
athena-dynamodb-dsv2	com.amazonaws.athena.connectors.dsv2.dynamodb.DDBTableProvider	<a href="#">DynamoDB</a>

Para descargar archivos `.jar` para los conectores DSV2, visite la página de GitHub [Amazon Athena Query Federation DSV2](#) y consulte la sección Releases (Versiones), Release **<version>** (Versión <version>), Assets (Activos).

### Especificación del archivo jar para Spark

Para usar los conectores DSV2 de Athena con Spark, envíe el archivo `.jar` del conector al entorno de Spark que esté utilizando. En las siguientes secciones se describen casos específicos.

### Athena para Spark

Para obtener información sobre cómo agregar archivos `.jar` personalizados y configuraciones personalizadas a Amazon Athena para Apache Spark, consulte [Adición de archivos JAR y configuración personalizada de Spark](#).

### Spark general

Para pasar el archivo `.jar` del conector a Spark, utilice el comando `spark-submit` y especifique el archivo `.jar` en la opción `--jars`, como en el siguiente ejemplo:

```
spark-submit \  
  --deploy-mode cluster \  
  --jars https://github.com/aws-labs/aws-athena-query-federation-dsv2/releases/  
download/some_version/athena-dynamodb-dsv2-some_version.jar
```

## Amazon EMR Spark

Para ejecutar un comando `spark-submit` con el parámetro `--jars` en Amazon EMR, debe agregar un paso a su clúster de Amazon EMR Spark. Para obtener más información sobre cómo usar `spark-submit` en Amazon EMR, consulte [Add a Spark step](#) (Agregar un paso de Spark) en la Guía de publicación de Amazon EMR.

## ETL de AWS Glue para Spark

En el caso de ETL de AWS Glue, puede pasar la URL de GitHub.com del archivo `.jar` al argumento `--extra-jars` del comando `aws glue start-job-run`. En la documentación de AWS Glue, se describe que el parámetro `--extra-jars` toma una ruta de Amazon S3, pero el parámetro también puede tomar una URL HTTPS. Para obtener más información, consulte [Job parameter reference](#) (Referencia de parámetros de trabajos) en la Guía para desarrolladores de AWS Glue.

## Consultas del conector de Spark

Para enviar el equivalente a su consulta federada de Athena existente en Apache Spark, use la función `spark.sql()`. Por ejemplo, supongamos que tiene la siguiente consulta de Athena que desea utilizar en Apache Spark.

```
SELECT somecola, somecolb, somecolc  
FROM ddb_datasource.some_schema_or_glue_database.some_ddb_or_glue_table  
WHERE somecola > 1
```

Para realizar la misma consulta en Spark mediante el conector DynamoDB DSV2 de Amazon Athena, utilice el siguiente código:

```
dynamoDf = (spark.read  
  .option("athena.connectors.schema", "some_schema_or_glue_database")  
  .option("athena.connectors.table", "some_ddb_or_glue_table")  
  .format("com.amazonaws.athena.connectors.dsv2.dynamodb.DDBTableProvider")  
  .load())
```



```
dynamoDf.createOrReplaceTempView("ddb_spark_table")

spark.sql('''
SELECT somecola, somecolb, somecolc
FROM ddb_spark_table
WHERE somecola > 1
''')
```

## Especificación de parámetros

Las versiones DSV2 de los conectores de orígenes de datos de Athena utilizan los mismos parámetros que los conectores de orígenes de datos de Athena correspondientes. Para obtener información sobre los parámetros, consulte la documentación del conector de origen de datos de Athena correspondiente.

En su código de PySpark, use la siguiente sintaxis para configurar sus parámetros.

```
spark.read.option("athena.connectors.conf.parameter", "value")
```

Por ejemplo, el siguiente código establece el parámetro `disable_projection_and_casing` del conector DynamoDB de Amazon Athena en `always`.

```
dynamoDf = (spark.read
    .option("athena.connectors.schema", "some_schema_or_glue_database")
    .option("athena.connectors.table", "some_ddb_or_glue_table")
    .option("athena.connectors.conf.disable_projection_and_casing", "always")
    .format("com.amazonaws.athena.connectors.dsv2.dynamodb.DDBTableProvider")
    .load())
```

## Políticas de IAM para acceder a catálogos de datos

Para controlar el acceso a los catálogos de datos, utilice permisos de IAM de nivel de recursos o políticas de IAM basadas en identidad.

El siguiente procedimiento es específico de Athena.

Para obtener información específica sobre IAM, consulte los enlaces que se enumeran al final de esta sección. Para obtener información sobre políticas de catálogo de datos JSON de ejemplo, consulte [Políticas de catálogos de datos de ejemplo](#).

Para utilizar el editor visual en la consola de IAM para crear una política de catálogo de datos

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación de la izquierda, elija Políticas y, a continuación, elija Crear política.
3. En la pestaña Editor visual, elija Elegir un servicio. A continuación, elija un servicio de Athena para agregar a la política.
4. Elija Seleccionar acciones y, a continuación, elija las acciones que desea añadir a la política. El editor visual muestra las acciones disponibles en Athena. Para obtener más información, consulte [Acciones, recursos y claves de condición de Amazon Athena](#) en la Referencia de autorizaciones de servicio.
5. Elija Añadir acciones para escribir una acción específica o utilice comodines (\*) para especificar varias acciones.

De forma predeterminada, la política que está creando permite las acciones que usted elija. Si eligió una o más acciones que admiten permisos en el nivel de recursos para el recurso `datacatalog` en Athena, el editor visual enumera el recurso `datacatalog`.

6. Elija Recursos para especificar catálogos de datos específicos para su política. Para ver políticas de catálogo de datos JSON de ejemplo, consulte [Políticas de catálogos de datos de ejemplo](#).
7. Especifique el recurso `datacatalog` como se indica a continuación:

```
arn:aws:athena:<region>:<user-account>:datacatalog/<datacatalog-name>
```

8. Elija Review policy (Revisar la política) y, a continuación, escriba un Name (Nombre) y una Description (Descripción) (opcional) para la política que está creando. Revise el resumen de la política para asegurarse de que ha concedido los permisos deseados.
9. Elija Create Policy (Crear política) para guardar la nueva política.
10. Asocie esta política basada en la identidad a un usuario, grupo o rol y especifique los recursos de `datacatalog` a los que pueden acceder.

Para obtener más información, consulte los siguientes temas en la Referencia de autorizaciones de servicio y la Guía del usuario de IAM:

- [Acciones, recursos y claves de condición de Amazon Athena](#)
- [Creación de políticas con el editor visual](#)
- [Adición y eliminación de políticas de IAM](#)

- [Control del acceso a los recursos](#)

Para ver políticas de catálogo de datos JSON de ejemplo, consulte [Políticas de catálogos de datos de ejemplo](#).

Para obtener información sobre los permisos de AWS Glue y los permisos del rastreador de AWS Glue, consulte [Configuración de permisos de IAM para AWS Glue](#) y [Requisitos previos del rastreador](#) en la Guía para desarrolladores de AWS Glue.

Para obtener una lista completa de las acciones de Amazon Athena, consulte los nombres de acciones de la API en la [Referencia de API de Amazon Athena](#).

## Políticas de catálogos de datos de ejemplo

En esta sección se incluyen políticas de ejemplo que puede utilizar para habilitar varias acciones en catálogos de datos.

Un catálogo de datos es un recurso de IAM administrado por Athena. Por lo tanto, si la política de catálogo de datos utiliza acciones que toman `datacatalog` como entrada, debe especificar el ARN del catálogo de datos de la siguiente manera:

```
"Resource": [arn:aws:athena:<region>:<user-account>:datacatalog/<datacatalog-name>]
```

El `<datacatalog-name>` es el nombre del catálogo de datos. Por ejemplo, para un catálogo de datos denominado `test_datacatalog`, especifíquelo como recurso de la siguiente manera:

```
"Resource": ["arn:aws:athena:us-east-1:123456789012:datacatalog/test_datacatalog"]
```

Para obtener una lista completa de las acciones de Amazon Athena, consulte los nombres de acciones de la API en la [Referencia de API de Amazon Athena](#). Para obtener más información sobre las políticas de IAM, consulte [Creación de políticas con el editor visual](#) en la Guía del usuario de IAM. Para obtener más información acerca de la creación de políticas de IAM para grupos de trabajo, consulte [Políticas de IAM para acceder a catálogos de datos](#).

- [Example Policy for Full Access to All Data Catalogs](#)
- [Example Policy for Full Access to a Specified Data Catalog](#)
- [Example Policy for Querying a Specified Data Catalog](#)
- [Example Policy for Management Operations on a Specified Data Catalog](#)

- [Example Policy for Listing Data Catalogs](#)
- [Example Policy for Metadata Operations on Data Catalogs](#)

Example Ejemplo de política para el acceso completo a todos los catálogos de datos

La siguiente política permite el acceso completo a todos los recursos de catálogos de datos que podrían existir en la cuenta. Le recomendamos que utilice esta política para aquellos usuarios en su cuenta que deben administrar catálogos de datos para los demás usuarios.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:*"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Example Ejemplo de política para el acceso completo a un catálogo de datos especificado

La siguiente política permite acceso completo al recurso de catálogo de datos específico individual, denominado `datacatalogA`. Puede utilizar esta política para los usuarios con control total sobre un catálogo de datos en particular.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:ListDataCatalogs",
        "athena:ListWorkGroups",
        "athena:GetDatabase",
        "athena:ListDatabases",
        "athena:ListTableMetadata",

```

```

        "athena:GetTableMetadata"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "athena:StartQueryExecution",
        "athena:GetQueryResults",
        "athena>DeleteNamedQuery",
        "athena:GetNamedQuery",
        "athena:ListQueryExecutions",
        "athena:StopQueryExecution",
        "athena:GetQueryResultsStream",
        "athena:ListNamedQueries",
        "athena:CreateNamedQuery",
        "athena:GetQueryExecution",
        "athena:BatchGetNamedQuery",
        "athena:BatchGetQueryExecution",
        "athena>DeleteWorkGroup",
        "athena:UpdateWorkGroup",
        "athena:GetWorkGroup",
        "athena:CreateWorkGroup"
    ],
    "Resource": [
        "arn:aws:athena:us-east-1:123456789012:workgroup/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "athena:CreateDataCatalog",
        "athena>DeleteDataCatalog",
        "athena:GetDataCatalog",
        "athena:GetDatabase",
        "athena:GetTableMetadata",
        "athena:ListDatabases",
        "athena:ListTableMetadata",
        "athena:UpdateDataCatalog"
    ],
    "Resource": "arn:aws:athena:us-east-1:123456789012:datacatalog/datacatalogA"
}
]

```

```
}

```

### Example Ejemplo de política para consultar un catálogo de datos especificado

En la siguiente política, un usuario puede ejecutar consultas en el `datacatalogA` especificado. El usuario no tiene permiso para realizar las tareas de administración para el catálogo de datos en sí, como, por ejemplo, actualizarlo o eliminarlo.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:StartQueryExecution"
      ],
      "Resource": [
        "arn:aws:athena:us-east-1:123456789012:workgroup/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "athena:GetDataCatalog"
      ],
      "Resource": [
        "arn:aws:athena:us-east-1:123456789012:datacatalog/datacatalogA"
      ]
    }
  ]
}
```

### Example Ejemplo de política para operaciones de administración en un catálogo de datos especificado

En la siguiente política, se permite a un usuario crear, eliminar, obtener detalles y actualizar un catálogo de datos `datacatalogA`.

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

```

    "Effect": "Allow",
    "Action": [
      "athena:CreateDataCatalog",
      "athena:GetDataCatalog",
      "athena>DeleteDataCatalog",
      "athena:UpdateDataCatalog"
    ],
    "Resource": [
      "arn:aws:athena:us-east-1:123456789012:datacatalog/datacatalogA"
    ]
  }
]
}

```

Example Ejemplo de política para generar listas de catálogos de datos

La siguiente política permite a todos los usuarios enumerar todos los catálogos de datos:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:ListDataCatalogs"
      ],
      "Resource": "*"
    }
  ]
}

```

Example Ejemplo de política para realizar operaciones de metadatos en catálogos de datos

La siguiente política permite operaciones de metadatos en catálogos de datos:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:GetDatabase",

```

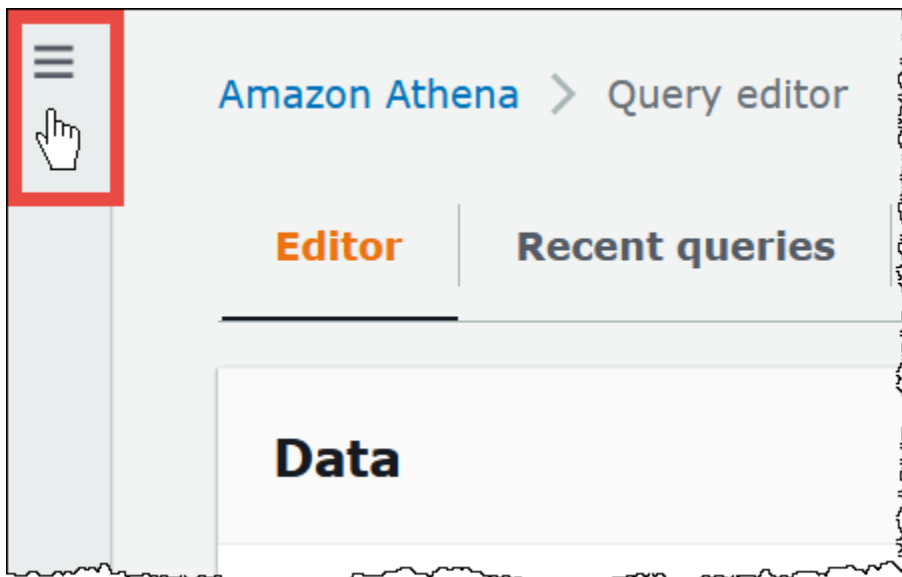
```
        "athena:GetTableMetadata",
        "athena:ListDatabases",
        "athena:ListTableMetadata"
    ],
    "Resource": "*"
}
]
```

## Administración de orígenes de datos

Puede utilizar la página Data Sources (Orígenes de datos) de la consola de Athena para administrar los orígenes de datos que cree.

Para ver un origen de datos, realice el siguiente procedimiento:

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.



3. En el panel de navegación, elija Data source (Orígenes de datos).
4. En la lista de orígenes de datos, elija el nombre del origen de datos que quiera ver.



 Note

Los elementos de la columna Data source name (Nombre del origen de datos) corresponden al resultado de la acción de la API [ListDataCatalogs](#) y del comando de la CLI [list-data-catalogs](#).

Para editar un origen de datos, realice el siguiente procedimiento:

1. En la página Data sources (Orígenes de datos), haga una de las siguientes acciones:
  - Seleccione el botón situado junto al nombre del catálogo y, a continuación, elija Actions (Acciones), Edit (Editar).
  - Elija el nombre del origen de datos. A continuación, en la página de detalles, elija Actions (Acciones), Edit (Editar).
2. En la página Edit (Editar), puede elegir otra función de Lambda para el origen de datos, cambiar la descripción o agregar etiquetas personalizadas. Para obtener más información acerca de las etiquetas, consulte [Etiquetado de recursos de Athena](#).
3. Elija Guardar.
4. Para editar el origen de datos AwsDataCatalog, elija el enlace AwsDataCatalog para abrir la página de detalles. A continuación, en la página de detalles, elija el enlace a la consola de AWS Glue, donde podrá editar el catálogo.

Para compartir un origen de datos

Para obtener información sobre cómo compartir orígenes de datos, consulte los enlaces siguientes.

- Para orígenes de datos basados en Lambda que no sean de Hive, consulte [Habilitación de las consultas federadas entre cuentas](#).
- Para AWS Glue Data Catalog, consulte [Acceso entre cuentas a los catálogos de datos de AWS Glue](#).

Eliminación de un origen de datos

1. En la página Data sources (Orígenes de datos), haga una de las siguientes acciones:

- Seleccione el botón situado junto al nombre del catálogo y, a continuación, elija Actions (Acciones), Delete (Eliminar).
- Elija el nombre del origen de datos y, a continuación, en la página de detalles, elija Actions, Delete

 Note

AwsDataCatalog es el origen de datos predeterminado de la cuenta y no se puede eliminar.

Se le advierte que, al eliminar un origen de datos, el catálogo de datos, las tablas y las vistas correspondientes se eliminan del editor de consultas. Las consultas guardadas que utilizaban el origen de datos dejarán de ejecutarse en Athena.

2. Para confirmar la eliminación, escriba el nombre del origen de datos y, a continuación, elija Delete (Borrar).

## Uso de Amazon DataZone en Athena

Puede utilizar [Amazon DataZone](#) para compartir, buscar y descubrir datos a escala más allá de los límites de la organización. DataZone simplifica su experiencia en servicios de análisis de AWS como Athena, AWS Glue y AWS Lake Formation. Por ejemplo, si tiene petabytes de datos en diferentes orígenes de datos, puede utilizar Amazon DataZone para crear agrupaciones de personas, datos y herramientas basadas en casos de uso empresarial. Para obtener más información, consulte [¿Qué es Amazon DataZone?](#)

En Athena, puede utilizar el editor de consultas para acceder a los entornos de DataZone y consultarlos. Un entorno DataZone especifica una combinación de proyecto y dominio de DataZone. Cuando utiliza un entorno DataZone desde la consola de Athena, asume el rol de IAM del entorno DataZone y solo ve las bases de datos y las tablas que pertenecen a ese entorno. Los permisos vienen determinados por los roles que especifique en DataZone.

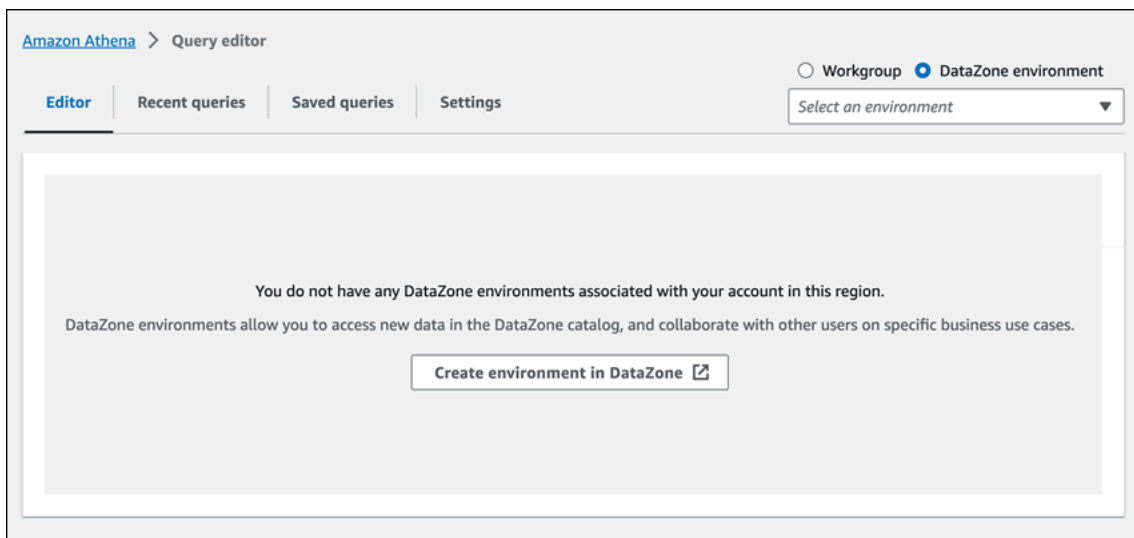
En Athena, puede usar el selector de entorno DataZone en la página del editor de consultas para elegir un entorno DataZone.

## Para abrir un entorno DataZone en Athena

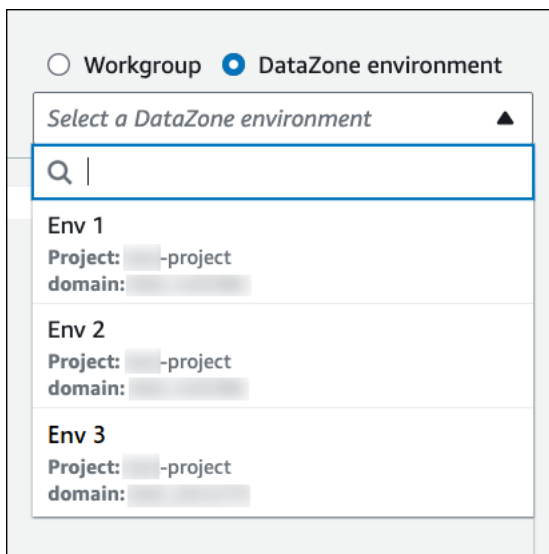
1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. En la parte superior derecha de la consola de Athena, junto a Workgroup, seleccione Entorno DataZone.

### Note

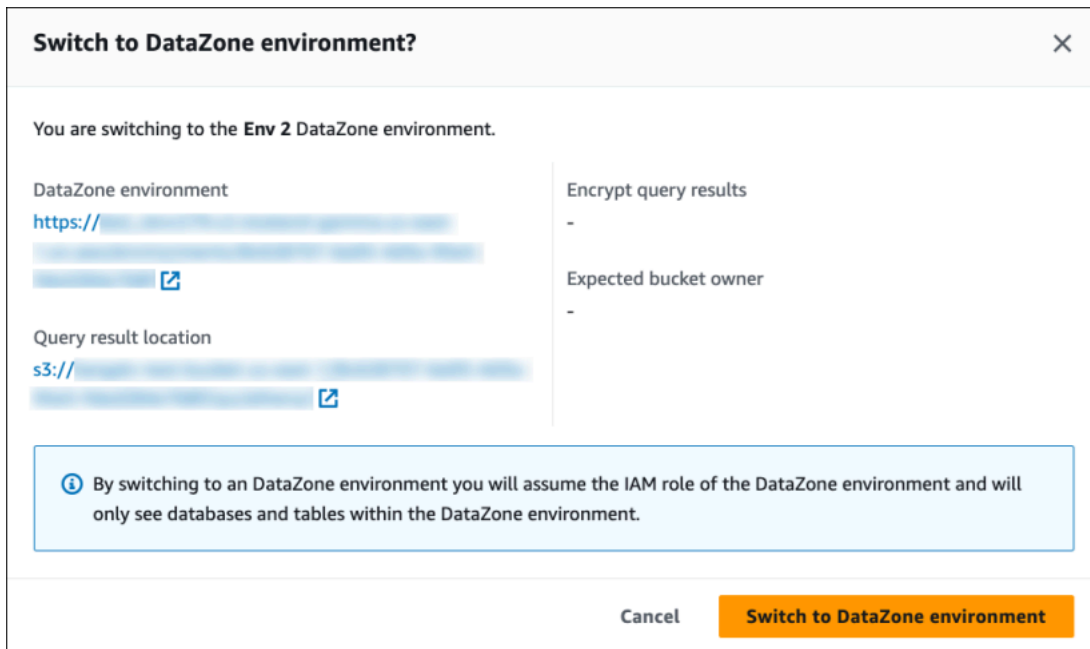
La opción DataZone environment solo está presente cuando tiene uno o más dominios disponibles en DataZone.



3. Utilice el selector de DataZone environment para elegir un entorno DataZone.



4. En el cuadro de diálogo Cambiar al entorno de DataZone, compruebe que el entorno es el que desea y, a continuación, seleccione Cambiar al entorno de DataZone.



Para obtener más información sobre cómo empezar a utilizar DataZone y Athena, consulte el tutorial [Introducción](#) en Guía del usuario de Amazon DataZone.

## Conexión a Amazon Athena con controladores ODBC y JDBC

Para explorar y visualizar sus datos con herramientas de inteligencia empresarial, descargue, instale y configure un controlador Open Database Connectivity (ODBC, Conectividad abierta de bases de datos) o Java Database Connectivity (JDBC, Conectividad de bases de datos Java).

### Temas

- [Conexión a Amazon Athena con JDBC](#)
- [Conexión a Amazon Athena con ODBC](#)

Consulte también los siguientes temas del Centro de Conocimientos de AWS y del blog de big data de AWS:

- [¿Cómo puedo utilizar mis credenciales de rol de IAM o cambiar a otro rol de IAM cuando me conecto a Athena mediante el controlador JDBC?](#)

- [Configuración de confianza entre ADFS y AWS y uso de las credenciales de Active Directory para conectarse a Amazon Athena con el controlador ODBC](#)

## Conexión a Amazon Athena con JDBC

Amazon Athena ofrece dos controladores JDBC, las versiones 2.x y 3.x. El controlador Athena JDBC 3.x es un controlador de nueva generación que ofrece mejor rendimiento y compatibilidad. El controlador JDBC 3.x es compatible con la lectura de los resultados de las consultas directamente desde Amazon S3, lo que mejora el rendimiento de las aplicaciones que consumen resultados de consultas de gran tamaño. El nuevo controlador también tiene menos dependencias de terceros, lo que facilita la integración con las herramientas de inteligencia empresarial y las aplicaciones personalizadas. En la mayoría de los casos, puede utilizar el nuevo controlador con cambios mínimos o nulos en la configuración existente.

- Para descargar el controlador JDBC 3.x, consulte [Controlador JDBC 3.x de Athena](#).
- Para descargar el controlador JDBC 2.x, consulte [Controlador JDBC 2.x de Athena](#).

### Temas

- [Controlador JDBC 3.x de Athena](#)
- [Controlador JDBC 2.x de Athena](#)

## Controlador JDBC 3.x de Athena

Puede utilizar el controlador Athena JDBC para conectarse a Amazon Athena desde distintas aplicaciones y herramientas de cliente SQL de terceros.

### Requisitos del sistema

- Entorno de tiempo de ejecución Java 8 (o superior)
- Contar con al menos 20 MB de espacio disponible en el disco

### Consideraciones y limitaciones

A continuación, se presentan algunas consideraciones y limitaciones del controlador Athena JDBC 3.x.

- Registro: el controlador 3.x utiliza [SLF4J](#), que es una capa de abstracción que permite el uso de algunos sistemas de registro en tiempo de ejecución.
- Cifrado: al utilizar el buscador de Amazon S3 con la opción de cifrado CSE\_KMS, el cliente de Amazon S3 no puede descifrar el resultado almacenado en el bucket de Amazon S3. Si necesita cifrado CSE\_KMS, puede seguir utilizando el buscador de transmisiones. Se prevé habilitar soporte para el cifrado CSE\_KMS con el buscador de Amazon S3.

## Descarga del controlador JDBC 3.x

Esta sección contiene información sobre la descarga y la licencia del controlador JDBC 3.x.

### Important

Cuando utilice el controlador JDBC 3.x, asegúrese de tener en cuenta los siguientes requisitos:

- Abrir el puerto 444: mantenga el puerto 444, que Athena utiliza para transmitir los resultados de las consultas, abierto al tráfico saliente. Cuando utilice un punto de conexión de PrivateLink para conectarse a Athena, asegúrese de que el grupo de seguridad asociado al punto de conexión de PrivateLink esté abierto al tráfico entrante en el puerto 444.
- Athena: política de flujo de resultados de consulta: agregue la acción de política `athena:GetQueryResultsStream` para las entidades principales de IAM que utilizan el controlador JDBC. Esta acción de política no se expone directamente con la API. Solo se utiliza con los controladores ODBC y JDBC como parte de la compatibilidad con los resultados de transmisión. Para ver una política de ejemplo, consulte [Política administrada de AWS: AWSQuicksightAthenaAccess](#).

Para descargar el controlador JDBC 3.x de Amazon Athena, visite los siguientes enlaces.

## Controlador uber jar JDBC

La siguiente descarga empaqueta el controlador y todas sus dependencias en el mismo archivo `.jar`. Esta descarga se suele utilizar para clientes SQL de terceros.

### [3.2.0 uber jar](#)

## Controlador lean jar JDBC

La siguiente descarga es un archivo .zip que contiene la versión lean .jar del controlador y archivos .jar separados para las dependencias del controlador. Esta descarga se suele utilizar para aplicaciones personalizadas que pueden tener dependencias que entren en conflicto con las dependencias que utiliza el controlador. Esta descarga resulta útil si desea elegir qué dependencias de controladores desea incluir en el lean jar y cuáles excluir si la aplicación personalizada ya contiene una o más de ellas.

### [3.2.0 lean jar](#)

#### Licencia

El siguiente enlace contiene el acuerdo de licencia del controlador JDBC 3.x.

#### [Licencia](#)

#### Temas

- [Introducción al controlador JDBC 3.x](#)
- [Parámetros de conexión JDBC 3.x de Amazon Athena](#)
- [Otra configuración de JDBC 3.x](#)
- [Notas de la versión de JDBC 3.x de Amazon Athena](#)
- [Versiones anteriores del controlador JDBC 3.x de Athena](#)

#### Introducción al controlador JDBC 3.x

Utilice la información de esta sección para empezar a usar el controlador JDBC 3.x de Amazon Athena.

#### Temas

- [Instrucciones de instalación](#)
- [Ejecución del controlador](#)
- [Configuración del controlador](#)
- [Actualización desde el controlador Athena JDBC v2](#)

## Instrucciones de instalación

Puede utilizar el controlador JDBC 3.x en una aplicación personalizada o desde un cliente SQL de terceros.

### En una aplicación personalizada

Descargue el archivo .zip que contiene el archivo jar del controlador y sus dependencias. Cada dependencia tiene su propio archivo .jar. Agregue el archivo jar del controlador como una dependencia en la aplicación personalizada. Agregue de forma selectiva las dependencias del contenedor de controladores en función de si ya las ha agregado a su aplicación desde otra fuente.

### En un cliente SQL de terceros

Descargue el archivo uber jar del controlador y agréguelo al cliente SQL de terceros siguiendo las instrucciones de ese cliente.

## Ejecución del controlador

Para ejecutar el controlador, puede utilizar una aplicación personalizada o un cliente SQL de terceros.

### En una aplicación personalizada

Utilice la interfaz de JDBC para interactuar con el controlador JDBC desde un programa. El siguiente código muestra un ejemplo de aplicación Java personalizada.

```
public static void main(String args[]) throws SQLException {
    Properties connectionParameters = new Properties();
    connectionParameters.setProperty("Workgroup", "primary");
    connectionParameters.setProperty("Region", "us-east-2");
    connectionParameters.setProperty("Catalog", "AwsDataCatalog");
    connectionParameters.setProperty("Database", "sampledatabase");
    connectionParameters.setProperty("OutputLocation", "s3://DOC-EXAMPLE-BUCKET");
    connectionParameters.setProperty("CredentialsProvider", "DefaultChain");
    String url = "jdbc:athena://";
    AthenaDriver driver = new AthenaDriver();
    Connection connection = driver.connect(url, connectionParameters);
    Statement statement = connection.createStatement();
    String query = "SELECT * from sample_table LIMIT 10";
    ResultSet resultSet = statement.executeQuery(query);
    printResults(resultSet); // A custom-defined method for iterating over a
                            // result set and printing its contents
}
```



```
}
```

## En un cliente SQL de terceros

Siga la documentación del cliente SQL que esté utilizando. Normalmente, se utiliza la interfaz gráfica de usuario del cliente SQL para introducir y enviar la consulta, y los resultados de la consulta se muestran en la misma interfaz.

## Configuración del controlador

Puede usar los parámetros de conexión para configurar el controlador JDBC de Amazon Athena. Para obtener información sobre los parámetros de conexión compatibles, consulte [Parámetros de conexión JDBC 3.x de Amazon Athena](#).

## En una aplicación personalizada

Para configurar los parámetros de conexión del controlador JDBC en una aplicación personalizada, realice una de las siguientes acciones:

- Añada los nombres de los parámetros y sus valores a un objeto `Properties`. Cuando llame a `Connection#connect`, pase ese objeto junto con la URL. Por ejemplo, consulte la aplicación de muestra Java en [Ejecución del controlador](#).
- En la cadena de conexión (la URL), utilice el siguiente formato para añadir los nombres de los parámetros y sus valores directamente después del prefijo del protocolo.

```
<parameterName>=<parameterValue>;
```

Utilice un punto y coma al final de cada par de nombre y valor de parámetro y no deje ningún espacio en blanco después del punto y coma, como en el siguiente ejemplo.

```
String url = "jdbc:athena://WorkGroup=primary;Region=us-east-1;...";AthenaDriver  
driver = new AthenaDriver();Connection connection = driver.connect(url, null);
```

### Note

Si se especifica un parámetro tanto en la cadena de conexión como en el objeto `Properties`, el valor de la cadena de conexión tiene prioridad. No se recomienda especificar el mismo parámetro en ambos lugares.

- Agregue los valores de los parámetros como argumentos a los métodos de `AthenaDataSource`, como en el siguiente ejemplo.

```
AthenaDataSource dataSource = new AthenaDataSource();
    dataSource.setWorkGroup("primary");
    dataSource.setRegion("us-east-2");
    ...
    Connection connection = dataSource.getConnection();
    ...
```

## En un cliente SQL de terceros

Siga las instrucciones del cliente SQL que esté utilizando. Normalmente, el cliente proporciona una interfaz gráfica de usuario para introducir los nombres de los parámetros y sus valores.

## Actualización desde el controlador Athena JDBC v2

La mayoría de los parámetros de conexión de la versión 3 de JDBC son compatibles con versiones anteriores de la versión 2 (Simba) del controlador JDBC. Esto significa que una cadena de conexión de la versión 2 se puede reutilizar con la versión 3 del controlador. Sin embargo, algunos parámetros de conexión han cambiado. Estos cambios se describen aquí. Cuando actualice a la versión 3 del controlador JDBC, actualice la configuración existente si es necesario.

## Clase de controlador

Algunas herramientas de BI le piden que proporcione la clase de controlador del archivo del controlador JDBC `.jar`. La mayoría de las herramientas encuentran esta clase automáticamente. El nombre completo de la clase en el controlador de la versión 3 es `com.amazon.athena.jdbc.AthenaDriver`. En el controlador de la versión 2, la clase era `com.simba.athena.jdbc.Driver`.

## Cadena de conexión

El controlador de la versión 3 utiliza `jdbc:athena://` para el protocolo al principio de la URL de la cadena de conexión de JDBC. El controlador de la versión 3 también es compatible con el protocolo de la versión 2 de `jdbc:awsathena://`; sin embargo, el uso del protocolo de la versión 2 está obsoleto. Para evitar comportamientos indefinidos, la versión 3 no acepta cadenas de conexión que comiencen con `jdbc:awsathena://` si la versión 2 (o cualquier otro controlador que acepte cadenas de conexión que comiencen con `jdbc:awsathena://`) se registró en la clase [DriverManager](#).

## Proveedores de credenciales

El controlador de la versión 2 utiliza nombres totalmente cualificados para identificar a los distintos proveedores de credenciales (por ejemplo, `com.simba.athena.amazonaws.auth.DefaultAWSCredentialsProviderChain`). El controlador de la versión 3 utiliza nombres más cortos (por ejemplo, `DefaultChain`). Los nuevos nombres se describen en las secciones correspondientes de cada proveedor de credenciales.

Los proveedores de credenciales personalizados escritos para la versión 2 del controlador deben modificarse para que la versión 3 del controlador implemente la interfaz [AwsCredentialsProvider](#) desde el nuevo AWS SDK for Java en lugar de la interfaz [AWSCredentialsProvider](#) del AWS SDK for Java anterior.

El `PropertiesFileCredentialsProvider` no es compatible con el controlador JDBC 3.x. El proveedor se utilizó en el controlador JDBC 2.x, pero pertenece a la versión anterior del AWS SDK para Java, que está a punto de finalizar su período de soporte. Para lograr la misma funcionalidad en el controlador JDBC 3.x, utilice el proveedor [Credenciales del perfil de configuración AWS](#) en su lugar.

### Nivel de registro

En la siguiente tabla se muestran las diferencias en los parámetros `LogLevel` de los controladores JDBC de la versión 2 y la versión 3.

Versión del controlador JDBC	Nombre del parámetro	Tipo de parámetro	Valor predeterminado	Valores posibles	Ejemplo de la cadena de conexión
v2	<code>LogLevel</code>	Opcional	0	De 0 a 6	<code>LogLevel=6;</code>
v3	<code>LogLevel</code>	Opcional	SEGUIMIENTO	DESACTIVADO, ERROR, ADVERTENCIA, INFORMACIÓN, DEPURACIÓN, RASTREO	<code>LogLevel=INFO;</code>

## Recuperación del ID de consulta

En el controlador de la versión 2, se desempaqueta una instancia `Statement` en `com.interfaces.core.IStatementQueryInfoProvider`, que es una interfaz que tiene dos métodos: `#getPReparedQueryId` y `#getQueryId`. Puedes usar estos métodos para obtener el identificador de ejecución de una consulta que se ha ejecutado.

En el controlador de la versión 3, desempaquete las instancias `Statement`, `PreparedStatement`, y `ResultSet` en la interfaz `com.amazon.athena.jdbc.AthenaResultSet`. La interfaz tiene un método: `#getQueryExecutionId`.

## Parámetros de conexión JDBC 3.x de Amazon Athena

Los parámetros de conexión compatibles se dividen aquí en tres secciones: [Parámetros de conexión básicos](#), [Parámetros de conexión avanzados](#) y [Parámetros de conexión de autenticación](#). Las secciones [Parámetros de conexión avanzados](#) y [Parámetros de conexión de autenticación](#) tienen subsecciones que agrupan parámetros relacionados.

### Temas

- [Parámetros de conexión básicos](#)
- [Parámetros de conexión avanzados](#)
- [Parámetros de conexión de autenticación](#)

## Parámetros de conexión básicos

En las siguientes secciones se describen los parámetros de conexión básicos del controlador JDBC 3.x.

### Región

El lugar Región de AWS donde se ejecutarán las consultas. Para obtener una lista de las regiones, consulte [Puntos de conexión y cuotas de Amazon Athena](#).

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
Región	AwsRegion (obsoleto)	Obligatorio (pero si no se proporciona, se buscará mediante la cadena <a href="#">DefaultAWSRegionProviderChain</a> )	Ninguno

## Catálogo

El catálogo que contiene las bases de datos y las tablas a las que se accederá con el controlador. Para obtener más información sobre los catálogos, consulte [DataCatalog](#).

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
Catálogo	Ninguno	Opcional	AwsDataCatalog

## Base de datos

La base de datos en la que se ejecutarán las consultas. Las tablas que no estén calificadas explícitamente con un nombre de base de datos se resuelven en esta base de datos. Para obtener información sobre las bases de datos, consulte [Base de datos](#).

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
Base de datos	Esquema	Opcional	predeterminada

## Grupo de trabajo

El grupo de trabajo en el que se ejecutarán las consultas. Para obtener información acerca de los grupos de trabajo, consulte [Grupo de trabajo](#).

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
WorkGroup	Ninguno	Opcional	primary

### Ubicación de salida

Especifica la ubicación en Amazon S3 en la que se almacenan los resultados de las consultas. Para obtener información sobre la ubicación de las salidas, consulte [ResultConfiguration](#).

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
OutputLocation	S3OutputLocation (obsoleto)	Obligatorio (a menos que el grupo de trabajo especifique una ubicación de salida)	Ninguno

### Parámetros de conexión avanzados

En las siguientes secciones se describen los parámetros de conexión avanzados del controlador JDBC 3.x.

#### Temas

- [Parámetros de cifrado de resultado](#)
- [Parámetros de obtención de resultados](#)
- [Parámetros de reutilización de resultados de las consultas](#)
- [Parámetros de sondeo de ejecución de consultas](#)
- [Parámetros de invalidación del punto de conexión](#)
- [Parámetros de configuración del proxy](#)
- [Parámetros de registro](#)
- [Nombre de la aplicación](#)
- [Prueba de conexión](#)

- [Número de reintentos](#)

## Parámetros de cifrado de resultado

Tenga en cuenta los siguientes puntos:

- La clave AWS KMS debe especificarse cuando EncryptionOption es SSE\_KMS o CSE\_KMS.
- La clave AWS KMS no se puede especificar cuando EncryptionOption no se especifica o cuando EncryptionOption es SSE\_S3.

## Opciones de cifrado

El tipo de cifrado que se utilizará para los resultados de las consultas tal como se almacenan en Amazon S3. Para obtener más información sobre las opciones de cifrado, consulte [EncryptionConfiguration](#) en la Referencia de la API de Amazon Athena.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado	Valores posibles
EncryptionOption	S3OutputEncryptionOption (obsoleto)	Opcional	Ninguno	SSE_S3, SSE_KMS, CSE_KMS

## Clave de KMS

El ARN o ID de la clave KMS, si se elige SSE\_KMS o CSE\_KMS se elige como opción de cifrado. Para obtener más información, consulte [EncryptionConfiguration](#) en Referencia de la API de Amazon Athena.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
KmsKey	S3OutputEnckMSKey (obsoleto)	Opcional	Ninguno

## Parámetros de obtención de resultados

### Buscador de resultados

El buscador que se utilizará para descargar los resultados de las consultas.

El buscador de resultados predeterminado, S3, descarga los resultados de las consultas directamente desde Amazon S3 sin utilizar las API de Athena. Esta es la opción más rápida en la mayoría de los casos. Esta opción no está disponible si los resultados de la consulta están cifrados con CSE\_KMS o si la política que permite al usuario acceder a los resultados de la consulta solo permite llamadas de Athena usando `s3:CalledVia`.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado	Valores posibles
ResultFetcher	Ninguno	Opcional	S3	S3, GetQueryResults, GetQueryResultsStream

#### Note

En el controlador JDBC 2.x, la configuración `UseResultSetStreaming = 1` ajusta el controlador para que utilice la API de transmisión del conjunto de resultados. En el controlador JDBC 3.x, el ajuste equivalente es `ResultFetcher=GetQueryResultsStream`.

### Tamaño de recuperación

El valor de este parámetro se utiliza como mínimo para los búferes internos y como tamaño de la página de destino cuando se obtienen los resultados. El valor 0 (cero) significa que el controlador debe usar sus valores predeterminados tal y como se describe a continuación. El valor máximo es 1 000 000



Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
FetchSize	RowsToFetchPerBlock (obsoleto)	Opcional	0

- El buscador `GetQueryResults` siempre utilizará un tamaño de página de 1000, que es el valor máximo que admite la llamada a la API. Cuando el tamaño de la búsqueda es superior a 1000, se realizan varias llamadas sucesivas a la API para llenar el búfer por encima del mínimo.
- El buscador `GetQueryResultsStream` utilizará el tamaño de búsqueda configurado como tamaño de página, o 10 000 de forma predeterminada.
- El buscador `S3` utilizará el tamaño de búsqueda configurado como tamaño de página, o 10 000 de forma predeterminada.

## Parámetros de reutilización de resultados de las consultas

### Habilitar la reutilización de resultados

Especifica si se pueden reutilizar los resultados anteriores de la misma consulta cuando se ejecuta una consulta. Para obtener información sobre la reutilización de los resultados de la consulta, consulte [ResultReuseByAgeConfiguration](#).

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
EnableResultReuseByAge	Ninguno	Opcional	FALSO

### Antigüedad máxima de reutilización de resultados

Especifica, en minutos, la antigüedad máxima de un resultado de consulta anterior que Athena debe tener en cuenta para su reutilización. Para obtener información sobre la antigüedad máxima de reutilización de los resultados, consulte [ResultReuseByAgeConfiguration](#).

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
MaxResultReuseAgeInMinutes	Ninguno	Opcional	60

### Parámetros de sondeo de ejecución de consultas

#### Intervalo mínimo de sondeo de ejecución de consultas

Valor mínimo en milisegundos que debe esperarse antes de sondear Athena para determinar el estado de ejecución de la consulta.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
MinQueryExecutionPollingIntervalInMillis	MinQueryExecutionPollingInterval (obsoleto)	Opcional	100

#### Intervalo máximo de sondeo de ejecución de consultas

Valor máximo en milisegundos que debe esperarse antes de sondear Athena para determinar el estado de ejecución de la consulta.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
MaxQueryExecutionPollingIntervalInMillis	MaxQueryExecutionPollingInterval (obsoleto)	Opcional	5000

## Multiplicador del intervalo de sondeo de ejecución de consultas

Factor para aumentar el periodo de sondeo. De forma predeterminada, el sondeo comienza con el valor de `MinQueryExecutionPollingIntervalMillis` y se duplica con cada sondeo hasta alcanzar el valor de `MaxQueryExecutionPollingIntervalMillis`.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
<code>QueryExecutionPollingIntervalMultiplier</code>	Ninguno	Opcional	2

## Parámetros de invalidación del punto de conexión

### Anulación del punto de conexión de Athena

El punto de conexión que el controlador utilizará para realizar llamadas de API a Athena.

Tenga en cuenta los siguientes puntos:

- Si los protocolos `https://` o `http://` no se especifican en la URL proporcionada, el controlador inserta el prefijo `https://`.
- Si no se especifica este parámetro, el controlador utiliza un punto de conexión predeterminado.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
<code>AthenaEndpoint</code>	<code>EndpointOverride</code> (obsoleto)	Opcional	Ninguno

### Anulación del punto de conexión del servicio de transmisión de Athena

El punto de conexión que el controlador utilizará para descargar los resultados de las consultas cuando utilice el servicio de transmisión de Athena. El servicio de transmisión de Athena está disponible a través del puerto 444.

Tenga en cuenta los siguientes puntos:

- Si los protocolos `https://` o `http://` no se especifican en la URL proporcionada, el controlador inserta el prefijo `https://`.
- Si no se especifica ningún puerto en la URL proporcionada, el controlador inserta el puerto 444 del servicio de transmisión.
- Si no se especifica el parámetro `AthenaStreamingEndpoint`, el controlador utiliza la anulación `AthenaEndpoint`. Si no se especifican las anulaciones `AthenaStreamingEndpoint` o `AthenaEndpoint`, el controlador utiliza un punto de conexión de transmisión predeterminado.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
<code>AthenaStreamingEndpoint</code>	<code>StreamingEndpointOverride</code> (obsoleto)	Opcional	Ninguno

### Anulación del punto de conexión de Lake Formation

El punto de conexión que el controlador utilizará para el servicio Lake Formation cuando utilice la API AWS Lake Formation [AssumeDecoratedRoleWithSAML](#) para recuperar credenciales temporales. Si no se especifica este parámetro, el controlador utiliza un punto de conexión de Lake Formation predeterminado.

Tenga en cuenta los siguientes puntos:

- Si los protocolos `https://` o `http://` no se especifican en la URL proporcionada, el controlador inserta el prefijo `https://`.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
<code>LakeFormationEndpoint</code>	<code>LfEndpointOverride</code> (obsoleto)	Opcional	Ninguno

## Anulación del punto de conexión S3

El punto de conexión que el controlador utilizará para descargar los resultados de las consultas cuando utilice el buscador de Amazon S3. Si no se especifica este parámetro, el controlador utiliza un punto de conexión de Amazon S3 predeterminado.

Tenga en cuenta los siguientes puntos:

- Si los protocolos `https://` o `http://` no se especifican en la URL proporcionada, el controlador inserta el prefijo `https://`.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
S3Endpoint	Ninguna	Opcional	Ninguno

## Anulación del punto de conexión STS

El punto de conexión que el controlador utilizará para el servicio AWS STS cuando utilice la API AWS STS [AssumeRoleWithSAML](#) para recuperar credenciales temporales. Si no se especifica este parámetro, el controlador utiliza un punto de conexión AWS STS predeterminado.

Tenga en cuenta los siguientes puntos:

- Si los protocolos `https://` o `http://` no se especifican en la URL proporcionada, el controlador inserta el prefijo `https://`.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
StendPoint	StsEndpointOverride (obsoleto)	Opcional	Ninguno

## Parámetros de configuración del proxy

### Host del proxy.

La URL del host proxy. Utilice este parámetro si necesita que las solicitudes de Athena pasen por un proxy.

#### Note

Asegúrese de incluir el protocolo `https://` o `http://` al principio de la URL de `ProxyHost`.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
ProxyHost	Ninguno	Opcional	Ninguno

### Puerto del proxy.

El puerto que se utilizará en el host proxy. Utilice este parámetro si necesita que las solicitudes de Athena pasen por un proxy.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
ProxyPort	Ninguno	Opcional	Ninguno

### Nombre de usuario del proxy

Nombre de usuario para autenticarse con el servidor proxy. Utilice este parámetro si necesita que las solicitudes de Athena pasen por un proxy.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
ProxyUsername	ProxyUID (obsoleto)	Opcional	Ninguno

## Contraseña del proxy.

Contraseña para autenticarse en el servidor proxy. Utilice este parámetro si necesita que las solicitudes de Athena pasen por un proxy.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
ProxyPassword	ProxyPWD (obsoleto)	Opcional	Ninguno

## Hosts exentos de proxy

Conjunto de nombres de host a los que el controlador se conecta sin usar un proxy cuando el proxy está habilitado (es decir, cuando los parámetros de conexión ProxyHost y ProxyPort están configurados). Los hosts deben estar separados por el carácter de barra vertical (|) (por ejemplo, host1.com|host2.com).

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
ProxyExemptHosts	NonProxyHosts	Opcional	Ninguno

## Proxy habilitado para proveedores de identidad

Especifica si se debe utilizar un proxy cuando el controlador se conecte a un proveedor de identidad.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
El proxy está habilitado para el IDP	UseProxyForIdP	Opcional	FALSO

## Parámetros de registro

En esta sección se describen los parámetros relacionados con el registro.

## Nivel de registro

Especifica el nivel de registro del complemento. No se registra nada a menos que el parámetro `LogPath` también esté establecido.

### Note

Recomendamos configurar solo el parámetro `LogPath`, a menos que tenga requisitos especiales. Si se establece solo el parámetro `LogPath`, se habilita el registro y se utiliza el nivel de registro predeterminado `TRACE`. El nivel de registro `TRACE` proporciona el registro más detallado.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado	Valores posibles
<code>LogLevel</code>	Ninguno	Opcional	SEGUIMIENTO	DESACTIVADO, ERROR, ADVERTENCIA, INFORMACIÓN, DEPURACIÓN, RASTREO

## Ruta de registro

La ruta a un directorio del equipo en el que se ejecuta el controlador y en el que se guardarán los registros del controlador. Se creará un archivo de registro con un nombre único en el directorio especificado. Si está configurado, habilita el registro del controlador.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
<code>LogPath</code>	Ninguno	Opcional	Ninguno



## Nombre de la aplicación

El nombre de la aplicación que utiliza el controlador. Si se especifica un valor para este parámetro, el valor se incluye en la cadena del agente de usuario de las llamadas a la API que el controlador realiza a Athena.

### Note

También puede establecer el nombre de la aplicación llamando `setApplicationName` al objeto `DataSource`.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
<code>ApplicationName</code>	Ninguno	Opcional	Ninguno

## Prueba de conexión

Si se establece en `TRUE`, el controlador realiza una prueba de conexión cada vez que se crea una conexión JDBC, incluso si no se ejecuta una consulta en la conexión.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
<code>ConnectionTest</code>	Ninguno	Opcional	<code>TRUE</code>

### Note

Una prueba de conexión envía una consulta `SELECT 1` a Athena para comprobar que la conexión se ha configurado correctamente. Esto significa que se almacenarán dos archivos en Amazon S3 (el conjunto de resultados y los metadatos) y se pueden aplicar cargos adicionales de acuerdo con la política de [precios de Amazon Athena](#).

## Número de reintentos

Número máximo de veces que el controlador debe volver a enviar una solicitud recuperable a Athena.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
NumRetries	MaxErrorRetry (obsoleto)	Opcional	Ninguno

## Parámetros de conexión de autenticación

El controlador Athena JDBC 3.x admite varios métodos de autenticación. Los parámetros de conexión necesarios dependen del método de autenticación que utilice.

## Temas

- [Credenciales de IAM](#)
- [Credenciales predeterminadas](#)
- [Credenciales del perfil de configuración AWS](#)
- [Credenciales del perfil de instancia](#)
- [Credenciales personalizadas](#)
- [Credenciales JWT](#)
- [Credenciales de Azure AD](#)
- [Credenciales de Okta](#)
- [Credenciales de Ping](#)
- [Credenciales de AD FS](#)
- [Credenciales de Browser Azure AD](#)
- [Credenciales del navegador SALM](#)

## Credenciales de IAM

Puede utilizar sus credenciales de IAM con el controlador JDBC para conectarse a Amazon Athena configurando los siguientes parámetros de conexión.

## Usuario

El ID de la clave de acceso de AWS. Para obtener más información sobre las claves de acceso, consulte las [credenciales de seguridad de AWS](#) en la Guía del usuario de IAM

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
Usuario	AccessKeyId	Obligatoria	Ninguno

## Password

ID de su clave secreta de AWS. Para obtener más información sobre las claves de acceso, consulte las [credenciales de seguridad de AWS](#) en la Guía del usuario de IAM

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
Password	SecretAccessKey	Opcional	Ninguno

## Token de sesión

Si utiliza credenciales de AWS temporales, debe especificar el token de sesión. Para obtener información acerca de las credenciales temporales, consulte [Credenciales de seguridad temporales en IAM](#) en la Guía del usuario de IAM.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
SessionToken	Ninguno	Opcional	Ninguno

## Credenciales predeterminadas

Puede utilizar las credenciales predeterminadas que configure en su sistema cliente para conectarse a Amazon Athena configurando los siguientes parámetros de conexión. Para obtener información sobre el uso de las credenciales predeterminadas, consulte [Uso de la cadena de proveedores de credenciales predeterminadas](#) en la Guía para desarrolladores de AWS SDK for Java.

## Proveedor de credenciales

El proveedor de credenciales que se utilizará para autenticar las solicitudes de AWS. Establezca el valor del parámetro en `DefaultChain`.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado	Valor para usar
<code>CredentialsProvider</code>	<code>AWSCredentialsProviderClass</code> (obsoleto)	Obligatoria	Ninguno	<code>DefaultChain</code>

## Credenciales del perfil de configuración AWS

Puede utilizar las credenciales almacenadas en un perfil de configuración de AWS configurando los siguientes parámetros de conexión. Los perfiles de configuración de AWS se almacenan normalmente en archivos del directorio `~/.aws`). Para obtener información sobre los perfiles de configuración AWS, consulte [Uso de perfiles](#) en la Guía para desarrolladores de AWS SDK for Java.

## Proveedor de credenciales


El proveedor de credenciales que se utilizará para autenticar las solicitudes de AWS. Establezca el valor del parámetro en `ProfileCredentials`.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado	Valor para usar
<code>CredentialsProvider</code>	<code>AWSCredentialsProviderClass</code> (obsoleto)	Obligatoria	Ninguno	<code>ProfileCredentials</code>

## Nombre del perfil

El nombre del perfil de configuración de AWS cuyas credenciales deben usarse para autenticar la solicitud a Athena.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
ProfileName	Ninguno	Obligatoria	Ninguno

 Note

El nombre del perfil también se puede especificar como el valor del parámetro `CredentialsProviderArguments`, aunque está en desuso.

### Credenciales del perfil de instancia

Este tipo de autenticación se utiliza en instancias de Amazon EC2. Un perfil de instancia es un perfil adjunto a una instancia de Amazon EC2. El uso de un proveedor de credenciales de perfil de instancia delega la administración de las credenciales de AWS al Servicio de metadatos de instancias de Amazon EC2. Esto elimina la necesidad de que los desarrolladores almacenen en credenciales de forma permanente en la instancia de Amazon EC2 o se preocupen de rotar o administrar credenciales temporales.

### Proveedor de credenciales

El proveedor de credenciales que se utilizará para autenticar las solicitudes de AWS. Establezca el valor del parámetro en `InstanceProfile`.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado	Valor para usar
<code>CredentialsProvider</code>	<code>AWSCredentialsProviderClass</code> (obsoleto)	Obligatoria	Ninguno	<code>InstanceProfile</code>

### Credenciales personalizadas

Puede usar este tipo de autenticación para proporcionar sus propias credenciales mediante una clase Java que implemente la interfaz [AwsCredentialsProvider](#).

## Proveedor de credenciales

El proveedor de credenciales que se utilizará para autenticar las solicitudes de AWS. Establezca el valor de este parámetro en el nombre de clase completo de la clase personalizada que implementa la interfaz [AwsCredentialsProvider](#). En tiempo de ejecución, esa clase debe estar en la ruta de clases de Java de la aplicación que usa el controlador JDBC.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado	Valor para usar
CredentialsProvider	AWSCredentialsProviderClass (obsoleto)	Obligatoria	Ninguno	El nombre de clase completo de la implementación personalizada de <code>AwsCredentialsProvider</code>

## Argumentos del proveedor de credenciales

Lista separada por comas de argumentos de cadena para el constructor del proveedor de credenciales personalizadas.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
CredentialsProviderArguments	AwsCredentialsProviderArguments (obsoleto)	Opcional	Ninguno

## Credenciales JWT

Con este tipo de autenticación, puede utilizar un token web JSON (JWT) obtenido de un proveedor de identidad externo como parámetro de conexión para autenticarse con Athena. El proveedor externo de credenciales ya debe estar federado con AWS.

## Proveedor de credenciales

El proveedor de credenciales que se utilizará para autenticar las solicitudes de AWS. Establezca el valor del parámetro en JWT.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado	Valor para usar
CredentialsProvider	AWSCredentialsProviderClass (obsoleto)	Obligatoria	Ninguno	JWT

## Token de identidad web JWT

El token JWT obtenido de un proveedor de identidad federado externo. Este token se utilizará para autenticarse con Athena.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
JwtWebIdentityToken	web_identity_token (obsoleto)	Obligatoria	Ninguno

## Rol ARN de JWT

El nombre de recurso de Amazon (ARN) del rol que se asignará. Para obtener información acerca de los roles de ARN, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
JwtRoleArn	role_arn (obsoleto)	Obligatoria	Ninguno

## Nombre de sesión del rol JWT

El nombre de la sesión cuando se utilizan las credenciales de JWT para la autenticación. El nombre puede ser cualquier nombre que usted elija.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
JwtRoleSessionName	role_session_name (obsoleto)	Obligatoria	Ninguno

## Credenciales de Azure AD

Un mecanismo de autenticación basado en SAML que permite la autenticación en Athena mediante el proveedor de identidad de Azure AD. Este método supone que ya se ha establecido una federación entre Athena y Azure AD.

### Note

Algunos de los nombres de los parámetros de esta sección tienen alias. Los alias son equivalentes funcionales de los nombres de los parámetros y se han proporcionado para permitir la retrocompatibilidad con el controlador de JDBC 2.x. Debido a que los nombres de los parámetros se han mejorado para que sigan una convención de nomenclatura más clara y coherente, se recomienda utilizarlos en lugar de los alias, que han quedado obsoletos.

## Proveedor de credenciales

El proveedor de credenciales que se utilizará para autenticar las solicitudes de AWS. Establezca el valor del parámetro en AzureAD.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado	Valor para usar
CredentialsProvider	AWSCredentialsProviderClass (obsoleto)	Obligatoria	Ninguno	AzureAD



## Usuario

La dirección de correo electrónico del usuario de Azure AD que se utilizará para la autenticación con Azure AD.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
Usuario	UID (obsoleto)	Obligatoria	Ninguno

## Password

La contraseña del usuario de Azure AD.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
Password	PWD (obsoleto)	Obligatoria	Ninguno

## ID de inquilino de Azure AD

ID de inquilino de su aplicación de Azure AD.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
ID de inquilino de Azure	tenant_id (obsoleto)	Obligatoria	Ninguno

## ID de cliente de Azure AD

ID de cliente de su aplicación de Azure AD.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
AzureAdClientId	client_id (obsoleto)	Obligatoria	Ninguno

## Secreto de cliente de Azure

El secreto de cliente de su aplicación de Azure AD.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
El secreto del cliente de Azure Read	client_secret (obsoleto)	Obligatoria	Ninguno

## Rol preferido

El nombre de recurso de Amazon (ARN) del rol que se asignará. Para obtener información acerca de los roles de ARN, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
PreferredRole	preferred_role (obsoleto)	Opcional	Ninguno

## Duración de la sesión de rol

La duración de la sesión del rol en segundos. Para obtener más información, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
RoleSessionDuration	Duración (obsoleto)	Opcional	3600

## Lake Formation habilitada

Utiliza la acción de la API de Lake Formation [AssumeDecoratedRoleWithSAML](#) para recuperar las credenciales de IAM temporales en lugar de la acción de la API de AWS STS [AssumeRoleWithSAML](#).

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
LakeFormationEnabled	Ninguno	Opcional	FALSE

## Credenciales de Okta

Un mecanismo de autenticación basado en SAML que permite la autenticación en Athena mediante el proveedor de identidad de Okta. Este método supone que ya se ha establecido una federación entre Athena y Okta.

### Proveedor de credenciales

El proveedor de credenciales que se utilizará para autenticar las solicitudes de AWS. Establezca el valor del parámetro en Okta.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado	Valor para usar
CredentialsProvider	AWSCredentialsProviderClass (obsoleto)	Obligatoria	Ninguno	Okta

## Usuario

La dirección de correo electrónico del usuario de Okta que se utilizará para la autenticación con Okta.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
Usuario	UID (obsoleto)	Obligatoria	Ninguno

## Password

La contraseña del usuario Okta.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
Password	PWD (obsoleto)	Obligatoria	Ninguno

### Nombre de host de Okta

La URL de su organización de Okta. Puede extraer el parámetro `idp_host` de la URL de Incrustar enlace en la aplicación de Okta. Para ver los pasos, consulte [Recuperación de información de configuración de ODBC desde Okta](#). El primer segmento que sigue a `https://`, incluido `okta.com`, es el host del proveedor de identidad (por ejemplo, `trial-1234567.okta.com` para una URL que empiece por `https://trial-1234567.okta.com`).

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
OktaHostName	IdP_Host (obsoleto)	Obligatoria	Ninguno

### ID de aplicación de Okta

El identificador de dos partes de su aplicación. Puede extraer el ID de la aplicación de la URL del enlace incrustado en la aplicación de Okta. Para ver los pasos, consulte [Recuperación de información de configuración de ODBC desde Okta](#). El ID de la aplicación son los dos últimos segmentos de la URL, incluida la barra del medio. Los segmentos son dos cadenas de 20 caracteres con una combinación de números y letras mayúsculas y minúsculas (por ejemplo, `Abc1de2fghi3J45kL678/abc1defghij2k1mNo3p4`).

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
OktaAppId	App_ID (obsoleto)	Obligatoria	Ninguno

### Nombre de la aplicación de Okta

El nombre de su aplicación de Okta.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
OktaAppName	App_Name (obsoleto)	Obligatoria	Ninguno

### Tipo de MFA de Okta

Si ha configurado Okta para que requiera la autenticación multifactor (MFA), debe especificar el tipo de MFA de Okta y los parámetros adicionales en función del segundo factor que desee utilizar.

El tipo MFA de Okta es el segundo tipo de factor de autenticación (después de la contraseña) que se utiliza para autenticarse con Okta. Los segundos factores admitidos incluyen las notificaciones push enviadas a través de la aplicación Okta Verify y las contraseñas temporales de un solo uso (TOTP) generadas por Okta Verify, Google Authenticator o enviadas por SMS. Las políticas de seguridad de las organizaciones individuales determinan si se requiere o no la MFA para el inicio de sesión de los usuarios.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado	Valores posibles
OktaMfaType	okta_mfa_type (obsoleto)	Obligatorio, si Okta está configurado para requerir MFA	Ninguno	oktaverifywithpush, oktaverifywithtotp, googleauthenticator, smsauthentication

### Número de teléfono de Okta

El número de teléfono al que Okta enviará una contraseña temporal de un solo uso mediante SMS cuando se elija el tipo de MFA `smsauthentication`. El número de teléfono debe ser un número de teléfono de EE. UU. o Canadá.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
OktaPhoneNumber	okta_phone_number (obsoleto)	Obligatorio si OkTamfaType es smsauthentication	Ninguno

### Tiempo de espera de MFA de Okta

El tiempo, en segundos, que se tarda en esperar a que el usuario confirme una notificación automática de Okta antes de que el controlador emita una excepción de tiempo de espera.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
OktaMfaWaitTime	okta_mfa_wait_time (obsoleto)	Opcional	60

### Rol preferido

El nombre de recurso de Amazon (ARN) del rol que se asignará. Para obtener información acerca de los roles de ARN, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
PreferredRole	preferred_role (obsoleto)	Opcional	Ninguno

### Duración de la sesión de rol

La duración de la sesión del rol en segundos. Para obtener más información, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
RoleSessionDuration	Duración (obsoleto)	Opcional	3600

## Lake Formation habilitada

Utiliza la acción de la API de Lake Formation [AssumeDecoratedRoleWithSAML](#) para recuperar las credenciales de IAM temporales en lugar de la acción de la API de AWS STS [AssumeRoleWithSAML](#).

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
LakeFormationEnabled	Ninguno	Opcional	FALSE

## Credenciales de Ping

Un mecanismo de autenticación basado en SAML que permite la autenticación en Athena mediante el proveedor de identidad de Ping. Este método supone que ya se ha establecido una federación entre Athena y Ping.

## Proveedor de credenciales

El proveedor de credenciales que se utilizará para autenticar las solicitudes de AWS. Establezca el valor del parámetro en Ping.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado	Valor para usar
CredentialsProvider	AWSCredentialsProviderClass (obsoleto)	Obligatoria	Ninguno	Ping

## Usuario

La dirección de correo electrónico del usuario federado de Ping que se utilizará para la autenticación federada con Ping.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
Usuario	UID (obsoleto)	Obligatoria	Ninguno

## Password

La contraseña para el usuario federado de Ping.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
Password	PWD (obsoleto)	Obligatoria	Ninguno

## PingHostName

La dirección del servidor de Ping. Para encontrar la dirección, visite la siguiente URL y consulte el campo Punto de conexión de la aplicación de SSO.

```
https://your-pf-host-#:9999/pingfederate/your-pf-app#/spConnections
```

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
PingHostName	IdP_Host (obsoleto)	Obligatoria	Ninguno

## PingPortNumber

El número de puerto que se va a utilizar para conectarse al host del proveedor de identidad.



Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
PingPortNumber	IdP_Port (obsoleto)	Obligatoria	Ninguno

### PingPartnerSpId

La dirección del proveedor de servicios. Para encontrar la dirección del proveedor de servicios, visite la siguiente URL y consulte el campo Punto de conexión de la aplicación de SSO.

```
https://your-pf-host-#:9999/pingfederate/your-pf-app#/spConnections
```

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
PingPartnerSpId	Partner_SPID (obsoleto)	Obligatoria	Ninguno

### Rol preferido

El nombre de recurso de Amazon (ARN) del rol que se asignará. Para obtener información acerca de los roles de ARN, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
PreferredRole	preferred_role (obsoleto)	Opcional	Ninguno

### Duración de la sesión de rol

La duración de la sesión del rol en segundos. Para obtener más información, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
RoleSessionDuration	Duración (obsoleto)	Opcional	3600

## Lake Formation habilitada

Utiliza la acción de la API de Lake Formation [AssumeDecoratedRoleWithSAML](#) para recuperar las credenciales de IAM temporales en lugar de la acción de la API de AWS STS [AssumeRoleWithSAML](#).

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
LakeFormationEnabled	Ninguno	Opcional	FALSE

## Credenciales de AD FS

Un mecanismo de autenticación basado en SAML que permite la autenticación en Athena mediante los servicios de federación de Microsoft Active Directory (AD FS). Este método supone que el usuario ya ha establecido una federación entre Athena y AD FS.

### Proveedor de credenciales

El proveedor de credenciales que se utilizará para autenticar las solicitudes de AWS. Establezca el valor del parámetro en ADFS.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado	Valor para usar
CredentialsProvider	AWSCredentialsProviderClass (obsoleto)	Obligatoria	Ninguno	ADFS

## Usuario

La dirección de correo electrónico del usuario de AD FS que se utilizará para la autenticación con AD FS.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
Usuario	UID (obsoleto)	Necesario para la autenticación basada en formularios. Opcional para la autenticación integrada de Windows.	Ninguno

## Password

La contraseña del usuario de AD FS.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
Password	PWD (obsoleto)	Necesario para la autenticación basada en formularios. Opcional para la autenticación integrada de Windows.	Ninguno

## Nombre del host de AD FS

Dirección del servidor de AD FS.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
AdfsHostName	IdP_Host (obsoleto)	Obligatoria	Ninguno

### Número del puerto de AD FS

El número de puerto que se va a utilizar para conectarse al servidor de AD FS.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
AdfsPortNumber	IdP_Port (obsoleto)	Obligatoria	Ninguno

### Usuario de confianza de AD FS

El usuario de confianza. Utilice este parámetro para anular la URL del punto de conexión del usuario de confianza de AD FS.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
AdfsRelyingParty	LoginToRP (obsoleto)	Opcional	urn:amazon:webservices

### WIA de AD FS habilitado

Booleano. Use este parámetro para habilitar la autenticación integrada de Windows (WIA) con AD FS.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
AdfsWiaEnabled	none	Opcional	FALSE

## Rol preferido

El nombre de recurso de Amazon (ARN) del rol que se asignará. Para obtener información sobre los roles de ARN, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
PreferredRole	preferred_role (obsoleto)	Opcional	Ninguno

## Duración de la sesión de rol

La duración de la sesión del rol en segundos. Para obtener más información, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
RoleSessionDuration	Duración (obsoleto)	Opcional	3600

## Lake Formation habilitada

Especifica si se debe utilizar la acción de la API de Lake Formation [AssumeDecoratedRoleWithSAML](#) para recuperar las credenciales de IAM temporales en lugar de la acción de la API de AWS STS [AssumeRoleWithSAML](#).

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
LakeFormationEnabled	none	Opcional	FALSE

## Credenciales de Browser Azure AD

Browser Azure AD es un complemento de autenticación basado en SAML que funciona con el proveedor de identidad de Azure AD y admite la autenticación multifactor. A diferencia del

mecanismo de autenticación estándar de Azure AD, este mecanismo no requiere un nombre de usuario, contraseña o secreto de cliente en los parámetros de conexión. Al igual que el mecanismo de autenticación estándar de Azure AD, Browser Azure AD también asume que el usuario ya ha configurado la federación entre Athena y Azure AD.

### Proveedor de credenciales

El proveedor de credenciales que se utilizará para autenticar las solicitudes de AWS. Establezca el valor del parámetro en `BrowserAzureAD`.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado	Valor para usar
<code>CredentialsProvider</code>	<code>AWSCredentialsProviderClass</code> (obsoleto)	Obligatoria	Ninguno	<code>BrowserAzureAD</code>

### ID de inquilino de Azure AD

El ID de inquilino de su aplicación Azure AD

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
ID de inquilino de Azure	<code>tenant_id</code> (obsoleto)	Obligatoria	Ninguno

### ID de cliente de Azure AD

El ID de cliente de su aplicación Azure AD

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
<code>AzureAdClientId</code>	<code>client_id</code> (obsoleto)	Obligatoria	Ninguno

## Tiempo de respuesta del proveedor de identidad

El tiempo, en segundos, antes de que el controlador deje de esperar la respuesta de SAML de Azure AD.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
IdpResponseTimeout	idp_response_timeout (obsoleto)	Opcional	120

## Rol preferido

El nombre de recurso de Amazon (ARN) del rol que se asignará. Para obtener información acerca de los roles de ARN, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
PreferredRole	preferred_role (obsoleto)	Opcional	Ninguno

## Duración de la sesión de rol

La duración de la sesión del rol en segundos. Para obtener más información, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
RoleSessionDuration	Duración (obsoleto)	Opcional	3600

## Lake Formation habilitada

Utiliza la acción de la API de Lake Formation [AssumeDecoratedRoleWithSAML](#) para recuperar las credenciales de IAM temporales en lugar de la acción de la API de AWS STS [AssumeRoleWithSAML](#).

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
LakeFormationEnabled	Ninguno	Opcional	FALSE

### Credenciales del navegador SAML

El navegador SAML es un complemento de autenticación genérico que puede funcionar con proveedores de identidad basados en SAML y admitir la autenticación multifactor.

### Proveedor de credenciales

El proveedor de credenciales que se utilizará para autenticar las solicitudes de AWS. Establezca el valor del parámetro en `BrowserSaml`.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado	Valor para usar
CredentialsProvider	AWSCredentialsProviderClass (obsoleto)	Obligatoria	Ninguno	BrowserSaml

### URL de inicio de sesión único

La URL de inicio de sesión único para la aplicación en el proveedor de identidad basado en SAML.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
SsoLoginUrl	login_url (obsoleto)	Obligatoria	Ninguno



## Puerto de escucha

El número del puerto que se usa para escuchar la respuesta de SAML. Este valor debe coincidir con la URL con la que se configuró el proveedor de identidad basado en SAML (por ejemplo, `http://localhost:7890/athena`).

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
ListenPort	listen_port (obsoleto)	Opcional	7890

## Tiempo de respuesta del proveedor de identidad

El tiempo, en segundos, antes de que el controlador deje de esperar la respuesta de SAML de Azure AD.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
IdpResponseTimeout	idp_response_timeout (obsoleto)	Opcional	120

## Rol preferido

El nombre de recurso de Amazon (ARN) del rol que se asignará. Para obtener información acerca de los roles de ARN, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
PreferredRole	preferred_role (obsoleto)	Opcional	Ninguno

## Duración de la sesión de rol

La duración de la sesión del rol en segundos. Para obtener más información, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
RoleSessionDuration	Duración (obsoleto)	Opcional	3600

### Lake Formation habilitada

Utiliza la acción de la API de Lake Formation [AssumeDecoratedRoleWithSAML](#) para recuperar las credenciales de IAM temporales en lugar de la acción de la API de AWS STS [AssumeRoleWithSAML](#).

Nombre del parámetro	Alias	Tipo de parámetro	Valor predeterminado
LakeFormationEnabled	Ninguno	Opcional	FALSE

### Otra configuración de JDBC 3.x

En las siguientes secciones se describen algunos ajustes de configuración adicionales para el controlador JDBC 3.x.

#### Tiempo de espera de la red

Período de tiempo, en milisegundos, que el controlador va a esperar para que llegue la respuesta del controlador cuando realice una llamada de API a Athena. Transcurrido este tiempo, el controlador lanza una excepción de tiempo de espera.

El tiempo de espera de la red no se puede configurar como parámetro de conexión. Para configurarlo, llame al método `setNetworkTimeout` en un objeto `Connection` JDBC. Este valor se puede cambiar durante el ciclo de vida de la conexión JDBC. El valor predeterminado de este parámetro es `infinity`.

En el siguiente ejemplo, se establece el tiempo de espera de la red en 5000 milisegundos.

```
...
AthenaDriver driver = new AthenaDriver();
Connection connection = driver.connect(url, connectionParameters);
connection.setNetworkTimeout(null, 5000);
```

```
...
```

## Tiempo de espera de consulta

Período de tiempo, en segundos, que el controlador esperará a que se complete una consulta en Athena después de que se haya enviado. Transcurrido este tiempo, el controlador intenta cancelar la consulta enviada y lanza una excepción de tiempo de espera.

El tiempo de espera de la consulta no se puede configurar como parámetro de conexión. Para configurarlo, llame al método `setQueryTimeout` en un objeto `Statement` JDBC. Este valor se puede cambiar durante el ciclo de vida de la instrucción de JDBC. El valor predeterminado de este parámetro es 0 (cero). Un valor de 0 significa que las consultas se pueden ejecutar hasta que se completen (sujeto a [Service Quotas](#)).

El siguiente ejemplo establece el tiempo de espera de la consulta en 5 segundos.

```
...
AthenaDriver driver = new AthenaDriver();
Connection connection = driver.connect(url, connectionParameters);
Statement statement = connection.createStatement();
statement.setQueryTimeout(5);
...
```

## Notas de la versión de JDBC 3.x de Amazon Athena

Estas notas de lanzamiento proporcionan detalles sobre las mejoras y correcciones en el controlador JDBC 3.x de Amazon Athena.

### 3.2.0

Se lanzó el 26/04/2024

#### Mejoras

- Rendimiento del funcionamiento del catálogo: se ha mejorado el rendimiento de las operaciones de catálogo que no utilizan caracteres comodín.
- Cambio del intervalo mínimo de sondeo: el intervalo mínimo de sondeo predeterminado se ha modificado para reducir la cantidad de llamadas a la API que el controlador realiza a Athena. Las finalizaciones de consultas se siguen detectando lo antes posible.
- Capacidad de detección de las herramientas de inteligencia empresarial: se ha hecho que el controlador sea más fácil de detectar para las herramientas de inteligencia empresarial.

- Asignación de tipos de datos: se ha mejorado la asignación de tipos de datos a los tipos de datos DDL `binary`, `array` y `struct` de Athena.
- Versión del AWS SDK: la versión del AWS SDK que se utilizó en el controlador se actualizó a la versión 2.25.34.

## Correcciones

- Listados de tablas del catálogo federado: se corrigió un problema que causaba que los catálogos federados devolvieran una lista vacía de tablas.
- `getSchemas`: se corrigió un problema que hacía que el método JDBC [DatabaseMetaData#getSchemas](#) recuperara las bases de datos solo del catálogo predeterminado en lugar de todos los catálogos.
- `getColumn`: se corrigió un problema que provocaba que se devolviera un catálogo nulo cuando se llamaba al método JDBC [DatabaseMetaData#getColumn](#) con un nombre de catálogo nulo.

## 3.1.0

Publicado el 15/02/2024

## Mejoras

- Se añadió compatibilidad con la autenticación basada en formularios de autenticación integrada de Windows de Microsoft Active Directory Federation Services (AD FS).
- Para garantizar la compatibilidad con versiones anteriores de la versión 2.x, ahora se acepta el subprotocolo `awsathena` JDBC, pero genera una advertencia de obsolescencia. En su lugar, utilice el subprotocolo JDBC `athena`.
- `AwsDataCatalog` es ahora el valor predeterminado para el parámetro del catálogo y `default` es el valor predeterminado para el parámetro de la base de datos. Estos cambios garantizan que se devuelvan los valores correctos para el catálogo y la base de datos actuales en lugar de nulos.
- De conformidad con la especificación JDBC, `IS_AUTOINCREMENT` y `IS_GENERATEDCOLUMN` ahora devuelven una cadena vacía en lugar de `NO`.
- El tipo de datos `int` de Athena ahora se asigna al mismo tipo JDBC que el `integer` de Athena en lugar de `other`.
- Cuando los metadatos de las columnas de Athena no contienen los campos opcionales `precision` y `scale`, el controlador ahora devuelve cero para los valores correspondientes de una columna de `ResultSet`.

- La versión del AWS SDK se ha actualizado a la versión 2.21.39.

## Correcciones

- Se corrigió un problema con `GetQueryResultsStream` que provocaba que se produjera una excepción cuando los resultados de texto sin formato de Athena tenían un recuento de columnas inconsistente con el recuento de columnas en los metadatos de los resultados de Athena.

## 3.0.0

Se lanzó el 16/11/2023

El controlador Athena JDBC 3.x es un controlador de nueva generación que ofrece mejor rendimiento y compatibilidad. El controlador JDBC 3.x es compatible con la lectura de los resultados de las consultas directamente desde Amazon S3, lo que mejora el rendimiento de las aplicaciones que consumen resultados de consultas de gran tamaño. El nuevo controlador también tiene menos dependencias de terceros, lo que facilita la integración con las herramientas de inteligencia empresarial y las aplicaciones personalizadas.

## Versiones anteriores del controlador JDBC 3.x de Athena

Le recomendamos que utilice la [última versión](#) del controlador JDBC 3.x. La última versión del controlador contiene las mejoras y correcciones más recientes. Utilice una versión anterior solo si la aplicación presenta incompatibilidades con la versión más reciente.

## Controlador uber jar JDBC

La siguiente descarga empaqueta el controlador y todas sus dependencias en el mismo archivo `.jar`. Esta descarga se suele utilizar para clientes SQL de terceros.

- [3.1.0 uber jar](#)
- [3.0.0 uber jar](#)

## Controlador lean jar JDBC

La siguiente descarga es un archivo `.zip` que contiene la versión lean `.jar` del controlador y archivos `.jar` separados para las dependencias del controlador. Esta descarga se suele utilizar para aplicaciones personalizadas que pueden tener dependencias que entren en conflicto con las dependencias que utiliza el controlador. Esta descarga resulta útil si desea elegir qué dependencias

de controladores desea incluir en el `lean jar` y cuáles excluir si la aplicación personalizada ya contiene una o más de ellas.

- [3.1.0 lean jar](#)
- [3.0.0 lean jar](#)

## Controlador JDBC 2.x de Athena

Puede utilizar una conexión JDBC para conectar Athena con herramientas de inteligencia empresarial y otras aplicaciones, como [SQL Workbench](#). Para ello, utilice los enlaces de Amazon S3 de esta página para descargar, instalar y configurar el controlador JDBC 2.x de Athena. Para obtener información sobre la creación de la URL de conexión JDBC, consulte la [Guía de instalación y configuración del controlador JDBC](#). Para obtener información sobre los permisos, consulte [Acceso a través de conexiones JDBC y ODBC](#). Para enviar comentarios sobre el controlador JDBC, envíe un correo electrónico a [athena-feedback@amazon.com](mailto:athena-feedback@amazon.com). A partir de la versión 2.0.24, hay dos versiones del controlador disponibles: una que incluye el SDK de AWS, y una que no lo incluye.

### Important

Cuando utilice el controlador JDBC, asegúrese de tener en cuenta los siguientes requisitos:

- Abrir el puerto 444: mantenga el puerto 444, que Athena utiliza para transmitir los resultados de las consultas, abierto al tráfico saliente. Cuando utilice un punto de conexión de PrivateLink para conectarse a Athena, asegúrese de que el grupo de seguridad asociado al punto de conexión de PrivateLink esté abierto al tráfico entrante en el puerto 444. Si el puerto 444 está bloqueado, puede recibir el mensaje de error [Simba][AthenaJDBC](100123) An error has occurred. Exception during column initialization.
- Athena: política de flujo de resultados de consulta: agregue la acción de política `athena:GetQueryResultsStream` para las entidades principales de IAM que utilizan el controlador JDBC. Esta acción de política no se expone directamente con la API. Solo se utiliza con los controladores ODBC y JDBC como parte de la compatibilidad con los resultados de transmisión. Para ver una política de ejemplo, consulte [Política administrada de AWS: AWSQuicksightAthenaAccess](#).
- Uso del controlador JDBC para varios catálogos de datos: para usar el controlador JDBC para varios catálogos de datos con Athena (por ejemplo, cuando se usa un [metaalmacén externo de Hive](#) o [consultas federadas](#)), incluya `MetadataRetrievalMethod=ProxyAPI` en la cadena de conexión JDBC.

- Controladores 4.1: a partir de 2023, se interrumpirá la compatibilidad con los controladores para la versión 4.1 de JDBC. No se publicarán más actualizaciones. Si utiliza un controlador JDBC 4.1, se recomienda encarecidamente migrar al controlador 4.2.

## Controlador JDBC 2.x con AWS SDK

La versión 2.1.5 del controlador JDBC cumple con los estándares de datos de la versión 4.2 de la API de JDBC y requiere la versión 8.0 de JDK o posterior. Para obtener información sobre la verificación de la versión de Java Runtime Environment (JRE) que utiliza, consulte la [documentación](#) de Java.

Utilice el siguiente enlace para descargar el archivo `.jar` con la versión 4.2 del controlador JDBC.

- [AthenaJDBC42-2.1.5.1000.jar](#)

La siguiente descarga del archivo `.zip` contiene el archivo `.jar` para la versión 4.2 de JDBC e incluye AWS SDK y la documentación adjunta, las notas de la versión, las licencias y los acuerdos.

- [SimbaAthenaJDBC-2.1.5.1000.zip](#)

## Controlador JDBC 2.x sin AWS SDK

La versión 2.1.5 del controlador JDBC cumple con los estándares de datos de la versión 4.2 de la API de JDBC y requiere la versión 8.0 de JDK o posterior. Para obtener información sobre la verificación de la versión de Java Runtime Environment (JRE) que utiliza, consulte la [documentación](#) de Java.

Utilice el siguiente enlace para descargar el archivo `.jar` con la versión 4.2 del controlador JDBC sin AWS SDK.

- [AthenaJDBC42-2.1.5.1001.jar](#)

La siguiente descarga del archivo `.zip` contiene el archivo `.jar` para la versión 4.2 de JDBC, así como la documentación adjunta, las notas de la versión, las licencias y los acuerdos. No incluye el SDK de AWS.

- [SimbaAthenaJDBC-2.1.5.1001.zip](#)

## Notas de la versión, contrato de Licencia y avisos del controlador JDBC 2.x

Después de descargar la versión que necesita, lea las notas de la versión y examine el contrato de licencia y los avisos.

- [Notas de la versión](#)
- [Contrato de licencia](#)
- [Avisos](#)
- [Licencias de terceros](#)

## Documentación del controlador JDBC 2.x

Descargue la siguiente documentación del controlador:

- [Guía de configuración e instalación del controlador JDBC](#). Use esta guía para instalar y configurar el controlador.
- [Guía de migración del controlador JDBC](#). Use esta guía para migrar desde versiones anteriores a la versión actual.

## Conexión a Amazon Athena con ODBC

Amazon Athena ofrece dos controladores ODBC, las versiones 1.x y 2.x. El controlador ODBC 2.x de Athena es una nueva alternativa que es compatible con los sistemas Windows de 64 bits, Linux, macOS ARM y macOS Intel. El controlador 2.x de Athena es compatible con todos los complementos de autenticación que admite el controlador ODBC 1.x y casi todos los parámetros de conexión son compatibles con versiones anteriores.

- Para descargar el controlador ODBC 2.x, consulte [ODBC 2.x de Amazon Athena](#).
- Para descargar el controlador ODBC 1.x, consulte [Controlador ODBC 1.x de Athena](#).

## Temas

- [ODBC 2.x de Amazon Athena](#)
- [Controlador ODBC 1.x de Athena](#)
- [Uso del conector Power BI de Amazon Athena](#)



## ODBC 2.x de Amazon Athena

Puede utilizar una conexión ODBC para conectarse a Amazon Athena desde distintas aplicaciones y herramientas de cliente SQL de terceros. La conexión ODBC se configura en el equipo cliente.

### Consideraciones y limitaciones

- Para obtener información sobre la migración del controlador ODBC 1.x de Athena al controlador ODBC 2.x de Athena, consulte [Migración al controlador ODBC 2.x](#).
- Al utilizar la [recuperación de S3](#) con la [opción de cifrado](#) CSE\_KMS, el cliente de Amazon S3 no puede descifrar el resultado almacenado en el bucket de Amazon S3. Como solución alternativa, utilice la opción de la [API de transmisión de Athena](#) para recuperar el conjunto de resultados.

### Descarga del controlador ODBC 2.x

Para descargar el controlador ODBC 2.x de Amazon Athena, visite los enlaces de esta página.

#### Important

Cuando utilice el controlador ODBC 2.x, asegúrese de tener en cuenta los siguientes requisitos:

- Abrir el puerto 444: mantenga el puerto 444, que Athena utiliza para transmitir los resultados de las consultas, abierto al tráfico saliente. Cuando utilice un punto de conexión de PrivateLink para conectarse a Athena, asegúrese de que el grupo de seguridad asociado al punto de conexión de PrivateLink esté abierto al tráfico entrante en el puerto 444.
- Athena: política de flujo de resultados de consulta: agregue la acción de política `athena:GetQueryResultsStream` para las entidades principales de IAM que utilizan el controlador ODBC. Esta acción de política no se expone directamente con la API. Solo se utiliza con los controladores ODBC y JDBC como parte de la compatibilidad con los resultados de transmisión. Para ver una política de ejemplo, consulte [Política administrada de AWS: AWSQuicksightAthenaAccess](#).

## Linux

Versión de controlador	Enlace de descarga
ODBC 2.0.3.0 para Linux de 64 bits	<a href="#">Controlador ODBC 2.0.3.0 para Linux de 64 bits</a>

## macOS (ARM)

Versión de controlador	Enlace de descarga
ODBC 2.0.3.0 para macOS de 64 bits (ARM)	<a href="#">Controlador ODBC 2.0.3.0 para macOS de 64 bits (ARM)</a>

## macOS (Intel)

Versión de controlador	Enlace de descarga
ODBC 2.0.3.0 para macOS de 64 bits (Intel)	<a href="#">Controlador ODBC 2.0.3.0 para macOS de 64 bits (Intel)</a>

## Windows

Versión de controlador	Enlace de descarga
ODBC 2.0.3.0 para Windows de 64 bits	<a href="#">Controlador ODBC 2.0.3.0 para Windows de 64 bits</a>

## Temas

- [Introducción al controlador ODBC 2.x](#)
- [Parámetros de conexión ODBC 2.x de Athena](#)
- [Migración al controlador ODBC 2.x](#)
- [Solución de problemas del controlador ODBC 2.x](#)

- [Notas de la versión de ODBC 2.x de Amazon Athena](#)

## Introducción al controlador ODBC 2.x

Utilice la información de esta sección para empezar a usar el controlador ODBC 2.x de Amazon Athena. El controlador es compatible con los sistemas operativos de Windows, Linux y macOS.

### Temas

- [Windows](#)
- [Linux](#)
- [macOS](#)

## Windows

Si desea utilizar un equipo cliente con Windows para acceder a Amazon Athena, necesitará el controlador ODBC de Amazon Athena.

### Requisitos del sistema Windows

Instale el controlador ODBC de Amazon Athena en los equipos cliente que accederán directamente a las bases de datos de Amazon Athena en lugar de utilizar un navegador web.

El sistema Windows utilizado debe cumplir con los siguientes requisitos:

- Tener derechos de administrador disponibles.
- Uno de los sistemas operativos siguientes:
  - Windows 11, 10 u 8.1
  - Windows Server 2019, 2016 o 2012
- Contar con al menos 100 MB de espacio disponible en el disco.
- Tener [Microsoft Visual C++ Redistributable for Visual Studio](#) para Windows de 64 bits instalado.

## Instalación del controlador ODBC de Amazon Athena

Para descargar e instalar el controlador ODBC de Amazon Athena para Windows

1. [Descargue](#) el archivo de instalación AmazonAthenaODBC-2.x.x.x.msi.

2. Inicie el archivo de instalación y, a continuación, seleccione Siguiente.
3. Para aceptar los términos del contrato de licencia, seleccione la casilla de verificación y, a continuación, elija Siguiente.
4. Para cambiar la ubicación de instalación, seleccione Examinar, busque la carpeta deseada y luego elija Aceptar.
5. Para aceptar la ubicación de instalación, seleccione Siguiente.
6. Elija Instalar.
7. Cuando finalice la instalación, elija Finalizar.

### Formas de configurar las opciones de ajuste del controlador

Para controlar el comportamiento del controlador ODBC de Amazon Athena en Windows, puede configurar las opciones de ajuste del controlador de las siguientes maneras:

- En el programa Administrador de orígenes de datos ODBC, al configurar un nombre de origen de datos (DSN).
- Al agregar o cambiar las claves de registro de Windows en la siguiente ubicación:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\YOUR_DSN_NAME
```

- Al configurar las opciones del controlador en la cadena de conexión cuando se conecta mediante programación.


### Configuración del nombre de origen de datos en Windows

Después de descargar e instalar el controlador ODBC, debe agregar una entrada de nombre de origen de datos (DSN) al equipo cliente o a la instancia de Amazon EC2. Las herramientas de cliente SQL utilizan este origen de datos para conectarse a Amazon Athena y hacer consultas.

### Pasos para crear una entrada de sistema DSN

1. En el menú Inicio de Windows, haga clic con el botón derecho en Orígenes de datos ODBC (64 bits) y, a continuación, seleccione Más, Ejecutar como administrador.
2. En el Administrador de orígenes de datos ODBC, elija la pestaña Controladores.
3. En la columna Nombre, compruebe que Amazon Athena ODBC (x64) esté presente.
4. Realice una de las siguientes acciones siguientes:

- A fin de configurar el controlador para todos los usuarios del equipo, seleccione la pestaña DSN del sistema. Como es posible que las aplicaciones que utilizan una cuenta diferente para cargar datos no puedan detectar los DSN de usuario de otra cuenta, se recomienda la opción de configuración de DSN del sistema.

 Note

El uso de la opción DSN del sistema requiere privilegios administrativos.

- A fin de configurar el controlador únicamente para su cuenta de usuario, seleccione la pestaña DSN de usuario.
5. Elija Add (Agregar). Se abrirá el cuadro de diálogo Crear nuevo origen de datos.
  6. Elija Amazon Athena ODBC (x64) y, a continuación, elija Finalizar.
  7. En el cuadro de diálogo Configuración de ODBC de Amazon Athena, introduzca la siguiente información. Para obtener información detallada sobre estas opciones, consulte [Parámetros de conexión ODBC 2.x principales](#).
    - En Nombre de origen de datos, introduzca un nombre que desee utilizar para identificar el origen de datos.
    - En Descripción, escriba una descripción que lo ayude a identificar el origen de datos.
    - En Región, escriba el nombre de la Región de AWS en la que usará Athena (por ejemplo, **us-west-1**).
    - En Catálogo, escriba el nombre del catálogo de Amazon Athena. El valor predeterminado es AwsDataCatalog, que es el que AWS Glue utiliza.
    - En Base de datos, introduzca el nombre de la base de datos de Amazon Athena. El valor predeterminado es predeterminada.
    - En Grupo de trabajo, introduzca el nombre del grupo de trabajo de Amazon Athena. El valor predeterminado es primario.
    - En la ubicación de salida de S3, ingrese la ubicación en Amazon S3 donde se almacenarán los resultados de la consulta (por ejemplo, **s3://DOC-EXAMPLE-BUCKET/**).
    - (Opcional) En Opciones de cifrado, elija una opción de cifrado. El valor predeterminado es NOT\_SET.
    - (Opcional) En Clave de KMS, elija una clave de KMS de cifrado si es necesaria.

8. A fin de especificar las opciones de configuración para la autenticación de IAM, elija Opciones de autenticación.
9. Introduzca la información siguiente:
  - En Tipo de autenticación, elija Credenciales de IAM. Esta es la opción predeterminada. Para obtener más información acerca de los tipos de autenticación disponibles, consulte [Opciones de autenticación](#).
  - En Nombre de usuario, introduzca un nombre de usuario.
  - En Contraseña, introduzca una contraseña.
  - En Token de sesión, introduzca un token de sesión si desea usar credenciales de AWS temporales. Para obtener información acerca de las credenciales temporales, consulte [Uso de credenciales temporales con recursos de AWS](#) en la Guía del usuario de IAM.
10. Seleccione Aceptar.
11. En la parte inferior del cuadro de diálogo Configuración de ODBC de Amazon Athena, elija Prueba. Si el equipo cliente se conecta correctamente a Amazon Athena, el cuadro Prueba de conexión indica Conexión correcta. De lo contrario, el cuadro indica Error de conexión con la información de error correspondiente.
12. Para cerrar la prueba de conexión, elija Aceptar. El origen de datos que ha creado ahora aparece en la lista de nombres de orígenes de datos.

### Uso de una conexión sin DSN en Windows

Puede usar una conexión sin DSN para conectarse a una base de datos sin un nombre de origen de datos (DSN). En el siguiente ejemplo, se muestra una cadena de conexión para el controlador ODBC (x64) de Amazon Athena que se conecta a Amazon Athena.

```
DRIVER={Amazon Athena ODBC (x64)};Catalog=AwsDataCatalog;AwsRegion=us-west-1;Schema=test_schema;S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/;AuthenticationType=IAM Credentials;UID=YOUR_UID;PWD=YOUR_PWD;
```

### Linux

Si desea utilizar un equipo cliente con Linux para acceder a Amazon Athena, necesitará el controlador ODBC de Amazon Athena.

## Requisitos del sistema Linux

Cada equipo cliente con Linux en el que instala el controlador debe cumplir con los siguientes requisitos:

- Tiene acceso raíz.
- Utilice una de las siguientes distribuciones de Linux:
  - Red Hat Enterprise Linux (RHEL) 7 u 8
  - CentOS 7 u 8.
- Tenga 100 MB de espacio disponible en el disco.
- Utilice la versión 2.3.1 o posterior de [unixODBC](#).
- Utilice la versión 2.26 o posterior de la [Biblioteca C de GNU](#) (glibc).

## Instalación del conector de datos ODBC en Linux

Utilice el siguiente procedimiento para instalar el controlador ODBC de Amazon Athena en un sistema operativo Linux.

Para instalar del controlador ODBC de Amazon Athena en Linux

1. Especifique uno de los siguientes comandos:

```
sudo rpm -Uvh AmazonAthenaODBC-2.X.Y.Z.rpm
```

o

```
sudo yum --nogpgcheck localinstall AmazonAthenaODBC-2.X.Y.Z.rpm
```

2. Una vez finalizada la instalación, introduzca uno de los siguientes comandos para comprobar que el controlador esté instalado:

- ```
yum list | grep amazon-athena-odbc-driver
```

Salida:

```
amazon-athena-odbc-driver.x86_64 2.0.2.1-1.amzn2int installed
```

- ```
rpm -qa | grep amazon
```

Salida:

```
amazon-athena-odbc-driver-2.0.2.1-1.amzn2int.x86_64
```

## Configuración del nombre de origen de datos en Linux

Una vez instalado el controlador, encontrará los archivos de ejemplo `.odbc.ini` y `.odbcinst.ini` en la siguiente ubicación:

- `/opt/athena/odbc/ini/`.

Utilice los archivos `.ini` en esta ubicación como ejemplos para configurar el controlador ODBC de Amazon Athena y el nombre de origen de datos (DSN).

### Note

De manera predeterminada, los administradores de los controladores ODBC utilizan la versión oculta de los archivos de configuración `.odbc.ini` y `.odbcinst.ini`, que están ubicados en el directorio de inicio.

Para especificar la ruta a los archivos `.odbc.ini` y `.odbcinst.ini` mediante `unixODBC`, realice los siguientes pasos.

Para especificar las ubicaciones de los archivos **.ini** ODBC mediante `unixODBC`

1. Configure `ODBCINI` en la ruta completa y el nombre del archivo `odbc.ini`, como en el siguiente ejemplo:

```
export ODBCINI=/opt/athena/odbc/ini/odbc.ini
```

2. Configure `ODBCSYSINI` en la ruta completa del directorio que contiene el archivo `odbcinst.ini`, como en el siguiente ejemplo:

```
export ODBCSYSINI=/opt/athena/odbc/ini
```



3. Introduzca el siguiente comando para comprobar que está utilizando el administrador de controladores unixODBC y los archivos `odbc*.ini` correctos:

```
username % odbcinst -j
```

### Resultado de ejemplo

```
unixODBC 2.3.1
DRIVERS.....: /opt/athena/odbc/ini/odbcinst.ini
SYSTEM DATA SOURCES: /opt/athena/odbc/ini/odbc.ini
FILE DATA SOURCES..: /opt/athena/odbc/ini/ODBCDataSources
USER DATA SOURCES..: /opt/athena/odbc/ini/odbc.ini
SQLULEN Size.....: 8
SQLLEN Size.....: 8
SQLSETPOSIR0W Size.: 8
```

4. Si desea utilizar un nombre de origen de datos (DSN) para conectar su almacenamiento de datos, configure el archivo `odbc.ini` para definir los nombres de origen de datos (DSN). Establezca las propiedades del archivo `odbc.ini` para crear un DSN que especifique la información de conexión para el almacén de datos, como se observa en el siguiente ejemplo.

```
[ODBC Data Sources]
athena_odbc_test=Amazon Athena ODBC (x64)

[ATHENA_WIDE_SETTINGS] # Special DSN-name to signal driver about logging
configuration.
LogLevel=0             # To enable ODBC driver logs, set this to 1.
UseAwsLogger=0         # To enable AWS-SDK logs, set this to 1.
LogPath=/opt/athena/odbc/logs/ # Path to store the log files. Permissions to the
location are required.

[athena_odbc_test]
Driver=/opt/athena/odbc/lib/libathena-odbc.so
AwsRegion=us-west-1
Workgroup=primary
Catalog=AwsDataCatalog
Schema=default
AuthenticationType=IAM Credentials
UID=
PWD=
S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/
```

5. Configure el archivo `odbcinst.ini`, como en el siguiente ejemplo:

```
[ODBC Drivers]
Amazon Athena ODBC (x64)=Installed

[Amazon Athena ODBC (x64)]
Driver=/opt/athena/odbc/lib/libathena-odbc.so
Setup=/opt/athena/odbc/lib/libathena-odbc.so
```

6. Después de instalar y configurar el controlador ODBC de Amazon Athena, utilice la herramienta de línea de comandos `isql` de unixODBC para comprobar la conexión, como se muestra en el siguiente ejemplo.

```
username % isql -v "athena_odbc_test"
+-----+
| Connected! |
|           |
| sql-statement |
| help [tablename] |
| quit |
|           |
+-----+
SQL>
```

## macOS

Si desea utilizar un equipo cliente con macOS para acceder a Amazon Athena, necesitará el controlador ODBC de Amazon Athena.

### Requisitos del sistema macOS

Cada equipo con macOS en el que instala el controlador debe cumplir con los siguientes requisitos.

- Versión 14 de macOS o posterior.
- 100 MB de espacio disponible en el disco.
- Versión 3.52.16 o posterior de [iODBC](#).

## Instalación del conector de datos ODBC en macOS

Utilice el siguiente procedimiento para descargar e instalar el controlador ODBC de Amazon Athena para los sistemas operativos macOS.

### Cómo descargar e instalar el controlador ODBC de Amazon Athena para macOS

1. Descargue el archivo de paquete .pkg.
2. Haga doble clic en el archivo .pkg.
3. Siga las instrucciones del asistente para instalar el controlador.
4. En la página del Contrato de licencia, pulse Continuar y, a continuación, seleccione Acepto.
5. Elija Instalar.
6. Cuando finalice la instalación, elija Finalizar.
7. Ejecute el siguiente comando para comprobar que se haya instalado el controlador:

```
> pkgutil --pkgs | grep athenaodbc
```

En función del sistema, es posible que el resultado tenga un aspecto similar al siguiente.

```
com.amazon.athenaodbc-x86_64.Config  
com.amazon.athenaodbc-x86_64.Driver
```

o


```
com.amazon.athenaodbc-arm64.Config  
com.amazon.athenaodbc-arm64.Driver
```

## Configuración del nombre de origen de datos en macOS

Una vez instalado el controlador, encontrará los archivos de ejemplo .odbc.ini y .odbcinst.ini en las siguientes ubicaciones:

- Equipos con procesador Intel: /opt/athena/odbc/x86\_64/ini/
- Equipos con procesador ARM: /opt/athena/odbc/arm64/ini/

Utilice los archivos `.ini` de esta ubicación como ejemplos para configurar el controlador ODBC de Amazon Athena y el nombre de origen de datos (DSN).

 Note

De manera predeterminada, los administradores de los controladores ODBC utilizan la versión oculta de los archivos de configuración `.odbc.ini` y `.odbcinst.ini`, que están ubicados en el directorio de inicio.

Para especificar la ruta a los archivos `.odbc.ini` y `.odbcinst.ini` mediante el administrador de controladores iODBC, realice los siguientes pasos.

Cómo especificar las ubicaciones de los archivos `.ini` de OBCD mediante el administrador de controladores iODBC

1. Configure `ODBCINI` en la ruta completa y el nombre del archivo `odbc.ini`.

- Para equipos con macOS que tienen procesadores Intel, utilice la siguiente sintaxis.

```
export ODBCINI=/opt/athena/odbc/x86_64/ini/odbc.ini
```

- Para equipos con macOS que tienen procesadores ARM, utilice la siguiente sintaxis.

```
export ODBCINI=/opt/athena/odbc/arm64/ini/odbc.ini
```

2. Configure `ODBCSYSINI` en la ruta completa y el nombre del archivo `odbcinst.ini`.

- Para equipos con macOS que tienen procesadores Intel, utilice la siguiente sintaxis.

```
export ODBCSYSINI=/opt/athena/odbc/x86_64/ini/odbcinst.ini
```

- Para equipos con macOS que tienen procesadores ARM, utilice la siguiente sintaxis.

```
export ODBCSYSINI=/opt/athena/odbc/arm64/ini/odbcinst.ini
```

3. Si desea utilizar un nombre de origen de datos (DSN) para conectar su almacenamiento de datos, configure el archivo `odbc.ini` para definir los nombres de origen de datos (DSN). Establezca las propiedades del archivo `odbc.ini` para crear un DSN que especifique la información de conexión para el almacén de datos, como se observa en el siguiente ejemplo.

```
[ODBC Data Sources]
athena_odbc_test=Amazon Athena ODBC (x64)

[ATHENA_WIDE_SETTINGS] # Special DSN-name to signal driver about logging
configuration.
LogLevel=0             # set to 1 to enable ODBC driver logs
UseAwsLogger=0        # set to 1 to enable AWS-SDK logs
LogPath=/opt/athena/odbc/logs/ # Path to store the log files. Permissions to the
location are required.

[athena_odbc_test]
Description=Amazon Athena ODBC (x64)
# For ARM:
Driver=/opt/athena/odbc/arm64/lib/libathena-odbc-arm64.dylib
# For Intel:
# Driver=/opt/athena/odbc/x86_64/lib/libathena-odbc-x86_64.dylib
AwsRegion=us-west-1
Workgroup=primary
Catalog=AwsDataCatalog
Schema=default
AuthenticationType=IAM Credentials
UID=
PWD=
S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/
```

4. Configure el archivo `odbcinst.ini`, como en el siguiente ejemplo.

```
[ODBC Drivers]
Amazon Athena ODBC (x64)=Installed

[Amazon Athena ODBC (x64)]
# For ARM:
Driver=/opt/athena/odbc/arm64/lib/libathena-odbc-arm64.dylib
Setup=/opt/athena/odbc/arm64/lib/libathena-odbc-arm64.dylib
# For Intel:
# Driver=/opt/athena/odbc/x86_64/lib/libathena-odbc-x86_64.dylib
# Setup=/opt/athena/odbc/x86_64/lib/libathena-odbc-x86_64.dylib
```

5. Después de instalar y configurar el controlador ODBC de Amazon Athena, utilice la herramienta de línea de comandos `iodbctest` para comprobar la conexión, como se muestra en el siguiente ejemplo.

```
username@ % iodbctest
iODBC Demonstration program
This program shows an interactive SQL processor
Driver Manager: 03.52.1623.0502

Enter ODBC connect string (? shows list): ?

DSN                                | Driver
-----|-----
athena_odbc_test                    | Amazon Athena ODBC (x64)

Enter ODBC connect string (? shows list): DSN=athena_odbc_test;
Driver: 2.0.2.1 (Amazon Athena ODBC Driver)

SQL>
```

## Parámetros de conexión ODBC 2.x de Athena

Las opciones del cuadro de diálogo Configuración de ODBC de Amazon Athena incluyen Opciones de autenticación, Opciones avanzadas, Opciones de registro, Anulaciones de puntos de conexión y Opciones de proxy. Para obtener más información sobre cada una, visite los enlaces correspondientes.

- [Parámetros de conexión ODBC 2.x principales](#)
- [Opciones de autenticación](#)
- [Opciones avanzadas](#)
- [Opciones de registro](#)
- [Anulaciones de puntos de conexión](#)
- [Opciones de proxy](#)

## Parámetros de conexión ODBC 2.x principales

En las siguientes secciones se describe cada uno de los parámetros de conexión principales.

### Nombre de origen de datos

Especifica el nombre del origen de datos.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
DSN	Opcional para los tipos de conexión sin DSN	none	DSN=AmazonAthena0dbcUsWest1;

### Descripción

Contiene una descripción del origen de datos.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
Descripción	Opcional	none	Description=Connection to Amazon Athena us-west-1;

### Catálogo

Especifica el nombre del catálogo de datos. Para obtener más información sobre los catálogos, consulte [DataCatalog](#) en la Referencia de la API de Amazon Athena.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
Catálogo	Opcional	AwsDataCatalog	Catalog=AwsDataCatalog;

### Región

Especifica el Región de AWS. Para obtener información sobre las Regiones de AWS, consulte [Regiones y zonas de disponibilidad](#).

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
AwsRegion	Obligatorio	none	AwsRegion =us-west-1;

## Base de datos

Especifica el nombre de la base de datos. Para obtener más información sobre las bases de datos, consulte [Bases de datos](#) en la Referencia de la API de Amazon Athena.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
Esquema	Opcional	default	Schema=default;

## Grupo de trabajo

Especifica el nombre del grupo de trabajo. Para obtener más información sobre los grupos de trabajo, consulte [WorkGroup](#) en la Referencia de la API de Amazon Athena.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
Grupo de trabajo	Opcional	primary	Workgroup =primary;

## Ubicación de salida

Especifica la ubicación en Amazon S3 en la que se almacenan los resultados de las consultas. Para obtener más información sobre la ubicación de salida, consulte [ResultConfiguration](#) en la Referencia de la API de Amazon Athena.



Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
S3OutputLocation	Obligatorio	none	S3outputLocation=s3://DOC-EXAMPLE-BUCKET/;

## Opciones de cifrado

Nombre del parámetro de diálogo: opciones de cifrado

Especifica la opción de cifrado. Para obtener más información sobre las opciones de cifrado, consulte [EncryptionConfiguration](#) en la Referencia de la API de Amazon Athena.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Valores posibles	Ejemplo de la cadena de conexión
S3OutputEncryptionOption	Opcional	none	NOT_SET, SSE_S3, SSE_KMS, CSE_KMS	S3outputEncryptionOption=SSE_S3;

## Clave de KMS

Especifica una clave de KMS para el cifrado. Para obtener más información sobre la configuración de cifrado de las claves de KMS, consulte [EncryptionConfiguration](#) en la Referencia de la API de Amazon Athena.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
S3OutputEncKMSKey	Opcional	none	S3outputEncKMSKey=your_key;

## Prueba de conexión

El administrador de orígenes de datos ODBC ofrece la opción Prueba que puede utilizar para probar la conexión de ODBC 2.x a Amazon Athena. Para ver los pasos, consulte [Configuración del nombre de origen de datos en Windows](#). Al probar una conexión, el controlador ODBC llama a la acción de la API de Athena [GetWorkGroup](#). La llamada utiliza el tipo de autenticación y el proveedor de credenciales correspondiente que especificó para recuperar las credenciales. La prueba de conexión es gratuita cuando se utiliza el controlador ODBC 2.x. La prueba no genera resultados de consultas en el bucket de Amazon S3.

## Opciones de autenticación

Puede conectarse a Amazon Athena mediante los siguientes tipos de autenticación. Para todos los tipos, el nombre de la cadena de conexión es `AuthenticationType`, el tipo de parámetro es `Required` y el valor predeterminado es `IAM Credentials`. Para obtener información sobre los parámetros de cada tipo de autenticación, visite el enlace correspondiente. Para ver los parámetros de autenticación comunes, consulte [Parámetros de autenticación comunes](#).

Tipo de autenticación	Ejemplo de la cadena de conexión
<a href="#">Credenciales de IAM</a>	<code>AuthenticationType=IAM Credentials;</code>
<a href="#">Perfil de IAM</a>	<code>AuthenticationType=IAM Profile;</code>
<a href="#">AD FS</a>	<code>AuthenticationType=ADFS;</code>
<a href="#">Azure AD</a>	<code>AuthenticationType=AzureAD;</code>
<a href="#">Browser Azure AD</a>	<code>AuthenticationType=BrowserAzureAD;</code>
<a href="#">Browser SAML</a>	<code>AuthenticationType=BrowserSAML;</code>
<a href="#">Browser SSO OIDC</a>	<code>AuthenticationType=BrowserSSOOIDC;</code>
<a href="#">Credenciales predeterminadas</a>	<code>AuthenticationType=Default Credentials;</code>
<a href="#">Credenciales externas</a>	<code>AuthenticationType=External Credentials;</code>

Tipo de autenticación	Ejemplo de la cadena de conexión
<a href="#">Perfil de instancia</a>	<code>AuthenticationType=Instance Profile;</code>
<a href="#">JWT</a>	<code>AuthenticationType=JWT;</code>
<a href="#">Okta</a>	<code>AuthenticationType=Okta;</code>
<a href="#">Ping</a>	<code>AuthenticationType=Ping;</code>

## Credenciales de IAM

Puede usar sus credenciales de IAM para conectarse a Amazon Athena con el controlador ODBC mediante los parámetros de cadena de conexión que se describen en esta sección.

### Tipo de autenticación

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
AuthenticationType	Obligatoria	IAM Credentials	<code>AuthenticationType=IAM Credentials;</code>

### ID de usuario

El ID de su clave de acceso de AWS. Para obtener más información sobre las claves de acceso, consulte [Credenciales de seguridad de AWS](#) en la Guía del usuario de IAM

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
UID	Obligatoria	none	<code>UID=AKIAIOSFODNN7EXAMPLE;</code>

## Password

El ID de su clave secreta de AWS. Para obtener más información sobre las claves de acceso, consulte [Credenciales de seguridad de AWS](#) en la Guía del usuario de IAM

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
PWD	Obligatoria	none	PWD=wJa1r XUtnFEMI/ K7MDENG/b PxRfiCYEX AMPLEKE;

## Token de sesión

Si utiliza credenciales de AWS temporales, debe especificar el token de sesión. Para obtener información acerca de las credenciales temporales, consulte [Credenciales de seguridad temporales en IAM](#) en la Guía del usuario de IAM.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
SessionToken	Opcional	none	SessionTo ken=AQoDY XdzEJr... <remainder of session token>;

## Perfil de IAM

Puede configurar un perfil con nombre para conectarse a Amazon Athena mediante el controlador ODBC. Para utilizar las credenciales disponibles en el perfil de instancia de Amazon EC2 de alojamiento, defina el parámetro `credential_source` en `Ec2InstanceMetadata`. Si desea utilizar un proveedor de credenciales personalizado en un perfil con nombre, especifique un valor para el parámetro `plugin_name` en la configuración de su perfil.

## Tipo de autenticación

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
AuthenticationType	Obligatoria	IAM Credentials	AuthenticationType=IAM Profile;

## Perfil de AWS

El nombre del perfil que se va a utilizar para la conexión ODBC. Para obtener más información sobre los perfiles, consulte [Uso de perfiles con nombre](#) en la Guía del usuario de AWS Command Line Interface.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
Perfil de AWS	Obligatoria	none	AWSProfile=default;

## Rol preferido

El nombre de recurso de Amazon (ARN) del rol que se asignará. El parámetro de rol preferido se usa cuando el parámetro `plugin_name` especifica el proveedor de credenciales personalizado en la configuración del perfil. Para obtener más información sobre los roles de ARN, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
preferred_role	Opcional	none	preferred_role=arn: aws:IAM:

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
			:12345678 9012:id/user1;

### Duración de la sesión

La duración de la sesión del rol en segundos. Para obtener más información sobre la duración de la sesión, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service. El parámetro de duración de la sesión se usa cuando el parámetro `plugin_name` especifica el proveedor de credenciales personalizado en la configuración del perfil.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
duration	Opcional	900	duration=900;

### Nombre del complemento

Especifica el nombre de un proveedor de credenciales personalizado utilizado en un perfil con nombre. Este parámetro puede tomar los mismos valores que los del campo Tipo de autenticación del administrador de orígenes de datos ODBC, pero solo se usa en la configuración `AWSPProfile`.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
plugin_name	Opcional	none	plugin_name=AzureAD;

### AD FS

AD FS es un complemento de autenticación basado en SAML que funciona con el proveedor de identidad de Active Directory Federation Service (AD FS). El complemento admite la [autenticación integrada de Windows](#) y la autenticación basada en formularios. Si utiliza la autenticación integrada de Windows, puede omitir el nombre de usuario y la contraseña. Para obtener información sobre la

configuración de AD FS y Athena, consulte [Configuración del acceso federado a Amazon Athena para los usuarios de Microsoft AD FS mediante un cliente ODBC](#).

### Tipo de autenticación

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
AuthenticationType	Obligatoria	IAM Credentials	AuthenticationType=ADFS;

### ID de usuario

Su nombre de usuario para conectarse al servidor AD FS. Para la autenticación integrada de Windows, puede omitir el nombre de usuario. Si la configuración de AD FS requiere un nombre de usuario, debe proporcionarlo en el parámetro de conexión.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
UID	Opcional para la autenticación integrada de Windows	none	UID=domain\username;

### Password

Su contraseña para conectarse al servidor AD FS. Al igual que el campo de nombre de usuario, puede omitir la contraseña si utiliza la autenticación integrada de Windows. Si la configuración de AD FS requiere una contraseña, debe proporcionarla en el parámetro de conexión.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
PWD	Opcional para la autenticación integrada de Windows	none	PWD=password_3EXAMPLE;

## Rol preferido

El nombre de recurso de Amazon (ARN) del rol que se asignará. Si su aserción de SAML tiene varios roles, puede especificar este parámetro para elegir el rol que se va a asumir. Este rol debe estar presente en la aserción de SAML. Para obtener más información sobre los roles de ARN, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
preferred_role	Opcional	none	preferred_role=arn:aws:IAM:123456789012:id/user1;

## Duración de la sesión

La duración de la sesión del rol en segundos. Para obtener más información sobre la duración de la sesión, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
duration	Opcional	900	duration=900;

## Host del IdP

El nombre del host del servicio AD FS.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
idp_host	Obligatorio	none	idp_host=<server-name>.<company.com>;



## Puerto del IdP

El puerto que se va a usar para conectarse al host de AD FS.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
idp_port	Obligatoria	none	idp_port=443;

## LoginToRP

El usuario de confianza. Utilice este parámetro para anular la URL del punto de conexión del usuario de confianza de AD FS.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
LoginToRP	Opcional	urn:amazon:webservices	LoginToRP=trustedparty;

## Azure AD

Azure AD es un complemento de autenticación basado en SAML que funciona con el proveedor de identidad de Azure AD. Este complemento no admite la autenticación multifactor (MFA). Si necesita compatibilidad con MFA, considere usar el complemento BrowserAzureAD en su lugar.

## Tipo de autenticación

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
AuthenticationType	Obligatoria	IAM Credentials	AuthenticationType=AzureAD;

## Rol preferido

El nombre de recurso de Amazon (ARN) del rol que se asignará. Para obtener información acerca de los roles de ARN, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
preferred_role	Opcional	none	preferred_role=arn:aws:iam: :123456789012:id/user1;

## Duración de la sesión

La duración de la sesión del rol en segundos. Para obtener más información, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
duration	Opcional	900	duration=900;

## ID de inquilino

Especifica el ID de inquilino de la aplicación.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
idp_tenant	Obligatoria	none	idp_tenant=123zz112z-z12d-1 z1f-11zz-f111aa111234;

## ID de cliente

Especifica el ID de cliente de la aplicación.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
client_id	Obligatoria	none	client_id=9178ac27-a1bc-1a2b-1a2b-a123abcd1234;

## Secreto del cliente

Especifica el secreto de cliente.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
client_secret	Obligatoria	none	client_secret=zG12q~.xzG1xxZ1wX1.~ZzXXX1XxkHZizeT1zzZ;

## Browser Azure AD

Browser Azure AD es un complemento de autenticación basado en SAML que funciona con el proveedor de identidad de Azure AD y admite la autenticación multifactor. A diferencia del complemento estándar de Azure AD, este complemento no requiere un nombre de usuario, contraseña o secreto de cliente en los parámetros de conexión.

## Tipo de autenticación

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
AuthenticationType	Obligatoria	IAM Credentials	AuthenticationType=BrowserAzureAD;

## Rol preferido

El nombre de recurso de Amazon (ARN) del rol que se asignará. Si su aserción de SAML tiene varios roles, puede especificar este parámetro para elegir el rol que se va a asumir. El rol especificado debe estar presente en la aserción de SAML. Para obtener más información sobre los roles de ARN, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
preferred_role	Opcional	none	preferred_role=arn:aws:IAM::123456789012:id/user1;

## Duración de la sesión

La duración de la sesión del rol en segundos. Para obtener más información sobre la duración de la sesión, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
duration	Opcional	900	duration=900;

## ID de inquilino

Especifica el ID de inquilino de la aplicación.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
idp_tenant	Obligatoria	none	idp_tenant=123zz112z-z12d-1z1f-11zz-f111aa111234;

## ID de cliente

Especifica el ID de cliente de la aplicación.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
client_id	Obligatoria	none	client_id=9178ac27-a1bc-1a2b-1a2b-a123abcd1234;

## Tiempo de espera

El tiempo, en segundos, antes de que el complemento deje de esperar la respuesta de SAML de Azure AD.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
timeout	Opcional	120	timeout=90;

## Habilitar la caché de los archivos de Azure

Activa una caché de credenciales temporal. Este parámetro de conexión a la caché de archivos permite almacenar en caché las credenciales temporales y reutilizarlas entre los múltiples procesos.

Utilice esta opción para reducir el número de ventanas del explorador abiertas cuando utilice herramientas de BI como Microsoft Power BI.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
browser_azure_cache	Opcional	1	browser_azure_cache=0;

### Browser SAML

Browser SAML es un complemento de autenticación genérico que puede funcionar con proveedores de identidad basados en SAML y admitir la autenticación multifactor. Para obtener información detallada sobre la configuración, consulte [Configuración del inicio de sesión único con ODBC, SAML 2.0 y el proveedor de identidades Okta](#).

### Tipo de autenticación

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
AuthenticationType	Obligatoria	IAM Credentials	AuthenticationType=BrowserSAML;

### Rol preferido

El nombre de recurso de Amazon (ARN) del rol que se asignará. Si su aserción de SAML tiene varios roles, puede especificar este parámetro para elegir el rol que se va a asumir. Este rol debe estar presente en la aserción de SAML. Para obtener más información sobre los roles de ARN, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
preferred_role	Opcional	none	preferred_role=arn:aws:IAM::123456789012:id/user1;

### Duración de la sesión

La duración de la sesión del rol en segundos. Para obtener más información, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
duration	Opcional	900	duration=900;

### URL de inicio de sesión

La URL de inicio de sesión único que se muestra para la aplicación.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
login_url	Obligatoria	none	login_url=https://trial-1234567.okta.com/app/trial-1234567_okta_browsersaml_1/zzz4izzzAzDFBzZz1234/sso/saml;

## Puerto de escucha

El número del puerto que se usa para escuchar la respuesta de SAML. Este valor debe coincidir con la URL de IAM Identity Center con la que configuró el proveedor de identidad (por ejemplo, `http://localhost:7890/athena`).

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
<code>listen_port</code>	Opcional	7890	<code>listen_port=7890;</code>

## Tiempo de espera

El tiempo, en segundos, antes de que el complemento deje de esperar la respuesta de SAML del proveedor de identidad.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
<code>timeout</code>	Opcional	120	<code>timeout=90;</code>

## Browser SSO OIDC

Browser SSO OIDC es un complemento de autenticación que funciona con AWS IAM Identity Center. Para obtener información sobre cómo habilitar y usar IAM Identity Center, consulte [Paso 1: habilitar IAM Identity Center](#) en la Guía del usuario de AWS IAM Identity Center.

## Tipo de autenticación

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
<code>AuthenticationType</code>	Obligatoria	IAM Credentials	<code>AuthenticationType=BrowserSSOIDC;</code>



## URL de inicio de IAM Identity Center

La URL del portal de acceso de AWS. La acción de la API [StartDeviceAuthorization](#) de IAM Identity Center utiliza este valor para el parámetro `startUrl`.

Para copiar la URL del portal de acceso de AWS

1. Inicie sesión en la AWS Management Console y abra la consola de AWS IAM Identity Center en <https://console.aws.amazon.com/singlesignon/>.
2. En el panel de navegación, seleccione Configuración.
3. En la página Configuración, en Origen de identidad, elija el icono del portapapeles para la URL del portal de acceso de AWS.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
<code>sso_oidc_start_url</code>	Obligatoria	none	<code>sso_oidc_start_url=https://app_id.awsapps.com/start;</code>

## Región de IAM Identity Center

La Región de AWS donde está configurado su SSO. Los clientes del SDK de AWS `SSOOIDCClient` y `SSOClient` utilizan este valor para el parámetro `region`.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
<code>sso_oidc_region</code>	Obligatoria	none	<code>sso_oidc_region=us-east-1;</code>

## Ámbitos

La lista de ámbitos que define el cliente. Tras la autorización, esta lista restringe los permisos cuando se concede un token de acceso. La acción de la API [RegisterClient](#) de IAM Identity Center utiliza este valor para el parámetro `scopes`.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
<code>sso_oidc_scopes</code>	Opcional	none	<code>sso_oidc_scopes=scope1,scope2,scope3;</code>

## ID de cuenta

El identificador de la Cuenta de AWS que se asigna al usuario. La API [GetRoleCredentials](#) de IAM Identity Center utiliza este valor para el parámetro `accountId`.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
<code>sso_oidc_account_id</code>	Obligatoria	none	<code>sso_oidc_account_id=123456789123;</code>

## Nombre de rol

El nombre descriptivo del rol que se asigna al usuario. El nombre que especifique para este conjunto de permisos aparece en el portal de acceso de AWS como un rol disponible. La acción de la API [GetRoleCredentials](#) de IAM Identity Center utiliza este valor para el parámetro `roleName`.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
<code>sso_oidc_role_name</code>	Obligatoria	none	<code>sso_oidc_role_name</code>

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
			=AthenaRe adAccess;

### Tiempo de espera

El número de segundos que la API de SSO de sondeo debe comprobar si existe el token de acceso.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
sso_oidc_timeout	Opcional	120	sso_oidc_ timeout=60;

### Habilitar la caché de los archivos

Activa una caché de credenciales temporal. Este parámetro de conexión a la caché de archivos permite almacenar en caché las credenciales temporales y reutilizarlas entre los múltiples procesos. Utilice esta opción para reducir el número de ventanas del explorador abiertas cuando utilice herramientas de BI como Microsoft Power BI.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
sso_oidc_cache	Opcional	1	sso_oidc_cache=0;

### Credenciales predeterminadas

Puede usar las credenciales predeterminadas que configure en su sistema cliente para conectarse a Amazon Athena. Para obtener información sobre el uso de las credenciales predeterminadas, consulte [Uso de la cadena de proveedores de credenciales predeterminadas](#) en la Guía para desarrolladores de AWS SDK for Java.

## Tipo de autenticación

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
AuthenticationType	Obligatoria	IAM Credentials	AuthenticationType=DefaultCredentials;

## Credenciales externas

Las credenciales externas son un complemento de autenticación genérico que puede usar para conectarse a cualquier proveedor de identidad externo basado en SAML. Para usar el complemento, debe pasar un archivo ejecutable que devuelva una respuesta SAML.

## Tipo de autenticación

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
AuthenticationType	Obligatoria	IAM Credentials	AuthenticationType=ExternalCredentials;

## Ruta de ejecución

La ruta al ejecutable que tiene la lógica de su proveedor de credenciales personalizado basado en SAML. El resultado del ejecutable debe ser la respuesta SAML analizada del proveedor de identidad.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
ExecutablePath	Obligatoria	none	ExecutablePath=C:\Users <i>\user_name \external_</i> <i>credential.exe</i>

### Lista de argumentos

La lista de argumentos que desea pasar al ejecutable.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
ArgumentList	Opcional	none	ArgumentList= <i>arg1 arg2</i> <i>arg3</i>

### Perfil de instancia

Este tipo de autenticación se utiliza en instancias de EC2 y se entrega a través del servicio de metadatos de Amazon EC2.

### Tipo de autenticación

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
AuthenticationType	Obligatoria	IAM Credentia ls	AuthenticationType =Instance Profile;

## JWT

El complemento JWT (JSON Web Token) proporciona una interfaz que utiliza los JSON Web Tokens para asumir un rol de IAM de Amazon. La configuración depende del proveedor de identidad. Para obtener información sobre cómo configurar la federación para Google Cloud y AWS, consulte cómo [configurar la federación de identidades en las cargas de trabajo con AWS o Azure](#) en la documentación de Google Cloud.

### Tipo de autenticación

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
AuthenticationType	Obligatoria	IAM Credentials	AuthenticationType=JWT;

### Rol preferido

El nombre de recurso de Amazon (ARN) del rol que se asignará. Para obtener más información sobre los roles de ARN, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
preferred_role	Opcional	none	preferred_role=arn:aws:IAM:123456789012:id/user1;

### Duración de la sesión

La duración de la sesión del rol en segundos. Para obtener más información sobre la duración de la sesión, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
duration	Opcional	900	duration=900;

## Token web JSON

El token web JSON que se utiliza para recuperar las credenciales temporales de IAM mediante la acción de la API de AWS STS [AssumeRoleWithWebIdentity](#). Para obtener información sobre cómo generar tokens web JSON para los usuarios de Google Cloud Platform (GCP), consulte el [uso de tokens de OAuth de JWT](#) en la documentación de Google Cloud.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
web_identity_token	Obligatoria	none	web_identity_token=eyJhbGc. ..<remainder of token>;

## Nombre de sesión de rol

Un nombre para la sesión. Una técnica habitual consiste en utilizar el nombre o identificador del usuario de la aplicación como nombre de la sesión del rol. Esto asocia de forma práctica las credenciales de seguridad temporales que utiliza la aplicación con el usuario correspondiente.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
role_session_name	Obligatoria	none	role_session_name=familiarn ame;

## Okta

Okta es un complemento de autenticación basado en SAML que funciona con el proveedor de identidad de Okta. Para obtener información sobre la configuración de la federación para Okta y

Amazon Athena, consulte [Configuración de SSO para ODBC mediante el complemento Okta y el proveedor de identidad Okta](#).

### Tipo de autenticación

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
AuthenticationType	Obligatoria	IAM Credentials	AuthenticationType=Okta;

### ID de usuario

El nombre de su usuario de Okta.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
UID	Obligatoria	none	UID=jane.doe@org.com;

### Password

La contraseña de su usuario de Okta.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
PWD	Obligatoria	none	PWD=oktauserpasswordexample;



## Rol preferido

El nombre de recurso de Amazon (ARN) del rol que se asignará. Para obtener más información sobre los roles de ARN, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
preferred_role	Opcional	none	preferred_role=arn:aws:IAM:123456789012:id/user1;

## Duración de la sesión

La duración de la sesión del rol en segundos. Para obtener más información, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
duration	Opcional	900	duration=900;

## Host del IdP

La URL de su organización de Okta. Puede extraer el parámetro `idp_host` de la URL de Incrustar enlace en la aplicación de Okta. Para ver los pasos, consulte [Recuperación de información de configuración de ODBC desde Okta](#). El primer segmento que sigue a `https://`, incluido `okta.com`, es el host del proveedor de identidad (por ejemplo, `http://trial-1234567.okta.com`).

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
idp_host	Obligatoria	None	idp_host=dev-99999 999.okta.com;

### Puerto del IdP

El número de puerto que se va a utilizar para conectarse al host del proveedor de identidad.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
idp_port	Obligatoria	None	idp_port=443;

### ID de aplicación de Okta

El identificador de dos partes de su aplicación. Puede extraer el parámetro `app_id` de la URL de Incrustar enlace en la aplicación de Okta. Para ver los pasos, consulte [Recuperación de información de configuración de ODBC desde Okta](#). El ID de la aplicación son los dos últimos segmentos de la URL, incluida la barra del medio. Los segmentos son dos cadenas de 20 caracteres con una combinación de números y letras mayúsculas y minúsculas (por ejemplo, `Abc1de2fghi3J45kL678/abc1defghij2klmNo3p4`).

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
app_id	Obligatoria	None	app_id=0o a25kx8ze9 A3example /alnexamp lea0piaWa0g7;

## Nombre de aplicación de Okta

El nombre de la aplicación de Okta.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
app_name	Obligatoria	None	app_name= amazon_aws_redshift;

## Tiempo de espera de Okta

Especifica en segundos la duración de espera del código de autenticación multifactor (MFA).

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
okta_mfa_wait_time	Opcional	10	okta_mfa_ wait_time=20;

## Tipo de MFA de Okta

El tipo de factor de MFA. Los tipos compatibles son Google Authenticator, SMS (Okta), Okta Verify con Push y Okta Verify con TOTP. Las políticas de seguridad de las organizaciones individuales determinan si se requiere o no la MFA para el inicio de sesión de los usuarios.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Valores posibles	Ejemplo de la cadena de conexión
okta_mfa_type	Optional	None	googleauthenticator, smsauthentication, oktaverif	okta_mfa_ type=oktaverifywithpush;

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Valores posibles	Ejemplo de la cadena de conexión
			ywithpush , oktaverif ywithtotp	

### Número de teléfono de Okta

El número de teléfono que se va a usar con la autenticación AWS SMS. Este parámetro solo es obligatorio para la inscripción multifactor. Si su número de teléfono móvil ya está registrado o si la política de seguridad no utiliza la autenticación AWS SMS, puede ignorar este campo.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
okta_mfa_phone_number	Es obligatorio para la inscripción MFA; de lo contrario, es opcional.	None	okta_mfa_phone_number=19991234567;

### Habilitación de la caché de archivos de Okta

Activa una caché de credenciales temporal. Este parámetro de conexión a la caché de archivos permite almacenar en caché las credenciales temporales y reutilizarlas entre los múltiples procesos abiertos por las aplicaciones de BI. Utilice esta opción para evitar el límite de limitación de la API de Okta.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
okta_cache	Opcional	0	okta_cache=1;

## Ping

Ping es un complemento basado en SAML que funciona con el proveedor de identidad [PingFederate](#).

### Tipo de autenticación

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
AuthenticationType	Obligatoria	IAM Credentials	AuthenticationType=Ping;

### ID de usuario

El nombre de usuario del servidor PingFederate.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
UID	Obligatoria	none	UID=pingusername@omain.com;

### Password

La contraseña del servidor PingFederate.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
PWD	Obligatoria	none	PWD=pingpassword;

## Rol preferido

El nombre de recurso de Amazon (ARN) del rol que se asignará. Si su aserción de SAML tiene varios roles, puede especificar este parámetro para elegir el rol que se va a asumir. Este rol debe estar presente en la aserción de SAML. Para obtener más información sobre los roles de ARN, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
preferred_role	Opcional	none	preferred_role=arn:aws:iam: :123456789012:id/user1;

## Duración de la sesión

La duración de la sesión del rol en segundos. Para obtener más información sobre la duración de la sesión, consulte [AssumeRole](#) en la Referencia de la API de AWS Security Token Service.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
duration	Opcional	900	duration=900;

## Host del IdP

La dirección del servidor de Ping. Para encontrar la dirección, visite la siguiente URL y consulte el campo Punto de conexión de la aplicación de SSO.

```
https://your-pf-host-#:9999/pingfederate/your-pf-app#/spConnections
```

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
idp_host	Obligatoria	none	idp_host=ec2-1-83-65-12.com pute-1.amazonaws.com;

### Puerto del IdP

El número de puerto que se va a utilizar para conectarse al host del proveedor de identidad.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
idp_port	Obligatoria	None	idp_port=443;

### SPID de socio

La dirección del proveedor de servicios. Para encontrar la dirección del proveedor de servicios, visite la siguiente URL y consulte el campo Punto de conexión de la aplicación de SSO.

<https://your-pf-host-#:9999/pingfederate/your-pf-app#/spConnections>

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
partner_spid	Obligatoria	None	partner_spid=https://us-east-1.signin.amazonaws.com/platform/saml/<...>;

### Parámetro URI de Ping

Pasa un argumento URI para una solicitud de autenticación a Ping. Utilice este parámetro para evitar la limitación de función única de Lake Formation. Configure Ping para que reconozca el parámetro

transferido y compruebe que el rol transferido existe en la lista de roles asignados al usuario. A continuación, envíe un rol único en la aserción de SAML.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
ping_uri_param	Opcional	None	ping_uri_param=rol e=my_iam_role;

### Parámetros de autenticación comunes

Como se indica, los parámetros de esta sección son comunes a los tipos de autenticación.

#### Utilizar proxy para el IdP

Permite la comunicación entre el controlador y el proveedor de identidad a través del proxy. Esta opción está disponible para los siguientes complementos de autenticación:

- AD FS
- Azure AD
- Browser Azure AD
- Browser SSO OIDC
- JWT
- Okta
- Ping

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
UseProxyForIdP	Opcional	0	UseProxyForIdP=1;



## Utilizar Lake Formation

Utiliza la acción de la API de Lake Formation [AssumeDecoratedRoleWithSAML](#) para recuperar las credenciales de IAM temporales en lugar de la acción de la API de AWS STS [AssumeRoleWithSAML](#). Esta opción está disponible para los complementos de autenticación Azure AD, Browser Azure AD, Browser SAML, Okta, Ping y AD FS.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
LakeformationEnabled	Opcional	0	LakeformationEnabled=1;

## SSL inseguro (IdP)

Deshabilita el SSL cuando se comunica con el proveedor de identidad. Esta opción está disponible para los complementos de autenticación Azure AD, Browser Azure AD, Okta, Ping y AD FS.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
SSL_Insecure	Opcional	0	SSL_Insecure=1;

## Anulaciones de puntos de conexión

### Anulación del punto de conexión de Athena

La clase `endpointOverride` `ClientConfiguration` usa este valor para anular el punto de conexión HTTP predeterminado del cliente Amazon Athena. Para obtener más información, consulte [Configuración de cliente de AWS](#) en la Guía para desarrolladores de AWS SDK for C++.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
EndpointOverride	Opcional	none	EndpointOverride=athena.us-west-2.amazonaws.com;

## Anulación del punto de conexión de transmisión de Athena

El método `ClientConfiguration.endpointOverride` usa este valor para anular el punto de conexión HTTP predeterminado del cliente de transmisión Amazon Athena. Para obtener más información, consulte [Configuración de cliente de AWS](#) en la Guía para desarrolladores de AWS SDK for C++. El servicio de transmisión de Athena está disponible a través del puerto 444.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
StreamingEndpointOverride	Opcional	none	<code>StreamingEndpointOverride=athena.us-west-1.amazonaws.com:444;</code>

## Anulación del punto de conexión de AWS STS

El método `ClientConfiguration.endpointOverride` usa este valor para anular el punto de conexión HTTP predeterminado del cliente AWS STS. Para obtener más información, consulte [Configuración de cliente de AWS](#) en la Guía para desarrolladores de AWS SDK for C++.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
StsEndpointOverride	Opcional	none	<code>StsEndpointOverride=sts.us-west-1.amazonaws.com;</code>

## Anulación del punto de conexión de Lake Formation

El método `ClientConfiguration.endpointOverride` usa este valor para anular el punto de conexión HTTP predeterminado del cliente Lake Formation. Para obtener más información, consulte [Configuración de cliente de AWS](#) en la Guía para desarrolladores de AWS SDK for C++.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
LakeFormationEndpointOverride	Opcional	none	LakeFormationEndpointOverride=lakeformation.us-west-1.amazonaws.com;

### Anulación del punto de conexión SSO

El método `ClientConfiguration.endpointOverride` usa este valor para anular el punto de conexión HTTP predeterminado del cliente SSO. Para obtener más información, consulte [Configuración de cliente de AWS](#) en la Guía para desarrolladores de AWS SDK for C++.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
SSOEndpointOverride	Opcional	none	SSOEndpointOverride=portal.sso.us-east-2.amazonaws.com;

### Anulación del punto de conexión SSO OIDC

El método `ClientConfiguration.endpointOverride` usa este valor para anular el punto de conexión HTTP predeterminado del cliente SSO OIDC. Para obtener más información, consulte [Configuración de cliente de AWS](#) en la Guía para desarrolladores de AWS SDK for C++.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
SSOIDCEndpointOverride	Opcional	none	SSOIDCEndpointOverride=oidc.us-east-2.amazonaws.com

## Opciones avanzadas

### Tamaño de recuperación

El número máximo de resultados (filas) que se devuelven en esta solicitud. Para obtener información sobre el parámetro, consulte [GetQuery MaxResults](#). Para la API de transmisión, el valor máximo es 10 000 000.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
RowsToFetchPerBlock	Opcional	1000 en caso de no transmisión  20000 en caso de transmisión	RowsToFetchPerBlock=20000;

### Habilitar la reutilización de resultados

Especifica si los resultados de la consulta anterior se pueden reutilizar al ejecutar la consulta. Para obtener información sobre el parámetro, consulte [ResultReuseByAgeConfiguration](#).

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
EnableResultReuse	Opcional	0	EnableResultReuse=1;

### Antigüedad máxima de reutilización de resultados

Especifica, en minutos, la antigüedad máxima de un resultado de consulta anterior que Athena debe tener en cuenta para su reutilización. Para obtener información sobre el parámetro, consulte [ResultReuseByAgeConfiguration](#).

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
ReusedResultMaxAgeInMinutes	Opcional	60	ReusedResultMaxAgeInMinutes=90;

### Habilitar la API de transmisión

Elige si se va a utilizar la API de transmisión de Athena para recuperar el conjunto de resultados.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
UseResultsetStreaming	Opcional	0	UseResultsetStreaming=1;

### Habilitar recuperación de S3

Recupera el conjunto de resultados generado por Athena desde el bucket de Amazon S3 al interactuar directamente con Amazon S3.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
EnableS3Fetcher	Opcional	1	EnableS3Fetcher=1;

### Utilizar varios hilos de S3

Recupera datos de Amazon S3 mediante varios hilos. Cuando esta opción está habilitada, el archivo de resultados almacenado en el bucket de Amazon S3 se recupera en paralelo mediante varios hilos.

Habilite esta opción solo si tiene un buen ancho de banda de la red. Por ejemplo, en nuestras mediciones en una instancia [c5.2xlarge](#) de EC2, un cliente S3 de un solo hilo alcanzó 1 Gbps, mientras que los clientes S3 de hilos múltiples alcanzaron un rendimiento de red de 4 Gbps.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
UseMultipleS3Threads	Opcional	0	UseMultipleS3Threads=1;

### Utilizar un único catálogo y esquema

De forma predeterminada, el controlador ODBC consulta a Athena para obtener la lista de catálogos y esquemas disponibles. Esta opción obliga al controlador a utilizar el catálogo y el esquema especificados en el cuadro de diálogo de configuración del administrador de orígenes de datos ODBC o en los parámetros de conexión.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
UseSingleCatalogAndSchema	Opcional	0	UseSingleCatalogAndSchema=1;

### Utilice una consulta para enumerar tablas

Para los tipos de catálogo LAMBDA, permite que el controlador ODBC envíe una consulta [SHOW TABLES](#) para obtener una lista de las tablas disponibles. Esta configuración es la predeterminada. Si este parámetro se establece en 0, el controlador ODBC utiliza la API [ListTableMetadata](#) de Athena para obtener una lista de las tablas disponibles. Tenga en cuenta que, en el caso de los tipos de catálogos LAMBDA, el uso de `ListTableMetadata` conduce a una regresión del rendimiento.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
UseQueryToListTables	Opcional	1	UseQueryToListTables=1;

## Utilice WCHAR para los tipos de cadenas

De forma predeterminada, el controlador ODBC utiliza SQL\_CHAR y SQL\_VARCHAR para Athena, los tipos de datos de cadena char, varchar, string, array, map<>, struct<> y row. Si se establece este parámetro a 1, se fuerza al controlador a utilizar SQL\_WCHAR y SQL\_WVARCHAR para los tipos de datos de cadena. Los tipos de caracteres y variables anchos se utilizan para garantizar que los caracteres de diferentes idiomas se puedan almacenar y recuperar correctamente.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
UseWCharForStringTypes	Opcional	0	UseWCharForStringTypes=1;

## Consultar catálogos externos

Especifica si el controlador necesita consultar catálogos externos desde Athena. Para obtener más información, consulte [Migración al controlador ODBC 2.x](#).

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
QueryExternalCatalogs	Opcional	0	QueryExternalCatalogs=1;

## Verificar SSL

Controla si se deben verificar los certificados SSL cuando se usa el SDK de AWS. Este valor se transfiere al parámetro `ClientConfiguration.verifySSL`. Para obtener más información, consulte [Configuración de cliente de AWS](#) en la Guía para desarrolladores de AWS SDK for C++.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
VerifySSL	Opcional	1	VerifySSL=0;

## Tamaño del bloque de resultados de S3

Especifica, en bytes, el tamaño del bloque que se va a descargar para una sola solicitud de API [GetObject](#) de Amazon S3. El valor predeterminado es 67108864 (64 MB). Los valores mínimo y máximo permitidos son 10485760 (10 MB) y 2146435072 (aproximadamente 2 GB).

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
S3ResultBlockSize	Opcional	67108864	S3ResultBlockSize=268435456;

## Longitud de columna de cadena

Especifica la longitud de las columnas con el tipo de datos `string`. Como Athena usa el [tipo de datos de cadena Apache Hive](#), que no tiene una precisión definida, la longitud predeterminada que informa Athena es 2147483647 (`INT_MAX`). Como las herramientas de inteligencia empresarial suelen preasignar memoria a las columnas, esto puede provocar un consumo elevado de memoria. Para evitarlo, el controlador ODBC de Athena limita la precisión notificada para las columnas del tipo de datos `string` y expone el parámetro de conexión `StringColumnLength` para que se pueda cambiar el valor predeterminado.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
StringColumnLength	Opcional	255	StringColumnLength=65535;

## Longitud de columna de tipo complejo

Especifica la longitud de las columnas con tipos de datos complejos, como `map`, `struct`, y `array`. Al igual que [StringColumnLength](#), Athena informa de una precisión de 0 para las columnas con tipos de datos complejos. El controlador ODBC de Athena establece la precisión predeterminada para las columnas con tipos de datos complejos y expone el parámetro de conexión `ComplexTypeColumnLength` para que se pueda cambiar el valor predeterminado.



Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
ComplexTypeColumnLength	Opcional	65535	ComplexTypeColumnLength=123456;

### Certificado de CA de confianza

Indica al cliente HTTP dónde encontrar el almacén de confianza de certificados SSL. Este valor se transfiere al parámetro `ClientConfiguration.caFile`. Para obtener más información, consulte [Configuración de cliente de AWS](#) en la Guía para desarrolladores de AWS SDK for C++.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
TrustedCerts	Opcional	%INSTALL_PATH%/bin	TrustedCerts=C:\\Program Files\\Amazon Athena ODBC Driver\\bin\\cacert.pem;

### Periodo mínimo de sondeo

Especifica el valor mínimo en milisegundos que debe esperarse antes de sondear Athena para determinar el estado de ejecución de la consulta.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
MinQueryExecutionPollingInterval	Opcional	100	MinQueryExecutionPollingInterval=200;

## Periodo máximo de sondeo

Especifica el valor máximo en milisegundos que se debe esperar antes de sondear Athena para determinar el estado de ejecución de la consulta.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
MaxQueryExecutionPollingInterval	Opcional	60000	MaxQueryExecutionPollingInterval=1000;

## Multiplicador de sondeo

Especifica el factor para aumentar el periodo de sondeo. De forma predeterminada, el sondeo comienza con el valor del periodo mínimo de sondeo y se duplica con cada sondeo hasta alcanzar el valor del periodo máximo de sondeo.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
QueryExecutionPollingIntervalMultiplier	Opcional	2	QueryExecutionPollingIntervalMultiplier=2;

## Duración máxima del sondeo

Especifica el valor máximo en milisegundos del tiempo durante el cual un controlador puede sondear Athena para determinar el estado de ejecución de la consulta.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
MaxPollDuration	Opcional	1800000	MaxPollDuration=1800000;

## Tiempo de espera de la conexión

La cantidad de tiempo (en milisegundos) que la conexión HTTP espera a que se establezca una conexión. Este valor se establece para el cliente de Athena `ClientConfiguration.connectTimeoutMs`. Si no se especifica, se usa el valor predeterminado de curl. Para obtener más información sobre los parámetros de conexión, consulte [Configuración de clientes](#) en la Guía para desarrolladores de AWS SDK for Java.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
ConnectionTimeout	Opcional	0	Connectio nTimeout=2000;

## Tiempo de espera de la solicitud

Especifica el tiempo de espera de lectura del socket para los clientes HTTP. Este valor se establece para el parámetro `ClientConfiguration.requestTimeoutMs` del cliente de Athena. Para obtener información sobre el parámetro, consulte [Configuración de clientes](#) en la Guía para desarrolladores de AWS SDK for Java.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
RequestTimeout	Opcional	10000	RequestTi meout=30000;

## Opciones de proxy

### Host del proxy.

Si necesita que los usuarios utilicen un proxy, utilice este parámetro para configurar el host del proxy. Este parámetro corresponde al parámetro `ClientConfiguration.proxyHost` en el SDK de AWS. Para obtener más información, consulte [Configuración de cliente de AWS](#) en la Guía para desarrolladores de AWS SDK for C++.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
ProxyHost	Opcional	none	ProxyHost=127.0.0.1;

Puerto del proxy.

Utilice este parámetro para configurar el puerto del proxy. Este parámetro corresponde al parámetro `ClientConfiguration.proxyPort` en el SDK de AWS. Para obtener más información, consulte [Configuración de cliente de AWS](#) en la Guía para desarrolladores de AWS SDK for C++.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
ProxyPort	Opcional	none	ProxyPort=8888;

Nombre de usuario del proxy.

Utilice este parámetro para establecer el nombre de usuario del proxy. Este parámetro corresponde al parámetro `ClientConfiguration.proxyUserName` en el SDK de AWS. Para obtener más información, consulte [Configuración de cliente de AWS](#) en la Guía para desarrolladores de AWS SDK for C++.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
ProxyUID	Opcional	none	ProxyUID=username;

Contraseña del proxy.

Utilice este parámetro para establecer la contraseña del proxy. Este parámetro corresponde al parámetro `ClientConfiguration.proxyPassword` en el SDK de AWS. Para obtener más información, consulte [Configuración de cliente de AWS](#) en la Guía para desarrolladores de AWS SDK for C++.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
ProxyPWD	Opcional	none	ProxyPWD=password;

### Host sin proxy

Utilice este parámetro opcional para especificar un host al que se pueda conectar el controlador sin utilizar un proxy. Este parámetro corresponde al parámetro `ClientConfiguration.nonProxyHosts` en el SDK de AWS. Para obtener más información, consulte [Configuración de cliente de AWS](#) en la Guía para desarrolladores de AWS SDK for C++.

El parámetro de conexión `NonProxyHost` se transfiere a la opción curl `CURLOPT_NOPROXY`. Para obtener información sobre el formato `CURLOPT_NOPROXY`, consulte [CURLOPT\\_NOPROXY](#) en la documentación de curl.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
NonProxyHost	Opcional	none	NonProxyHost=.amazonaws.com,localhost,.example.net,.example.com;

### Utilizar proxy

Habilita el tráfico de usuarios a través del proxy especificado.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
UseProxy	Opcional	none	UseProxy=1;

## Opciones de registro

Se requieren derechos de administrador para modificar la configuración que se describe aquí. Para realizar los cambios, puede utilizar el cuadro de diálogo Opciones de registro del administrador de orígenes de datos ODBC o modificar el registro de Windows directamente.

### Nivel de registro

Esta opción habilita los registros del controlador ODBC. En Windows, puede utilizar el registro o un cuadro de diálogo para habilitar o deshabilitar el registro. La opción se encuentra en la siguiente ruta de registro:

```
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Amazon Athena\ODBC\Driver
```

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
LogLevel	Opcional	0	LogLevel=1;

### Log path (Ruta de registro)

Especifica la ruta al archivo en el que se almacenan los registros del controlador ODBC. Puede utilizar el registro o un cuadro de diálogo para establecer este valor. La opción se encuentra en la siguiente ruta de registro:

```
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Amazon Athena\ODBC\Driver
```

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
LogPath	Opcional	none	LogPath=C:\Users\ <i>username</i> \projects\internal\trunk\;

## Utilizar el registrador de AWS

Especifica si el registro del SDK de AWS está habilitado. Especifique 1 para habilitarlo y 0 para deshabilitarlo.

Nombre de la cadena de conexión	Tipo de parámetro	Valor predeterminado	Ejemplo de la cadena de conexión
UseAwsLogger	Opcional	0	UseAwsLogger=1;

## Migración al controlador ODBC 2.x

Como la mayoría de los parámetros de conexión ODBC 2.x de Athena son compatibles con versiones anteriores del controlador ODBC 1.x, puede reutilizar la mayor parte de la cadena de conexión existente con el controlador ODBC 2.x de Athena. Sin embargo, es necesario modificar los siguientes parámetros de conexión.

### Nivel de registro

Si bien el controlador ODBC actual ofrece una gama de opciones de registro disponibles, desde LOG\_OFF (0) hasta LOG\_TRACE (6), el controlador ODBC de Amazon Athena solo tiene dos valores: 0 (deshabilitado) y 1 (habilitado).

Para obtener más información sobre el registro del controlador ODBC 2.x, consulte [Opciones de registro](#).

	Controlador ODBC 1.x	Controlador ODBC 2.x
Nombre de la cadena de conexión	LogLevel	LogLevel
Tipo de parámetro	Opcional	Opcional
Valor predeterminado	0	0
Valores posibles	0-6	0, 1
Ejemplo de la cadena de conexión	LogLevel=6;	LogLevel=1;

## MetadataRetrievalMethod

El controlador ODBC actual ofrece varias opciones para recuperar los metadatos de Athena. El controlador ODBC de Amazon Athena ya no utiliza MetadataRetrievalMethod y siempre usa la API de Amazon Athena para extraer metadatos.

Se ha incorporado la marca QueryExternalCatalogs en Athena para consultar catálogos externos. Para consultar catálogos externos con el controlador ODBC actual, configure MetadataRetrievalMethod en ProxyAPI. Para consultar catálogos externos con el controlador ODBC de Athena, configure QueryExternalCatalogs en 1.

	Controlador ODBC 1.x	Controlador ODBC 2.x
Nombre de la cadena de conexión	MetadataRetrievalMethod	QueryExternalCatalogs
Tipo de parámetro	Opcional	Opcional
Valor predeterminado	Auto	0
Valores posibles	Auto, AWS Glue, ProxyAPI, Query	0,1
Ejemplo de la cadena de conexión	MetadataRetrievalMethod=ProxyAPI;	QueryExternalCatalogs=1;

### Prueba de conexión

Al probar la conexión de un controlador ODBC 1.x, el controlador ejecuta una consulta `SELECT 1` que genera dos archivos en el bucket de Amazon S3: uno para el conjunto de resultados y otro para los metadatos. La conexión de prueba se cobra de acuerdo con la política de [precios de Amazon Athena](#).

Al probar la conexión de un controlador ODBC 2.x, el controlador llama a la acción de la API de Athena [GetWorkGroup](#). La llamada utiliza el tipo de autenticación y el proveedor de credenciales correspondiente que especificó para recuperar las credenciales. La prueba de conexión es gratuita si utiliza el controlador ODBC 2.x y no genera resultados de consultas en el bucket de Amazon S3.



## Solución de problemas del controlador ODBC 2.x

Si tiene problemas con el controlador ODBC de Amazon Athena, puede ponerse en contacto con AWS Support (en la AWS Management Console, elija Asistencia, Centro de asistencia).

Asegúrese de incluir la siguiente información y de proporcionar cualquier detalle adicional que ayude al equipo de asistencia a entender su caso de uso.

- **Descripción:** (obligatoria) una descripción que incluya información detallada sobre el caso de uso y la diferencia entre el comportamiento esperado y el observado. Incluya cualquier información que pueda ayudar a los ingenieros de asistencia a resolver el problema con facilidad. Si el problema es intermitente, especifique las fechas, las marcas temporales o los puntos de intervalo en los que se produjo el problema.
- **Información sobre la versión:** (obligatoria) información sobre la versión del controlador, el sistema operativo y las aplicaciones que utilizó. Por ejemplo, "Controlador ODBC versión 1.2.3, Windows 10 (x64), Power BI".
- **Archivos de registro:** (obligatorio) el número mínimo de archivos de registro del controlador ODBC necesarios para comprender el problema. Para obtener información sobre las opciones de registro del controlador ODBC 2.x, consulte [Opciones de registro](#).
- **Cadena de conexión:** (obligatoria) la cadena de conexión ODBC o una captura de pantalla del cuadro de diálogo en la que se vean los parámetros de conexión que utilizó. Para obtener información sobre los parámetros de conexión, consulte [Parámetros de conexión ODBC 2.x de Athena](#).
- **Pasos del problema:** (opcional) si es posible, incluya pasos o un programa independiente que pueda ayudar a reproducir el problema.
- **Información sobre errores de consulta:** (opcional) si tiene errores relacionados con consultas DML o DDL, incluya la siguiente información:
  - Una versión completa o simplificada de la consulta DML o DDL fallida.
  - El ID y la Región de AWS de la cuenta que utilizó y el ID de ejecución de la consulta.
- **Errores de SAML:** (opcional) si tiene algún problema relacionado a la autenticación con la aserción de SAML, incluya la siguiente información:
  - El proveedor de identidad y el complemento de autenticación que utilizó.
  - Un ejemplo con el token de SAML.

## Notas de la versión de ODBC 2.x de Amazon Athena

Estas notas de la versión proporcionan detalles sobre las mejoras, las características, los problemas conocidos y los cambios en el flujo de trabajo del controlador ODBC 2.x de Amazon Athena.

### 2.0.3.0

Se lanzó el 08/04/2023

El controlador ODBC v2.0.3.0 de Amazon Athena cuenta con las siguientes mejoras y correcciones.

#### Mejoras

- Se agregó compatibilidad con MFA para el complemento de autenticación Okta en las plataformas Linux y Mac.
- Tanto la biblioteca `athena-odbc.dll` como el instalador `AmazonAthenaODBC-2.x.x.x.msi` para Windows ya están firmados.
- Se actualizó el archivo `cacert.pem` de certificado de CA que está instalado con el controlador.
- Se mejoró el tiempo necesario para incluir las tablas en los catálogos de Lambda. Para los tipos de catálogos de LAMBDA, el controlador ODBC ahora puede enviar una consulta [SHOW TABLES](#) para obtener una lista de las tablas disponibles. Para obtener más información, consulte [Utilice una consulta para enumerar tablas](#).
- Se introdujo el parámetro de conexión `UseWCharForStringTypes` para informar sobre los tipos de datos de cadenas mediante `SQL_WCHAR` y `SQL_WVARCHAR`. Para obtener más información, consulte [Utilice WCHAR para los tipos de cadenas](#).

#### Correcciones

- Se corrigió una advertencia de daños en el registro que se produjeron cuando se utilizó la herramienta `Get-ODBCDSN PowerShell`.
- Se actualizó la lógica de análisis para gestionar los comentarios al principio de las cadenas de consulta.
- Los tipos de datos de fecha y hora ahora permiten introducir cero en el campo del año.

Para descargar el nuevo controlador ODBC v2, consulte [Descarga del controlador ODBC 2.x](#). Para obtener información sobre la conexión, consulte [ODBC 2.x de Amazon Athena](#).

## 2.0.2.2

Publicado el 13/02/2022

El controlador ODBC v2.0.2.2 de Amazon Athena cuenta con las siguientes mejoras y correcciones.

### Mejoras

- Se agregaron dos parámetros de conexión, `StringColumnLength` y `ComplexTypeColumnLength`, que puede usar para cambiar la longitud de columna predeterminada para cadenas y tipos de datos complejos. Para obtener más información, consulte [Longitud de columna de cadena](#) y [Longitud de columna de tipo complejo](#).
- Se añadió soporte para los sistemas operativos Linux y macOS (Intel y ARM). Para obtener más información, consulte [Linux](#) y [macOS](#).
- Se actualizó AWS-SDK-CPP a la versión de etiquetas 1.11.245.
- La biblioteca curl se actualizó a la versión 8.6.0.

### Correcciones

- Se resolvió un problema que provocaba que se indicaran valores incorrectos en los metadatos del conjunto de resultados para tipos de datos tipo cadena en la columna de precisión.

Para descargar el controlador ODBC v2, consulte [Descarga del controlador ODBC 2.x](#). Para obtener información sobre la conexión, consulte [ODBC 2.x de Amazon Athena](#).

## 2.0.2.1

Se lanzó el 07/12/2023

El controlador ODBC v2.0.2.1 de Amazon Athena cuenta con las siguientes mejoras y correcciones.

### Mejoras

- Se mejoró la seguridad de los hilos del controlador ODBC para todas las interfaces.
- Cuando el registro está activado, actualmente, los valores de fecha y hora se graban con una precisión de milisegundos.
- Durante la autenticación con el complemento [Browser SSO OIDC](#), el terminal ahora se abre para mostrar el código del dispositivo al usuario.

## Correcciones

- Se resolvió un problema de pérdida de memoria que se producía al analizar los resultados de la API de transmisión.
- Las solicitudes de las interfaces `SQLTablePrivileges()`, `SQLSpecialColumns()`, `SQLProcedureColumns()` y `SQLProcedures()` ahora devuelven conjuntos de resultados vacíos.

Para descargar el controlador ODBC v2, consulte [Descarga del controlador ODBC 2.x](#). Para obtener información sobre la conexión, consulte [ODBC 2.x de Amazon Athena](#).

### 2.0.2.0

Se lanzó el 17/10/2023

El controlador ODBC v2.0.2.0 de Amazon Athena cuenta con las siguientes mejoras y correcciones.

#### Mejoras

- Se agregó la característica de caché de archivos para los complementos de autenticación basados en el navegador Azure AD, Browser SSO OIDC y Okta.

Las herramientas de inteligencia empresarial, como Power BI, y los complementos basados en el navegador utilizan varias ventanas del navegador. El nuevo parámetro de conexión a la caché de archivos permite almacenar en caché las credenciales temporales y reutilizarlas entre los múltiples procesos abiertos por las aplicaciones de BI.

- Las aplicaciones ahora pueden consultar información sobre el conjunto de resultados después de preparar una declaración.
- Se incrementaron los tiempos de espera predeterminados de conexión y solicitud para su uso con redes de clientes más lentas. Para obtener más información, consulte [Tiempo de espera de la conexión](#) y [Tiempo de espera de la solicitud](#).
- Se añadieron anulaciones de puntos de conexión para el SSO y el SSO OIDC. Para obtener más información, consulte [Anulaciones de puntos de conexión](#).
- Se agregó un parámetro de conexión para pasar un argumento URI para una solicitud de autenticación a Ping. Puede usar este parámetro para evitar la limitación de rol único de Lake Formation. Para obtener más información, consulte [Parámetro URI de Ping](#).

## Correcciones

- Se ha corregido un problema de desbordamiento de enteros que se producía al utilizar el mecanismo de enlace basado en filas.
- Se ha eliminado el tiempo de espera de la lista de parámetros de conexión necesarios para el complemento de autenticación OIDC del navegador SSO.
- Se agregaron las interfaces que faltaban para `SQLStatistics()`, `SQLPrimaryKeys()`, `SQLForeignKeys()` y `SQLColumnPrivileges()` y se agregó la capacidad de devolver conjuntos de resultados vacíos a pedido.

Para descargar el nuevo controlador ODBC v2, consulte [Descarga del controlador ODBC 2.x](#). Para obtener información sobre la conexión, consulte [ODBC 2.x de Amazon Athena](#).

### 2.0.1.1

Se lanzó el 10/08/2023

El controlador ODBC v2.0.1.1 de Amazon Athena cuenta con las siguientes mejoras y correcciones.

#### Mejoras

- Se agregó el registro de URI al complemento de autenticación de Okta.
- Se agregó el parámetro de rol preferido al complemento del proveedor de credenciales externo.
- Se ha agregado la gestión del prefijo del perfil en el nombre del perfil del archivo de configuración de AWS.

#### Correcciones

- Se corrigió un problema de uso de Región de AWS que se producía al trabajar con clientes de AWS STS y Lake Formation.
- Se restauraron las claves de partición que faltaban en la lista de columnas de la tabla.
- Se agregó el tipo de autenticación `BrowserSSOOIDC` que faltaba al perfil de AWS.

Para descargar el nuevo controlador ODBC v2, consulte [Descarga del controlador ODBC 2.x](#).

### 2.0.1.0

Se lanzó el 29/06/2023

## Amazon Athena lanza el controlador ODBC v2.0.1.0

Athena lanzó un controlador ODBC nuevo que mejora la experiencia de conexión, consulta y visualización de datos desde aplicaciones de inteligencia empresarial y desarrollo de SQL compatibles. La última versión del controlador ODBC de Athena es compatible con las características del controlador existente y es fácil de actualizar. La versión nueva incluye soporte para autenticar a los usuarios mediante [AWS IAM Identity Center](#). También ofrece la opción de leer los resultados de las consultas de Amazon S3, lo que permite que estos resultados se encuentren disponibles con mayor rapidez.

Para obtener más información, consulte [ODBC 2.x de Amazon Athena](#).

## Controlador ODBC 1.x de Athena

Utilice los enlaces de esta página para descargar el Contrato de licencia del controlador ODBC 1.x de Amazon Athena, los controladores ODBC y la documentación de ODBC. Para obtener más información sobre la cadena de conexión ODBC, consulte el archivo PDF de la Guía de instalación y configuración del controlador ODBC, que se puede descargar desde esta página. Para obtener información sobre los permisos, consulte [Acceso a través de conexiones JDBC y ODBC](#).

### Important

Cuando utilice el controlador ODBC 1.x, asegúrese de tener en cuenta los siguientes requisitos:

- Abrir el puerto 444: mantenga el puerto 444, que Athena utiliza para transmitir los resultados de las consultas, abierto al tráfico saliente. Cuando utilice un punto de conexión de PrivateLink para conectarse a Athena, asegúrese de que el grupo de seguridad asociado al punto de conexión de PrivateLink esté abierto al tráfico entrante en el puerto 444.
- Athena: política de flujo de resultados de consulta: agregue la acción de política `athena:GetQueryResultsStream` para las entidades principales de IAM que utilizan el controlador ODBC. Esta acción de política no se expone directamente con la API. Solo se utiliza con los controladores ODBC y JDBC como parte de la compatibilidad con los resultados de transmisión. Para ver una política de ejemplo, consulte [Política administrada de AWS: AWSQuicksightAthenaAccess](#).

## Windows

Versión de controlador	Enlace de descarga
ODBC 1.2.3.1000 para Windows de 32 bits	<a href="#">Controlador ODBC 1.2.3.1000 para Windows de 32 bits</a>
ODBC 1.2.3.1000 para Windows de 64 bits	<a href="#">Controlador ODBC 1.2.3.1000 para Windows de 64 bits</a>

## Linux

Versión de controlador	Enlace de descarga
ODBC 1.2.3.1000 para Linux de 32 bits	<a href="#">Controlador ODBC 1.2.3.1000 para Linux de 32 bits</a>
ODBC 1.2.3.1000 para Linux de 64 bits	<a href="#">Controlador ODBC 1.2.3.1000 para Linux de 64 bits</a>

## OSX

Versión de controlador	Enlace de descarga
ODBC 1.2.3.1000 para OSX	<a href="#">Controlador ODBC 1.2.3.1000 para OSX</a>

## Documentación

Contenidos	Enlace a la documentación
Contrato de licencia del controlador ODBC de Amazon Athena	<a href="#">Contrato de licencia</a>
Documentación para ODBC 1.2.3.1000	<a href="#">Guía de instalación y configuración de la versión 1.2.3.1000 del controlador ODBC</a>

Contenidos	Enlace a la documentación
Notas de la versión 1.2.3.1000 de ODBC	<a href="#">Notas de la versión 1.2.3.1000 del controlador ODBC</a>

## Notas del controlador ODBC

### Conexión sin uso de un proxy

Si desea especificar ciertos hosts a los que se conecta el controlador sin utilizar un proxy, puede usar la propiedad `NonProxyHost` opcional en su cadena de conexión ODBC.

La propiedad `NonProxyHost` especifica una lista separada por comas de hosts a los que el conector puede acceder sin pasar por el servidor proxy cuando se habilita una conexión proxy, como en el siguiente ejemplo:

```
.amazonaws.com,localhost,.example.net,.example.com
```

El parámetro de conexión `NonProxyHost` se transfiere a la opción `curl CURLOPT_NOPROXY`. Para obtener información sobre el formato `CURLOPT_NOPROXY`, consulte [CURLOPT\\_NOPROXY](#) en la documentación de `curl`.

## Configuración del acceso federado a Amazon Athena para los usuarios de Microsoft AD FS mediante un cliente ODBC

Para configurar el acceso federado a Amazon Athena para los usuarios de los Servicios de federación de Active Directory (AD FS) mediante un cliente ODBC, primero debe establecer una relación de confianza entre AD FS y su cuenta de AWS. Con esta relación de confianza, los usuarios de AD pueden [federarse](#) en AWS con sus credenciales de AD y asumir los permisos de un rol de [AWS Identity and Access Management](#) (IAM) para acceder a recursos de AWS como la API de Athena.

Para crear esta relación de confianza, agregue AD FS como proveedor SAML a su Cuenta de AWS y cree un rol de IAM que los usuarios federados puedan asumir. En el lado de AD FS, se agrega AWS como parte de confianza y se escriben reglas de reclamación SAML para enviar los atributos de usuario correctos a AWS para su autorización (específicamente, Athena y Amazon S3).

La configuración del acceso de AD FS a Athena implica los siguientes pasos principales:

### [1. Configuración de un rol y un proveedor SAML de IAM](#)



## [2. Configuración de AD FS](#)

### [3. Creación de usuarios y grupos de Active Directory](#)

## [4. Configuración de la conexión ODBC de AD FS a Athena](#)

### 1. Configuración de un rol y un proveedor SAML de IAM

En esta sección, agregue AD FS como proveedor SAML a su cuenta de AWS y cree un rol de IAM que los usuarios federados puedan asumir.

Para configurar un proveedor SAML

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación, elija Proveedores de identidades.
3. Elija Add Provider (Agregar proveedor).
4. En Provider type (Tipo de proveedor), elija SAML.

The screenshot shows the AWS IAM console interface for creating a new identity provider. The left-hand navigation pane is visible, with 'Identity providers' highlighted in red. The main content area is titled 'Add an Identity provider' and 'Configure provider'. Under 'Provider type', the 'SAML' option is selected with a radio button. Below this, the 'Provider name' field is filled with 'adfs-saml-provider'. The 'Metadata document' section includes a 'Choose file' button and a green checkmark next to 'FederationMetadata.xml'.

5. En Provider name (Nombre de proveedor), ingrese **adfs-saml-provider**.
6. En un navegador, introduzca la siguiente dirección para descargar el archivo XML de federación para su servidor de AD FS. Para realizar este paso, el navegador debe tener acceso al servidor de AD FS.

<https://adfs-server-name/federationmetadata/2007-06/federationmetadata.xml>

7. En la consola de IAM, en Metadata document (Documento de metadatos), seleccione Choose file (Elegir archivo) y, a continuación, suba el archivo de metadatos de la federación a AWS.
8. Para terminar, seleccione Add provider (Agregar proveedor).

A continuación, cree el rol de IAM que los usuarios federados puedan asumir.

Para crear un rol de IAM para usuarios federados

1. En el panel de navegación de la consola de IAM, elija Roles.
2. Seleccione Crear rol.
3. En Trusted entity type (Tipo de entidad de confianza), elija SAML 2.0 federation (Federación SAML 2.0).
4. En SAML 2.0-based provider (Proveedor basado en SAML 2.0), elija el proveedor adfs-saml-provider que creó.
5. Elija Permitir acceso mediante programación y mediante la Consola de administración de AWS y, a continuación, elija Siguiente.

## Select trusted entity

### Trusted entity type

**AWS service**  
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

**AWS account**  
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

**SAML 2.0 federation**  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

**Custom trust policy**  
Create a custom trust policy to enable others to perform actions in this account.

### SAML 2.0 federation

Allow users federated with SAML 2.0 from a corporate directory to perform actions in this a

SAML 2.0–based provider

adfs-saml-provider ▼ ↻ Create n

Allow programmatic access only

Allow programmatic and AWS Management Console access


Attribute

6. En la página Add permissions (Agregar permisos), filtre las políticas de permisos de IAM que necesita para este rol y, a continuación, seleccione las casillas de verificación correspondientes. En este tutorial se adjuntan las políticas AmazonAthenaFullAccess y AmazonS3FullAccess.

## Add permissions [Info](#)


### Permissions policies

(Selected 1/838)




 [Create policy](#)

#### Info

Choose one or more policies to attach to your new role.

1 match < 1 > 

"AmazonAthenaFull"  [Clear filters](#)

<input checked="" type="checkbox"/>	Policy name 	Type	Description
<input checked="" type="checkbox"/>	  AmazonAthenaFullAccess	AWS managed	Provide full access to

### ▶ Set permissions boundary - optional [Info](#)

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

## Add permissions [Info](#)

**Permissions policies**  
(Selected 2/838)

[Info](#)  
Choose one or more policies to attach to your new role.

Q  1 match < 1 > [Settings](#)

"AmazonS3FullAccess" [X](#) [Clear filters](#)

<input checked="" type="checkbox"/>	Policy name <a href="#">↗</a>	Type	Description
<input checked="" type="checkbox"/>	<a href="#">+</a> <a href="#">📦</a> AmazonS3FullAccess	AWS managed	Provides full access

**▶ Set permissions boundary - optional** [Info](#)  
Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

[Cancel](#) [Previous](#) [Next](#)

7. Elija Siguiente.
8. En la página Name, review, and create (Asignar nombre, revisar y crear), en Role name (Nombre de rol), ingrese un nombre para el rol de IAM. Este tutorial usa el nombre adfs-data-access.

En Step 1: Select trusted entities (Paso 1: Seleccionar entidades de confianza), el campo Principal (Entidad principal) debe rellenarse automáticamente con "Federated:"  
"arn:aws:iam::*account\_id*:saml-provider/adfs-saml-provider". El campo Condition debe contener "SAML:aud" y "https://signin.aws.amazon.com/saml".

Step 1: Select trusted entities Edit

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "sts:AssumeRolewithSAML",
7       "Principal": {
8         "Federated": "arn:aws:iam::[redacted]:saml-provider/adfs-saml-provider"
9       },
10      "Condition": {
11        "StringEquals": {
12          "SAML:aud": [
13            "https://signin.aws.amazon.com/saml"
14          ]
15        }
16      }
17    }
18  ]
19 }

```

En Step 2: Add permissions (Paso 2: Agregar permisos), se muestran las políticas que ha adjuntado al rol.

Step 2: Add permissions Edit

Permissions policy summary

Policy name <a href="#">↗</a>	Type	Attached as
<a href="#">AmazonAthenaFullAccess</a>	AWS managed	Permissions policy
<a href="#">AmazonS3FullAccess</a>	AWS managed	Permissions policy

9. Seleccione Crear rol. Un mensaje de banner confirma la creación del rol.
10. En la página Roles, seleccione el nombre del rol que acaba de crear. La página de resumen del rol muestra las políticas que se han adjuntado.

IAM > Roles > adfs-data-access

# adfs-data-access

## Summary





Creation date	August 30, 2022, 16:33 (UTC-07:00)	ARN	arn:aws:iam
Last activity	1 hour ago	Maximum sessi	1 hour

**Permissions** | Trust relationships | Tags | Access Advisor | Revoke session

### Permissions policies (2)

You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter

<input type="checkbox"/>	Policy name <a href="#">↗</a>	Type
<input type="checkbox"/>	  AmazonS3FullAccess	AWS managed
<input type="checkbox"/>	  AmazonAthenaFullAccess	AWS managed

## 2. Configuración de AD FS

Ahora está listo para agregar AWS como parte de confianza y escribir reglas de reclamación de SAML para poder enviar los atributos de usuario correctos a AWS para su autorización.

La federación basada en SAML tiene dos partes participantes: el IdP (Active Directory) y la parte de confianza (AWS), que es el servicio o la aplicación que utiliza la autenticación del IdP.



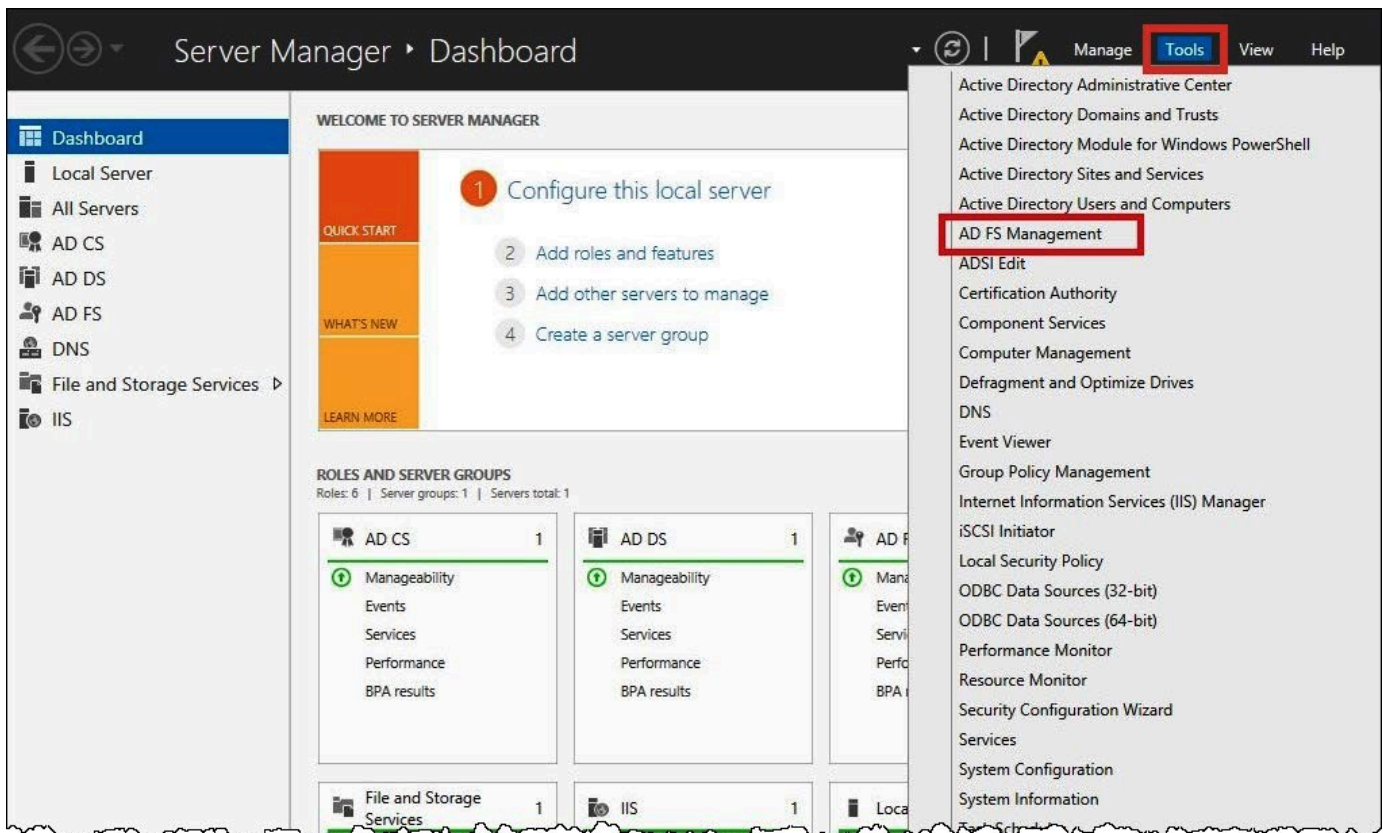
Para configurar AD FS, primero se agrega una parte de confianza y, a continuación, se configuran las reglas de reclamación de SAML para la parte de confianza. AD FS utiliza reglas de reclamación para formar una aserción SAML que se envía a una parte de confianza. La aserción SAML declara que la información sobre el usuario de AD es verdadera y que ha autenticado al usuario.

### Adición de una relación de confianza

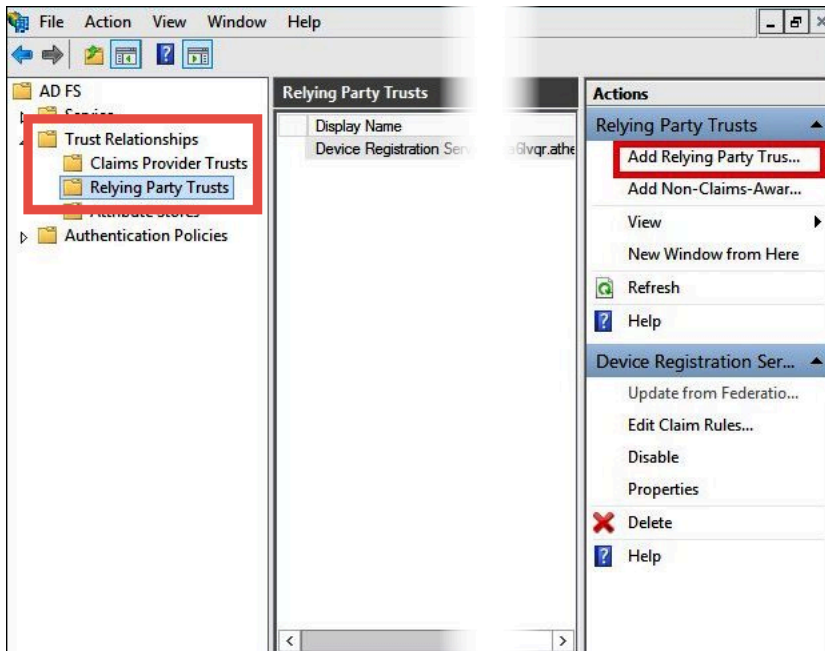
Para agregar una relación de confianza en AD FS, utilice el administrador de servidores de AD FS.

Para agregar una relación de confianza en AD FS

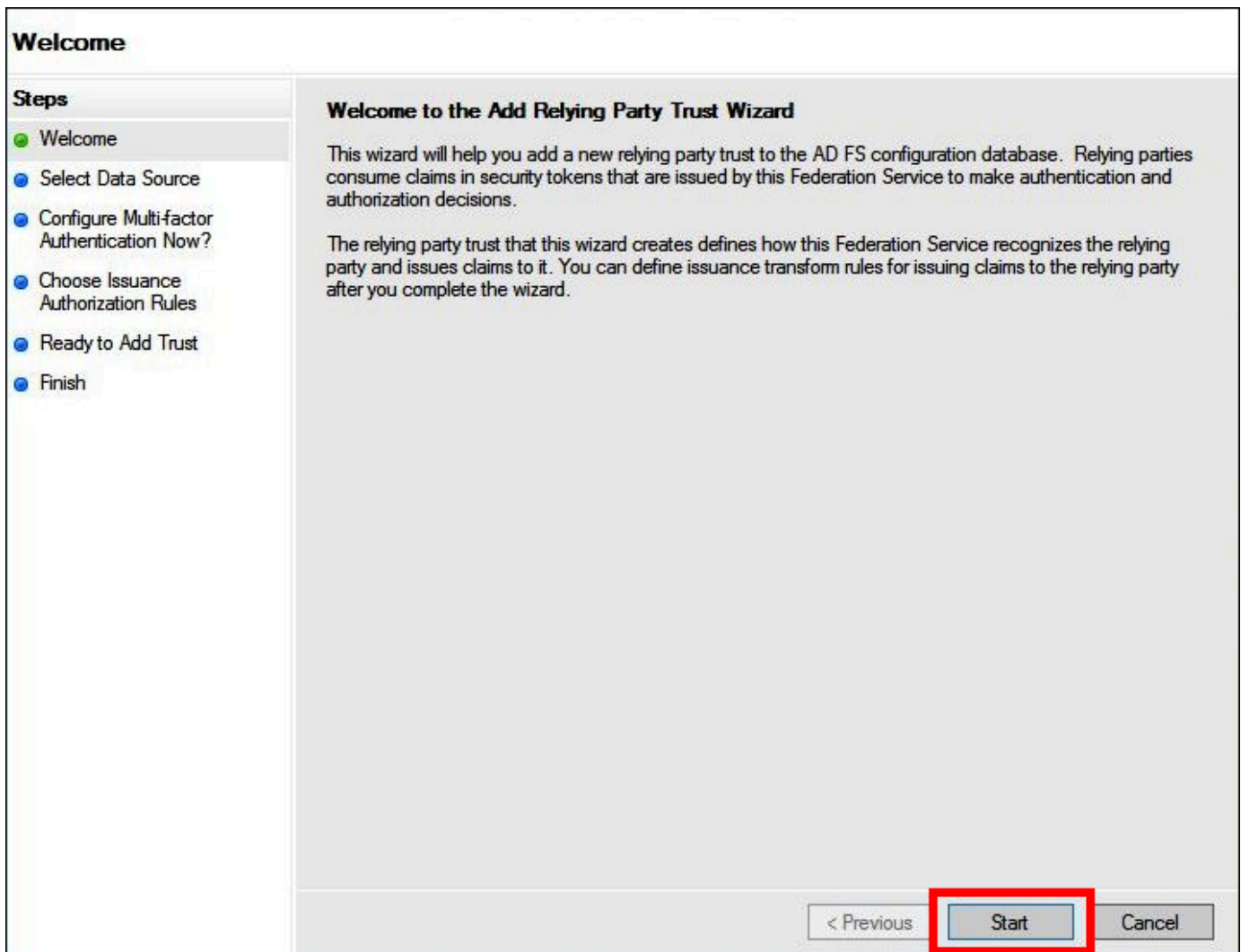
1. Inicie sesión en el servidor de AD FS.
2. En el menú Start (Inicio), abra Server Manager (Administrador de servidores).
3. Elija Tools (Herramientas) y, a continuación, elija AD FS Management (Administración de AD FS).



4. En el panel de navegación, en Trust Relationships (Relaciones de confianza), elija Relying Party Trusts (Relaciones de confianza para usuario autenticado).
5. En Actions (Acciones), seleccione Add Relying Party Trust (Agregar relación de confianza).



6. En la página Add Relying Party Trust Wizard (Asistente para agregar confianza de parte), elija Start (Inicio).



7. En la pantalla Select Data Source (Seleccionar origen de datos), elija Import data about the relying party published online or on a local network (Importar los datos sobre la parte de confianza publicados en línea o en una red local).
8. En Federation metadata address (host name or URL) (Dirección de metadatos de federación [nombre de host o URL]), introduzca la URL **`https://signin.aws.amazon.com/static/saml-metadata.xml`**.
9. Elija Siguiente.

### Select Data Source

**Steps**

- Welcome
- Select Data Source
- Configure Multi-factor Authentication Now?
- Choose Issuance Authorization Rules
- Ready to Add Trust
- Finish

Select an option that this wizard will use to obtain data about this relying party:

Import data about the relying party published online or on a local network

Use this option to import the necessary data and certificates from a relying party organization that publishes its federation metadata online or on a local network.

Federation metadata address (host name or URL):

Example: ts.contoso.com or https://www.contoso.com/app

Import data about the relying party from a file

Use this option to import the necessary data and certificates from a relying party organization that has exported its federation metadata to a file. Ensure that this file is from a trusted source. This wizard will not validate the source of the file.

Federation metadata file location:

Enter data about the relying party manually

Use this option to manually input the necessary data about this relying party organization.

< Previous 

10. En la página Specify Display Name (Especificar nombre para mostrar), en Display name (Nombre para mostrar), introduzca un nombre para la parte de confianza y, a continuación, elija Next (Siguiente).

**Specify Display Name**

Enter the display name and any optional notes for this relying party.

**Steps**

- Welcome
- Select Data Source
- Specify Display Name
- Configure Multi-factor Authentication Now?
- Choose Issuance Authorization Rules
- Ready to Add Trust
- Finish

Display name:  
signin.aws.amazon.com

Notes:

< Previous   **Next >**   Cancel

11. En la página Configure Multi-factor Authentication Now (Configurar la autenticación multifactor ahora), este tutorial selecciona I do not want to configure multi-factor authentication for this relying party trust at this time (No quiero configurar la autenticación multifactor para esta parte de confianza en este momento).

Para más seguridad, le recomendamos que configure la autenticación multifactor para ayudar a proteger sus recursos de AWS. Como utiliza un conjunto de datos de muestra, este tutorial no habilita la autenticación multifactor.

**Steps**

- Welcome
- Select Data Source
- Specify Display Name
- Configure Multi-factor Authentication Now?**
- Choose Issuance Authorization Rules
- Ready to Add Trust
- Finish

Configure multi-factor authentication settings for this relying party trust. Multi-factor authentication is required if there is a match for any of the specified requirements.

Multi-factor Authentication		Global Settings
Requirements	Users/Groups	Not configured
	Device	Not configured
	Location	Not configured

I do not want to configure multi-factor authentication settings for this relying party trust at this time.

Configure multi-factor authentication settings for this relying party trust.

You can also configure multi-factor authentication settings for this relying party trust by navigating to the Authentication Policies node. For more information, see [Configuring Authentication Policies](#).

< Previous   **Next >**   Cancel

12. Elija Siguiente.

13. En la página Choose Issuance Authorization Rules (Elegir reglas de autorización de emisión), seleccione Permit all users to access this relying party (Permitir que todos los usuarios tengan acceso a esta parte de confianza).

Esta opción permite que todos los usuarios de Active Directory usen AD FS con AWS como parte de confianza. Debe tener en cuenta sus requisitos de seguridad y ajustar esta configuración en consecuencia.

### Choose Issuance Authorization Rules

**Steps**

- Welcome
- Select Data Source
- Specify Display Name
- Configure Multi-factor Authentication Now?
- Choose Issuance Authorization Rules**
- Ready to Add Trust
- Finish

Issuance authorization rules determine whether a user is permitted to receive claims for the relying party. Choose one of the following options for the initial behavior of this relying party's issuance authorization rules.

Permit all users to access this relying party

The issuance authorization rules will be configured to permit all users to access this relying party. The relying party service or application may still deny the user access.

Deny all users access to this relying party

The issuance authorization rules will be configured to deny all users access to this relying party. You must later add issuance authorization rules to enable any users to access this relying party.

You can change the issuance authorization rules for this relying party trust by selecting the relying party trust and clicking Edit Claim Rules in the Actions pane.

< Previous   **Next >**   Cancel

14. Elija Siguiente.

15. En la página Ready to Add Trust (Listo para agregar confianza), elija Next (Siguiente) para agregar la parte de confianza a la base de datos de configuración de AD FS.

### Ready to Add Trust

**Steps**

- Welcome
- Select Data Source
- Specify Display Name
- Configure Multi-factor Authentication Now?
- Choose Issuance Authorization Rules
- Ready to Add Trust**
- Finish

The relying party trust has been configured. Review the following settings, and then click Next to add the relying party trust to the AD FS configuration database.

Monitoring | Identifiers | Encryption | Signature | Accepted Claims | Organization | Endpoints | Notes < >

Specify the monitoring settings for this relying party trust.

Relying party's federation metadata URL:

Monitor relying party

Automatically update relying party

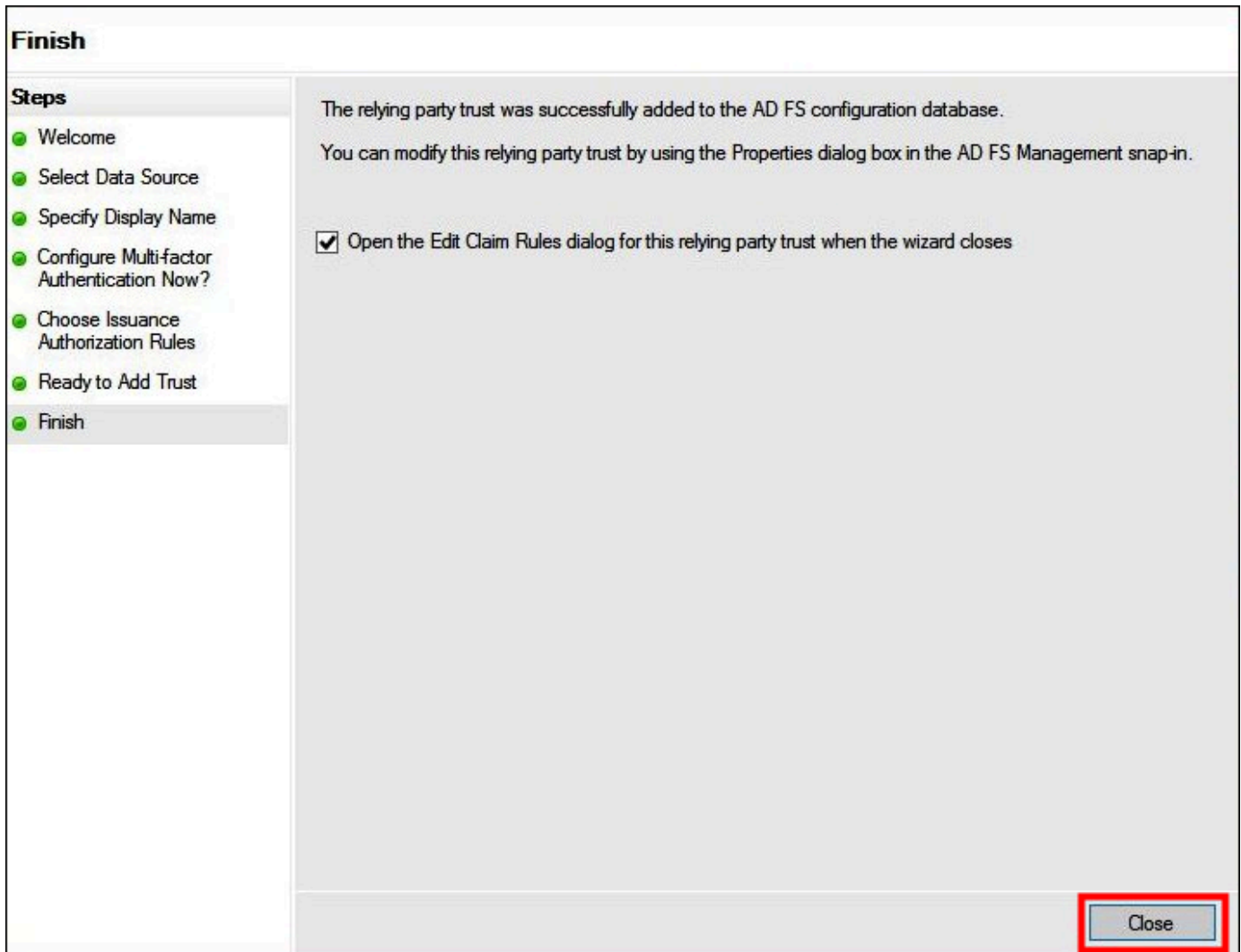
This relying party's federation metadata data was last checked on:  
9/1/2022

This relying party was last updated from federation metadata on:  
9/1/2022

< Previous **Next >** Cancel

16. En la página Finish (Finalizar), elija Close (Cerrar).





## Configuración de las reglas de reclamación SAML para la parte de confianza

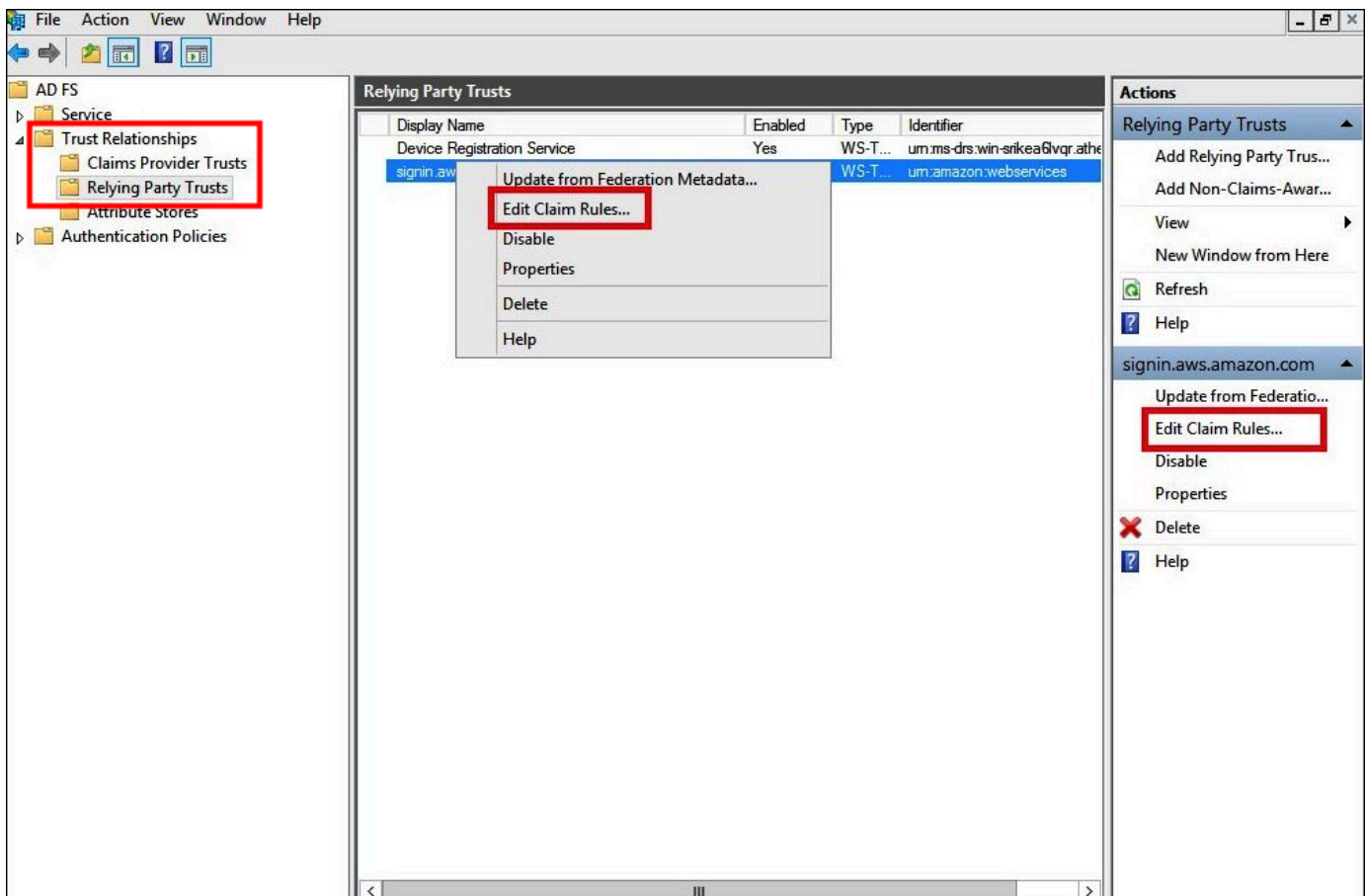
En esta tarea, creará dos conjuntos de reglas de reclamación.

El primer conjunto, de la regla 1 a la 4, contiene las reglas de reclamación de AD FS necesarias para asumir un rol de IAM basado en la pertenencia a un grupo de AD. Estas son las mismas reglas que se crean si se quiere establecer un acceso federado a la [AWS Management Console](#).

El segundo conjunto, reglas 5 y 6, son reglas de reclamación necesarias para el control de acceso de Athena.

## Para crear reglas de reclamación de AD FS

1. En el panel de navegación de la consola de administración de AD FS, elija Trust Relationships (Relaciones de confianza), Relying Party Trusts (Relaciones de confianza para usuario autenticado).
2. Busque la parte de confianza que creó en la sección anterior.
3. Haga clic con el botón derecho en la parte de confianza y seleccione Edit Claim Rules (Editar reglas de reclamación) o Edit Claim Rules (Editar reglas de reclamación) en el menú Actions (Acciones).



4. Seleccione Add Rule (Agregar regla).
5. En la página Configure Rule (Configurar regla) del asistente para agregar reglas de reclamación de transformación, introduzca la siguiente información para crear la regla de reclamación 1 y, a continuación, elija Finish (Finalizar).
  - En Claim Rule name (Nombre de regla de reclamación), escriba **NameID**.
  - En Rule template (Plantilla de reglas), utilice Transform an Incoming Claim (Transformar una reclamación entrante).

- En Incoming claim type (Tipo de reclamación entrante), elija Windows account name (Nombre de cuenta de Windows).
- En Outgoing claim type (Tipo de reclamación saliente), elija Name ID (ID de nombre).
- En Outgoing name ID format (Formato de ID de nombre saliente), elija Persistent Identifier (Identificador persistente).
- Elija Pass through all claim values (Acceso directo a todos los valores de reclamación).

**Configure Rule**

**Steps**

- Choose Rule Type
- Configure Claim Rule

You can configure this rule to map an incoming claim type to an outgoing claim type. As an option, you can also map an incoming claim value to an outgoing claim value. Specify the incoming claim type to map to the outgoing claim type and whether the claim value should be mapped to a new claim value.

Claim rule name:

Rule template: Transform an Incoming Claim

Incoming claim type:

Incoming name ID format:

Outgoing claim type:

Outgoing name ID format:

Pass through all claim values

Replace an incoming claim value with a different outgoing claim value

Incoming claim value:

Outgoing claim value:

Replace incoming e-mail suffix claims with a new e-mail suffix

New e-mail suffix:

Example: fabrikam.com

6. Elija Add Rule (Agregar regla) y, a continuación, introduzca la siguiente información para crear la regla de reclamación 2 y, finalmente, elija Finish (Finalizar).
  - En Claim rule name (Nombre de regla de reclamación), escriba **RoleSessionName**.
  - En Rule template (Plantilla de reglas), utilice Send LDAP Attribute as Claims (Enviar atributo LDAP como reclamaciones).

- En Attribute store (Almacén de atributos), elija Active Directory.
- En Mapping of LDAP attributes to outgoing claim types (Asignación de atributos LDAP a tipos de reclamación salientes), agregue el atributo **E-Mail-Addresses**. En Outgoing Claim Type (Tipo de reclamación saliente), introduzca **https://aws.amazon.com/SAML/Attributes/RoleSessionName**.

**Configure Rule**

**Steps**

- Choose Rule Type
- Configure Claim Rule

You can configure this rule to send the values of LDAP attributes as claims. Select an attribute store from which to extract LDAP attributes. Specify how the attributes will map to the outgoing claim types that will be issued from the rule.

Claim rule name:

Rule template: Send LDAP Attributes as Claims

Attribute store:

Mapping of LDAP attributes to outgoing claim types:

	LDAP Attribute (Select or type to add more)	Outgoing Claim Type (Select or type to add more)
▶	<input type="text" value="E-Mail-Addresses"/>	<input type="text" value="aws.amazon.com/SAML/Attributes/RoleSessionName"/>
*	<input type="text"/>	<input type="text"/>

< Previous   Finish   Cancel

7. Elija Add Rule (Agregar regla) y, a continuación, introduzca la siguiente información para crear la regla de reclamación 3 y, finalmente, elija Finish (Finalizar).

- En Claim rule name (Nombre de regla de reclamación), escriba **Get AD Groups**.
- En Rule template (Plantilla de reglas), utilice Send Claims Using a Custom Rule (Enviar reclamaciones utilizando una regla personalizada).
- En Custom rule (Regla personalizada), escriba el siguiente código:

```
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
  Issuer == "AD AUTHORITY"]=> add(store = "Active Directory", types = ("http://temp/variable"),
  query = ";tokenGroups;{0}", param = c.Value);
```

**Configure Rule**

**Steps**

- Choose Rule Type
- Configure Claim Rule

You can configure a custom claim rule, such as a rule that requires multiple incoming claims or that extracts claims from a SQL attribute store. To configure a custom rule, type one or more optional conditions and an issuance statement using the AD FS claim rule language.

Claim rule name:  
Get AD Groups

Rule template: Send Claims Using a Custom Rule

Custom rule:

```
c:[Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccount
name", Issuer == "AD AUTHORITY"]
=> add(store = "Active Directory", types = ("http://temp/variable"),
query = ";tokenGroups;{0}", param = c.Value);
```

< Previous   Finish   Cancel

8. Seleccione Add Rule (Agregar regla). Introduzca la siguiente información para crear la regla de reclamación 4 y, a continuación, seleccione Finish (Finalizar).
  - En Claim rule name (Nombre de regla de reclamación), escriba **Role**.
  - En Rule template (Plantilla de reglas), utilice Send Claims Using a Custom Rule (Enviar reclamaciones utilizando una regla personalizada).
  - En Custom rule (Regla personalizada), introduzca el siguiente código con su número de cuenta y el nombre del proveedor SAML que creó anteriormente:

```
c:[Type == "http://temp/variable", Value =~ "(?i)^aws-"]=> issue(Type = "https://aws.amazon.com/SAML/Attributes/Role", Value = RegExReplace(c.Value, "aws-", "arn:aws:iam::AWS_ACCOUNT_NUMBER:saml-provider/adfs-saml-provider,arn:aws:iam:: AWS_ACCOUNT_NUMBER:role/"));
```

### Configure Rule

**Steps**

- Choose Rule Type
- Configure Claim Rule

You can configure a custom claim rule, such as a rule that requires multiple incoming claims or that extracts claims from a SQL attribute store. To configure a custom rule, type one or more optional conditions and an issuance statement using the AD FS claim rule language.

Claim rule name:

Rule template: Send Claims Using a Custom Rule

Custom rule:

```
c:[Type == "http://temp/variable", Value =~ "(?i)^aws-"]=> issue(Type = "https://aws.amazon.com/SAML/Attributes/Role", Value = RegExReplace(c.Value, "aws-", "arn:aws:iam::123456789012:saml-provider/adfs-saml-provider,arn:aws:iam::123456789012:role/"));
```

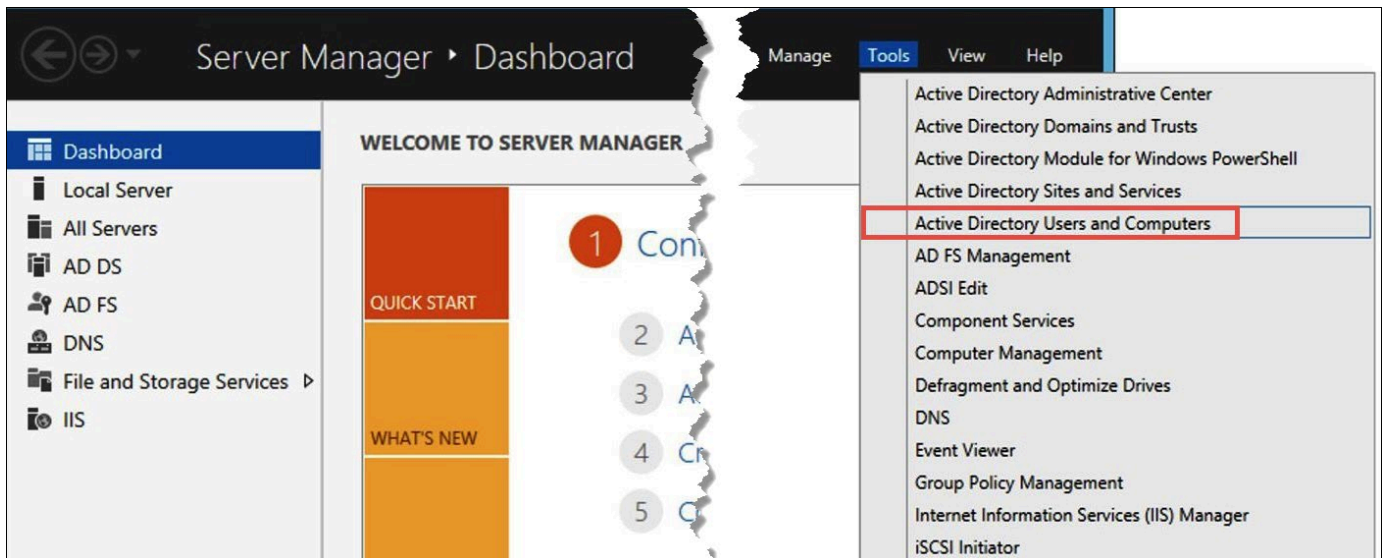
< Previous   Finish   Cancel

### 3. Creación de usuarios y grupos de Active Directory

Ahora está listo para crear usuarios de AD que accederán a Athena, y grupos de AD en los que colocarlos para poder controlar los niveles de acceso por grupo. Después de crear grupos de AD que clasifiquen los patrones de acceso a los datos, agregue sus usuarios a esos grupos.

## Para crear usuarios de AD para acceder a Athena

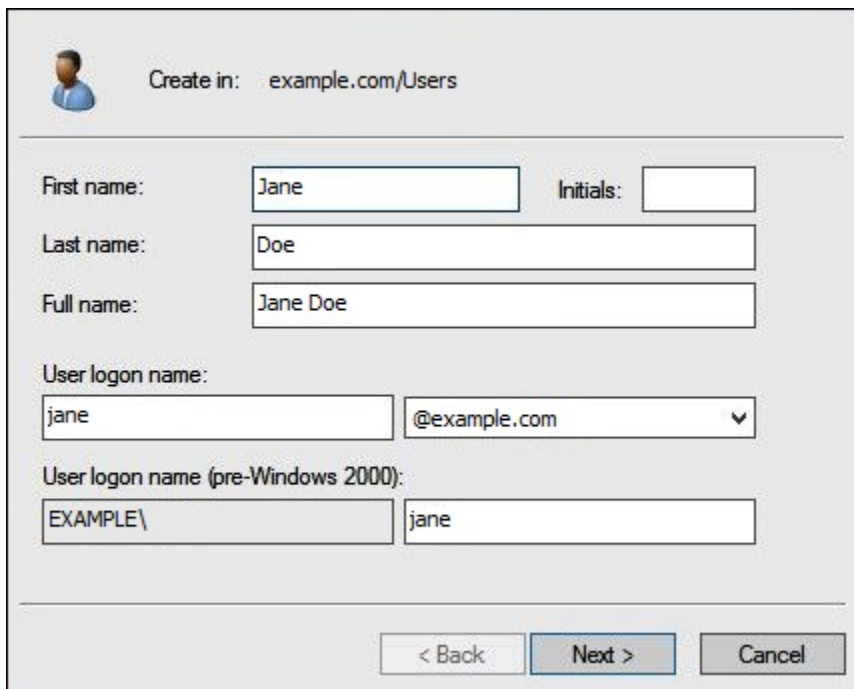
1. En el panel del administrador de servidores, seleccione Tools (Herramientas) y, a continuación, Active Directory Users and Computers (Usuarios y equipos de Active Directory).



2. En el panel de navegación, seleccione Usuarios.
3. En la barra de herramientas Active Directory Users and Computers (Usuarios y equipos de Active Directory), elija la opción Create user (Crear usuario).



4. En el cuadro de diálogo New Object - User (Nuevo objeto: usuario), rellene First name (Nombre), Last name (Apellidos) y Full name (Nombre completo). En este tutorial se utiliza un **Jane Doe**.



Create in: example.com/Users

First name: Jane Initials:

Last name: Doe

Full name: Jane Doe

User logon name: jane @example.com

User logon name (pre-Windows 2000): EXAMPLE\ jane

< Back Next > Cancel

5. Elija Siguiente.
6. En Password (Contraseña), introduzca una contraseña y, a continuación, vuelva a escribirla para confirmarla.

Para simplificar, este tutorial anula la selección de User must change password at next sign on (El usuario debe cambiar la contraseña la próxima vez que inicie sesión). En situaciones reales, debe exigir a los usuarios recién creados que cambien su contraseña.



Create in: example.com/Users

Password:

Confirm password:

User must change password at next logon

User cannot change password

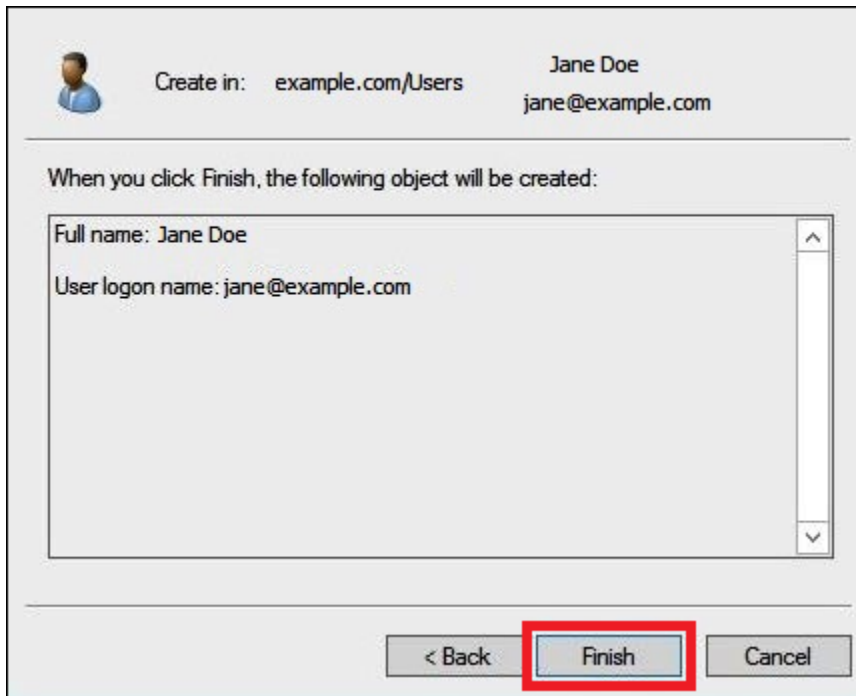
Password never expires

Account is disabled

< Back Next > Cancel



7. Elija Siguiente.
8. Seleccione Finalizar.



9. En Active Directory Users and Computers (Equipos y usuarios de Active Directory), elija el nombre de usuario.
10. En el cuadro de diálogo Properties (Propiedades) del usuario, introduzca una dirección de correo electrónico en E-mail (Correo electrónico). En este tutorial se utiliza un **jane@example.com**.

The image shows a Windows-style dialog box for editing a user profile. At the top, there are several tabs: 'Member Of', 'Dial-in', 'Environment', 'Sessions', 'Remote control', 'Remote Desktop Services Profile', and 'COM+'. The 'General' tab is selected. Below the tabs, there is a header area with a small profile picture and the name 'Jane Doe'. The main area contains several input fields: 'First name:' with 'Jane' and 'Initials:' with an empty field; 'Last name:' with 'Doe'; 'Display name:' with 'Jane Doe'; 'Description:' with an empty field; 'Office:' with an empty field; 'Telephone number:' with an empty field and an 'Other...' button; 'E-mail:' with 'jane@example.com'; and 'Web page:' with an empty field and an 'Other...' button. At the bottom, there are four buttons: 'OK', 'Cancel', 'Apply', and 'Help'.

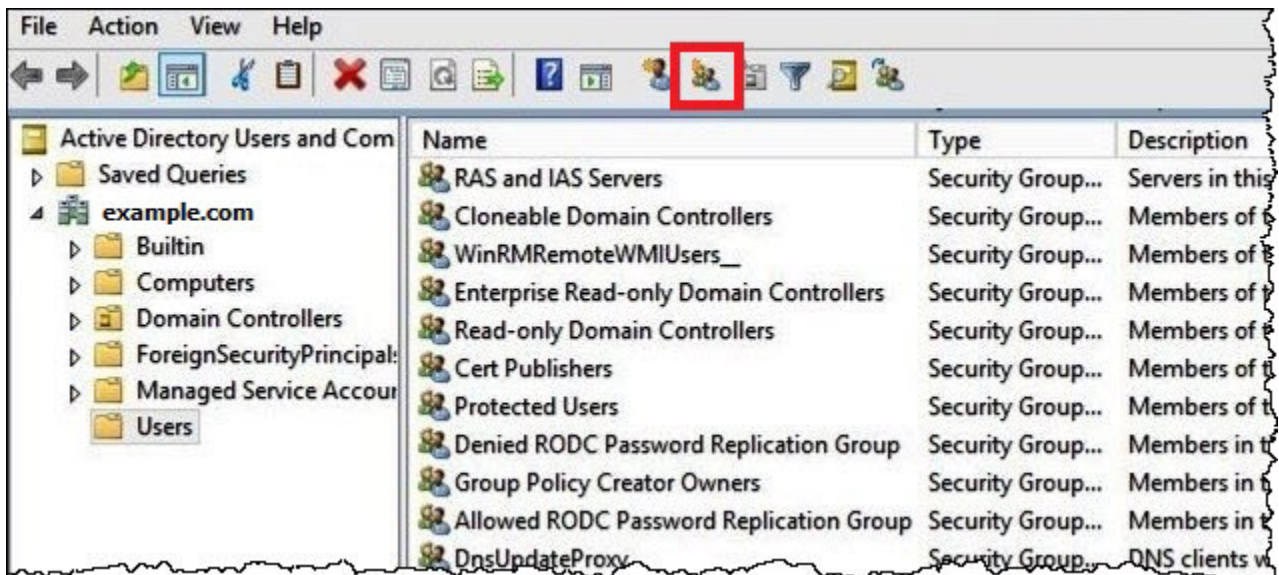
11. Seleccione Aceptar.

Creación de grupos de AD para representar patrones de acceso a los datos

Puede crear grupos de AD cuyos miembros asuman el rol de IAM `adfs-data-access` al iniciar sesión en AWS. En el siguiente ejemplo, se crea un grupo de AD denominado `aws-adfs-data-access`.

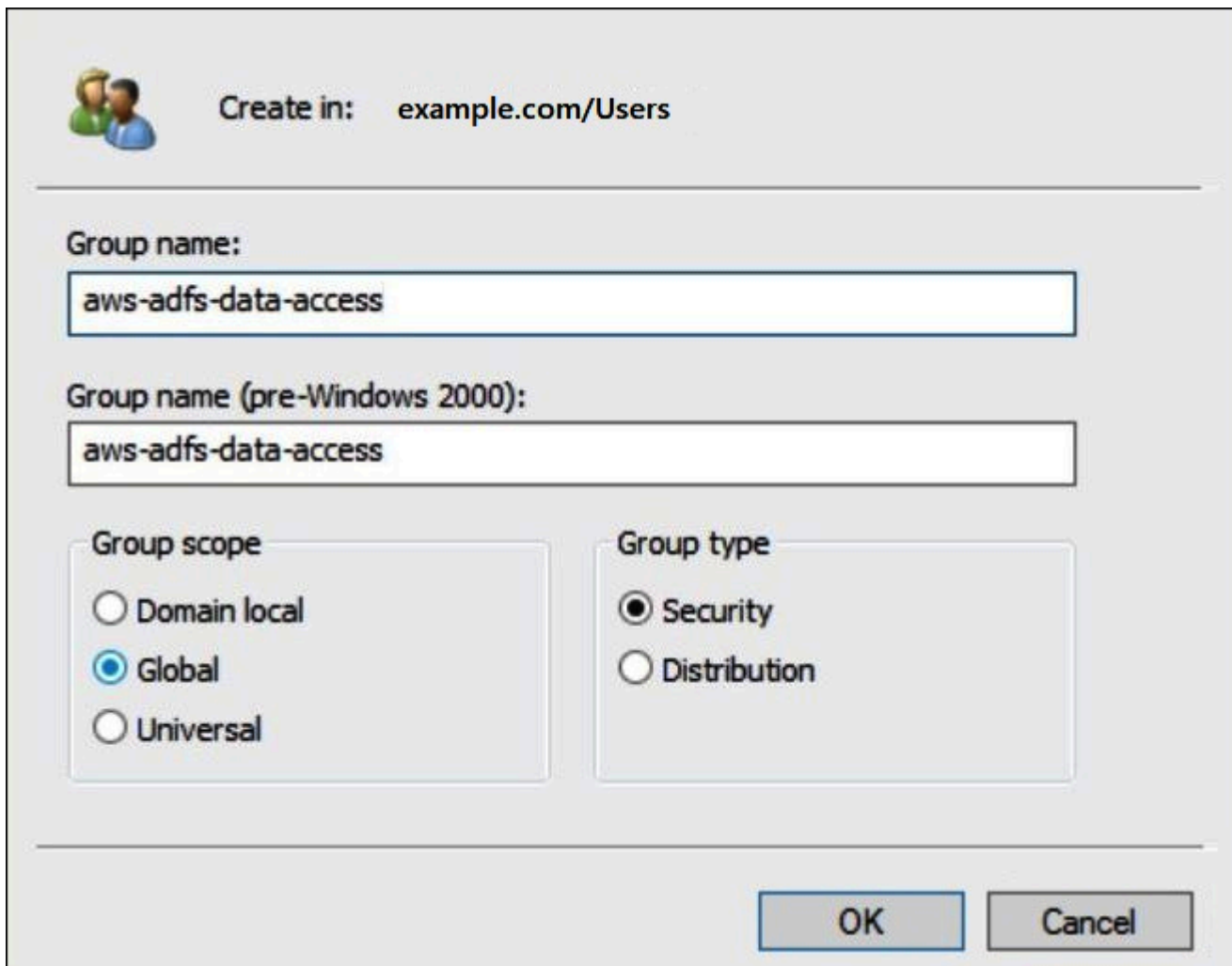
Para crear un grupo de AD

1. En el panel del administrador de servidores, en el menú Tools (Herramientas), elija Active Directory Users and Computers (Equipos y usuarios de Active Directory).
2. En la barra de herramientas, seleccione la opción Create new group (Crear nuevo grupo).



3. En el cuadro de diálogo New Object - Group (Nuevo objeto: grupo), introduzca la siguiente información:

- En Group Name (Nombre de grupo), introduzca **aws-ads-data-access**.
- En Group scope (Ámbito del grupo), seleccione Global.
- En Group type (Tipo de grupo), seleccione Security (Seguridad).



Create in: example.com/Users

Group name:  
aws-adfs-data-access

Group name (pre-Windows 2000):  
aws-adfs-data-access

Group scope

- Domain local
- Global
- Universal

Group type

- Security
- Distribution

OK Cancel

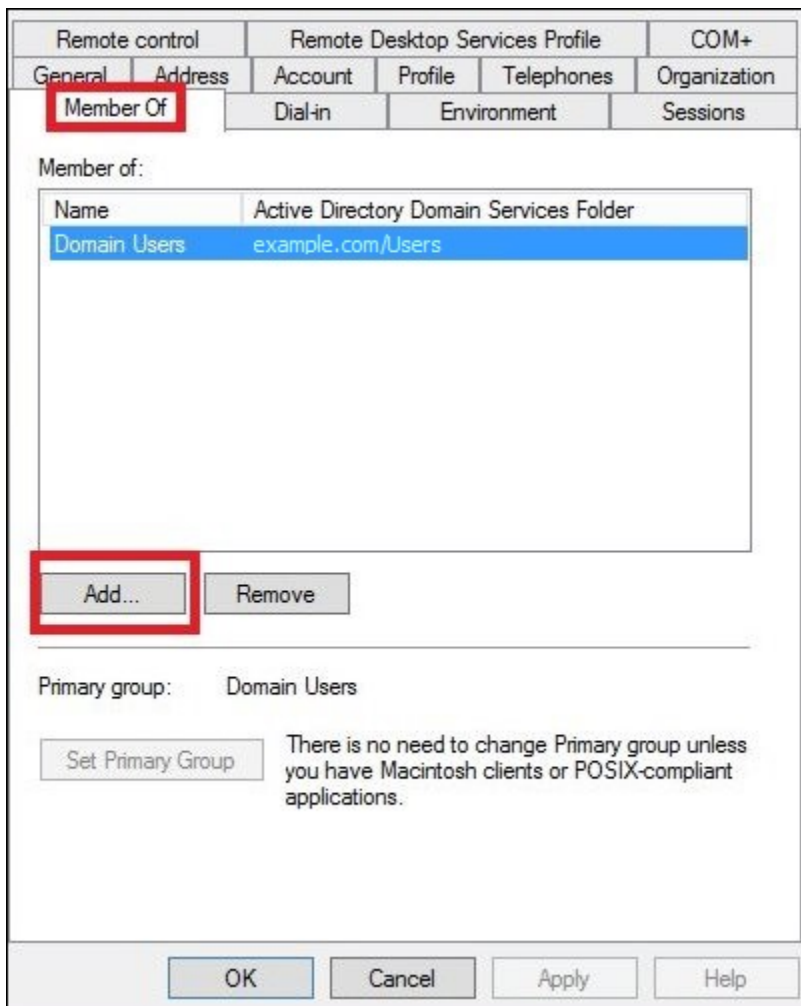
4. Seleccione Aceptar.

#### Adición de usuarios de AD a los grupos adecuados

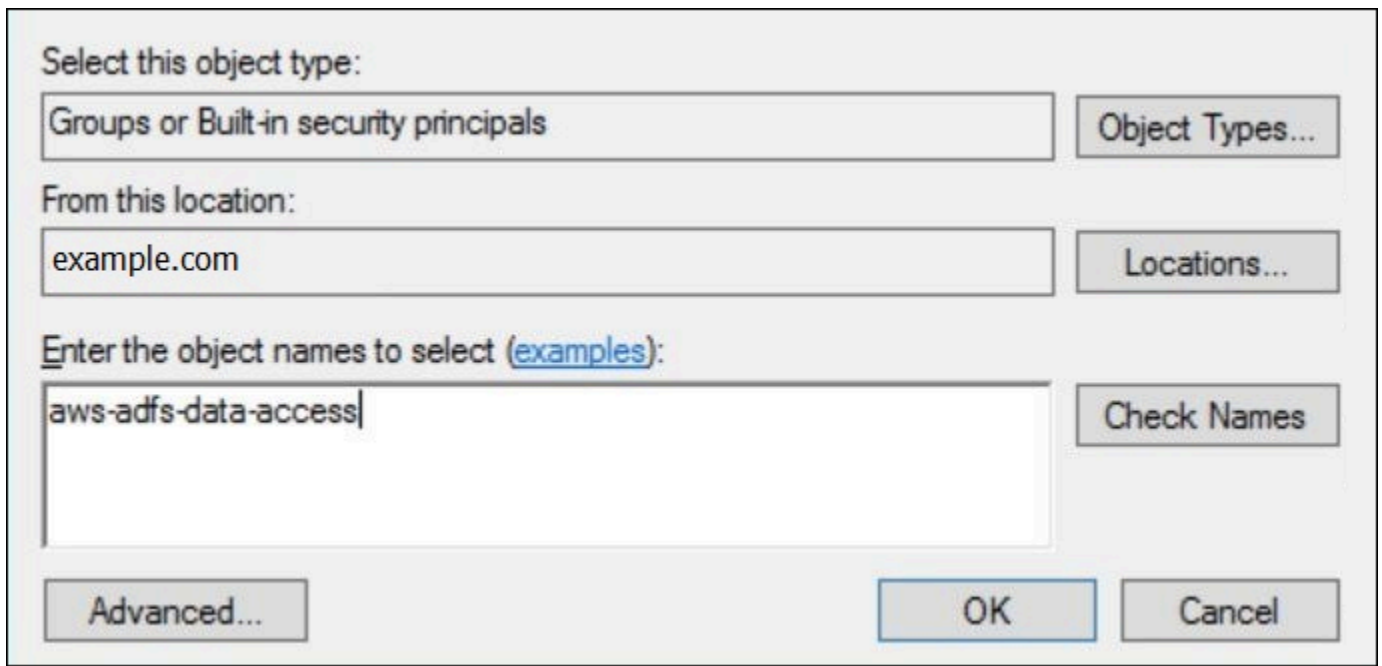
Ahora que ha creado un usuario de AD y un grupo de AD, puede agregar el usuario al grupo.

#### Para agregar un usuario de AD a un grupo de AD

1. En el panel del administrador de servidores, en el menú Tools (Herramientas), elija Active Directory Users and Computers (Equipos y usuarios de Active Directory).
2. En First name (Nombre) y Last name (Apellidos), elija un usuario (por ejemplo, Jane Doe).
3. En el cuadro de diálogo Properties (Propiedades) del usuario, en la pestaña Member Of (Miembro de), seleccione Add (Agregar).



4. Agregue uno o más grupos de AD FS según sus necesidades. Este tutorial agrega el grupo `aws-ads-data-access`.
5. En el cuadro de diálogo Select Groups (Seleccionar grupos), en Enter the object names to select (Introducir los nombres de los objetos que se deseen seleccionar), introduzca el nombre del grupo de AD FS que creó (por ejemplo, **aws-ads-data-access**) y, a continuación, seleccione Check Names (Comprobar nombres).

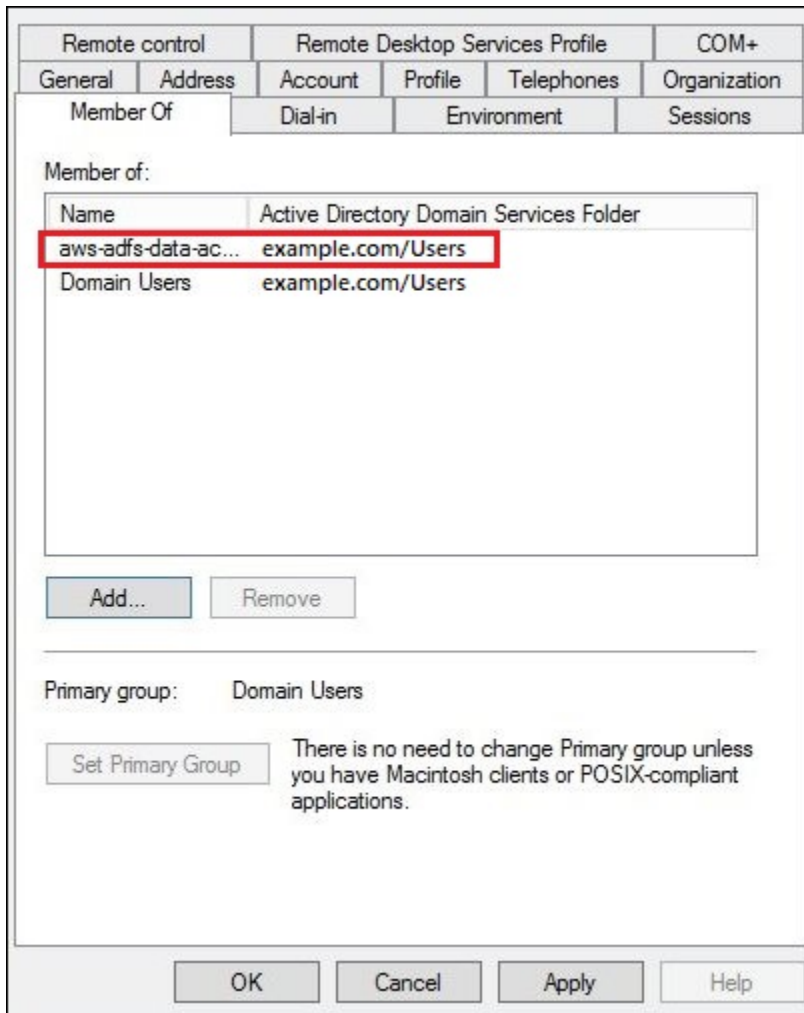


The image shows a dialog box with the following fields and buttons:

- Select this object type:** A text box containing "Groups or Built-in security principals" and a button labeled "Object Types...".
- From this location:** A text box containing "example.com" and a button labeled "Locations...".
- Enter the object names to select (examples):** A text box containing "aws-adfs-data-access" and a button labeled "Check Names".
- At the bottom, there are three buttons: "Advanced...", "OK", and "Cancel".

6. Seleccione Aceptar.

En el cuadro de diálogo Properties (Propiedades) del usuario, el nombre del grupo de AD aparece en la lista Member of (Miembro de).



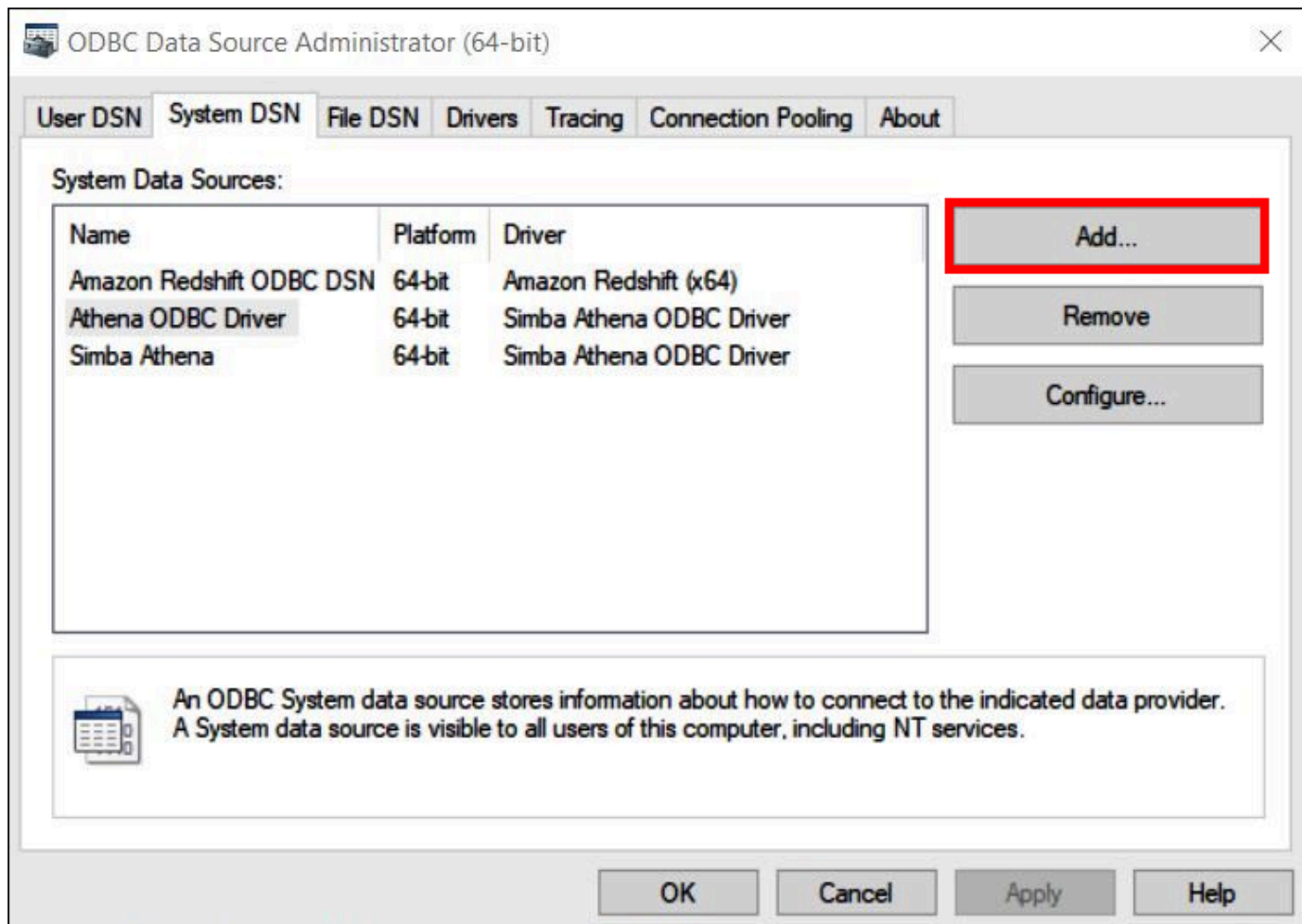
7. Elija Apply (Aplicar) y, después, OK (Aceptar).

#### 4. Configuración de la conexión ODBC de AD FS a Athena

Una vez que haya creado los usuarios y grupos de AD, estará listo para usar el programa de orígenes de datos ODBC en Windows para configurar la conexión ODBC a Athena de AD FS.

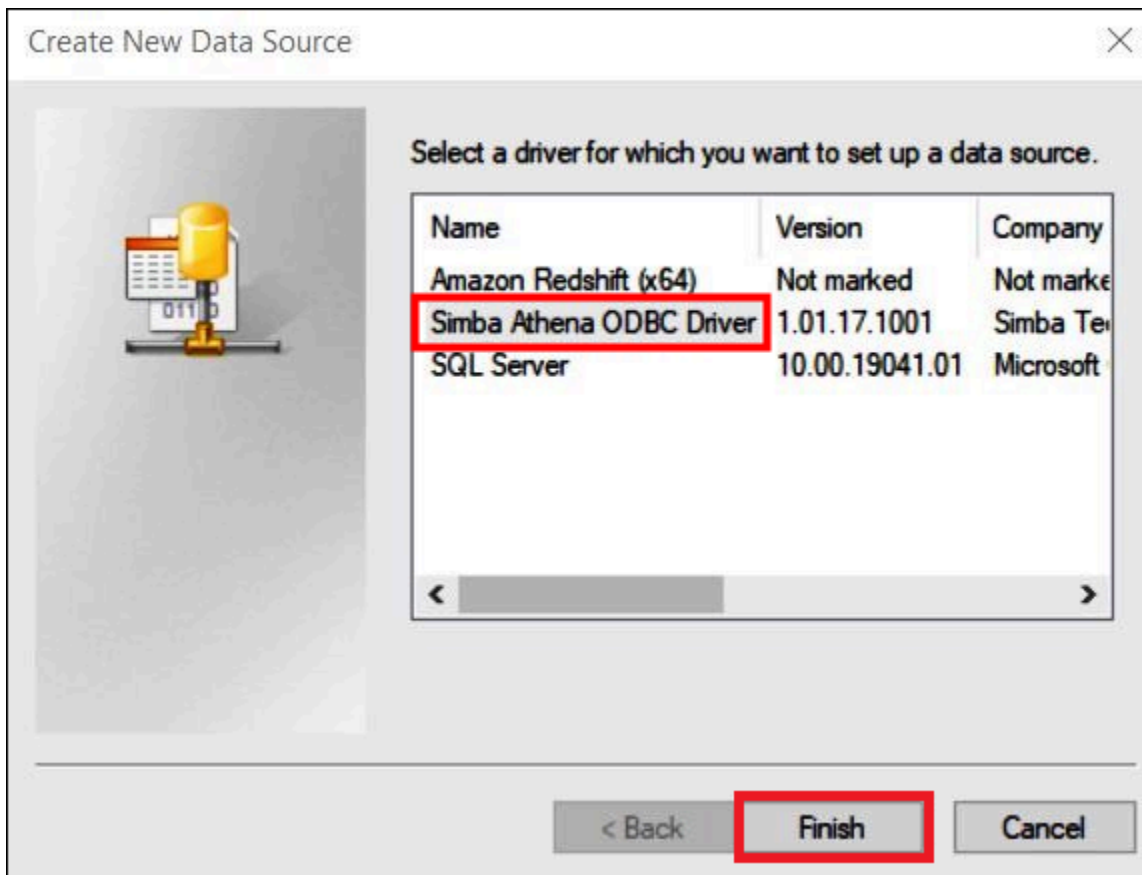
Para configurar la conexión ODBC de AD FS a Athena

1. Instale el controlador ODBC para Athena. Para obtener enlaces de descarga, consulte [Conexión a Amazon Athena con ODBC](#).
2. En Windows, elija Start (Inicio), ODBC Data Sources (Orígenes de datos ODBC).
3. En el programa ODBC Data Source Administrator (Administrador de orígenes de datos ODBC), elija Add (Agregar).



4. En el cuadro de diálogo Create New Data Source (Crear nuevo origen de datos), elija Simba Athena ODBC Driver (Controlador ODBC Simba Athena) y, a continuación, elija Finish (Finalizar).





5. En el cuadro de diálogo Simba Athena ODBC Driver DSN Setup (Configuración de DSN del controlador ODBC Simba Athena), introduzca los siguientes valores:
- En Data Source Name (Nombre de origen de datos), ingrese un nombre para el origen de datos (por ejemplo, **Athena-odbc-test**).
  - En Description (Descripción), ingrese una descripción para el origen de datos.
  - En Región de AWS, ingrese la Región de AWS que usa (por ejemplo, **us-west-1**).
  - En S3 Output Location (Ubicación de salida de S3), ingrese la ruta de Amazon S3 en la que desea que se almacenen los datos de salida.

Simba Athena ODBC Driver DSN Setup

Data Source Name: Athena-odbc-test

Description:

AWS Region: us-west-1

Catalog: AwsDataCatalog

Schema: default

Workgroup: odbc-test-group

Metadata Retrieval Method: Auto

Output Options

S3 Output Location: s3://DOC-EXAMPLE-BUCKET/

Encryption Options: NOT\_SET

KMS Key:

Endpoint Override:

Streaming Endpoint Override:

Authentication Options... Advanced Options... Logging Options... Proxy Options...

v1.1.17.1001 (64 bit) Test... OK Cancel

6. Elija Authentication Options (Opciones de autenticación).
7. En el cuadro de diálogo Authentication Options (Opciones de autenticación), indique los siguientes valores:
  - En Authentication Type (Tipo de autenticación), elija ADFS.
  - En User (Usuario), introduzca la dirección de correo electrónico del usuario (por ejemplo, **jane@example.com**).
  - En Password (Contraseña), introduzca la contraseña del usuario de ADFS.
  - En Host IdP (Host de IdP), introduzca el nombre del servidor de AD FS (por ejemplo, **adfs.example.com**).
  - En IdP Port (Puerto de IdP), utilice el valor predeterminado 443.
  - Seleccione la opción SSL Insecure (SSL inseguro).

Authentication Type: ADFS

User: jane@example.com

Password: [Redacted]

Session Token: [Empty]

Preferred Role: [Empty]

Session Duration: [Empty]

IdP Host: adfs.example.com

IdP Port: 443

Use HTTP Proxy For IdP Host       SSL Insecure

OK      Cancel

8. Elija OK (Aceptar) para cerrar las Authentication Options (Opciones de autenticación).
9. Elija Test (Probar) para probar la conexión o OK (Aceptar) para finalizar.

## Configuración de SSO para ODBC mediante el complemento Okta y el proveedor de identidad Okta

Esta página muestra cómo configurar el controlador ODBC de Amazon Athena y el complemento Okta para agregar el módulo de inicio de sesión único (SSO) con el proveedor de identidad Okta.

### Requisitos previos

Para completar los pasos de este tutorial, se necesitan las siguientes versiones:

- Controlador ODBC de Amazon Athena. Para obtener enlaces de descarga, consulte [Conexión a Amazon Athena con ODBC](#).
- Un rol de IAM que quiera usar con SAML. Para obtener más información, consulte [Creación de un rol para una federación SAML 2.0](#) en la Guía del usuario de IAM.
- Una cuenta de Okta. Para obtener información, visite [Okta.com](https://Okta.com).

### Creación de una integración de aplicaciones en Okta

En primer lugar, use el panel de control de Okta para crear y configurar una aplicación SAML 2.0 para el inicio de sesión único en Athena. Puede utilizar una aplicación de Redshift existente en Okta para configurar el acceso a Athena.

Para crear una integración de aplicaciones en Okta

1. Inicie sesión en la página de administración de su cuenta en [Okta.com](https://Okta.com).
2. En el panel de navegación, elija Applications (Aplicaciones), Applications.
3. En la página Applications, elija Browse App Catalog (Examinar catálogo de aplicaciones).
4. En la página Browse App Integration Catalog (Examinar catálogo de integración de aplicaciones), en la sección Use Case (Caso de uso), elija All Integrations (Todas las integraciones).
5. En el cuadro de búsqueda, ingrese Amazon Web Services Redshift y luego elija Amazon Web Services Redshift SAML (SAML de Amazon Web Services Redshift).
6. Seleccione Add Integration (Agregar integración).

Dashboard ▾

Directory ▾

Customizations ▾

Applications ▲

Applications

Self Service

Security ▾

Workflow ▾

Reports ▾

Settings ▾

Applications > Catalog > Single Sign-On > Amazon Web Services Redshift

Last updated: August 27, 2019

**Add Integration**

**Amazon Web Services Redshift**

SAML

**Okta Verified**

The integration was either created by Okta or by

**Overview**

Okta's integration with Amazon Web Services (AWS) Redshift allows end users to authenticate to AWS

7. En la sección General Settings Required (Configuración general necesaria), para Application label (Etiqueta de aplicación), ingrese un nombre para la aplicación. Este tutorial utiliza el nombre Athena-ODBC-Okta.

# Add Amazon Web Services Redshift

**1** General Settings

## General settings- Required

Application label

This label displays under the app on your home page


Application Visibility

- Do not display application icon to users
- Do not display application icon in the Okta Mobile App


**Cancel** **Done**

8. Seleccione Listo.
9. En la página de su solicitud de Okta (por ejemplo, Atena-ODBC-Okta), elija Sign On (Iniciar sesión).

← Back to Applications



# Athena-ODBC-Okta

**Active**  [View Logs](#) [Monitor Imports](#)

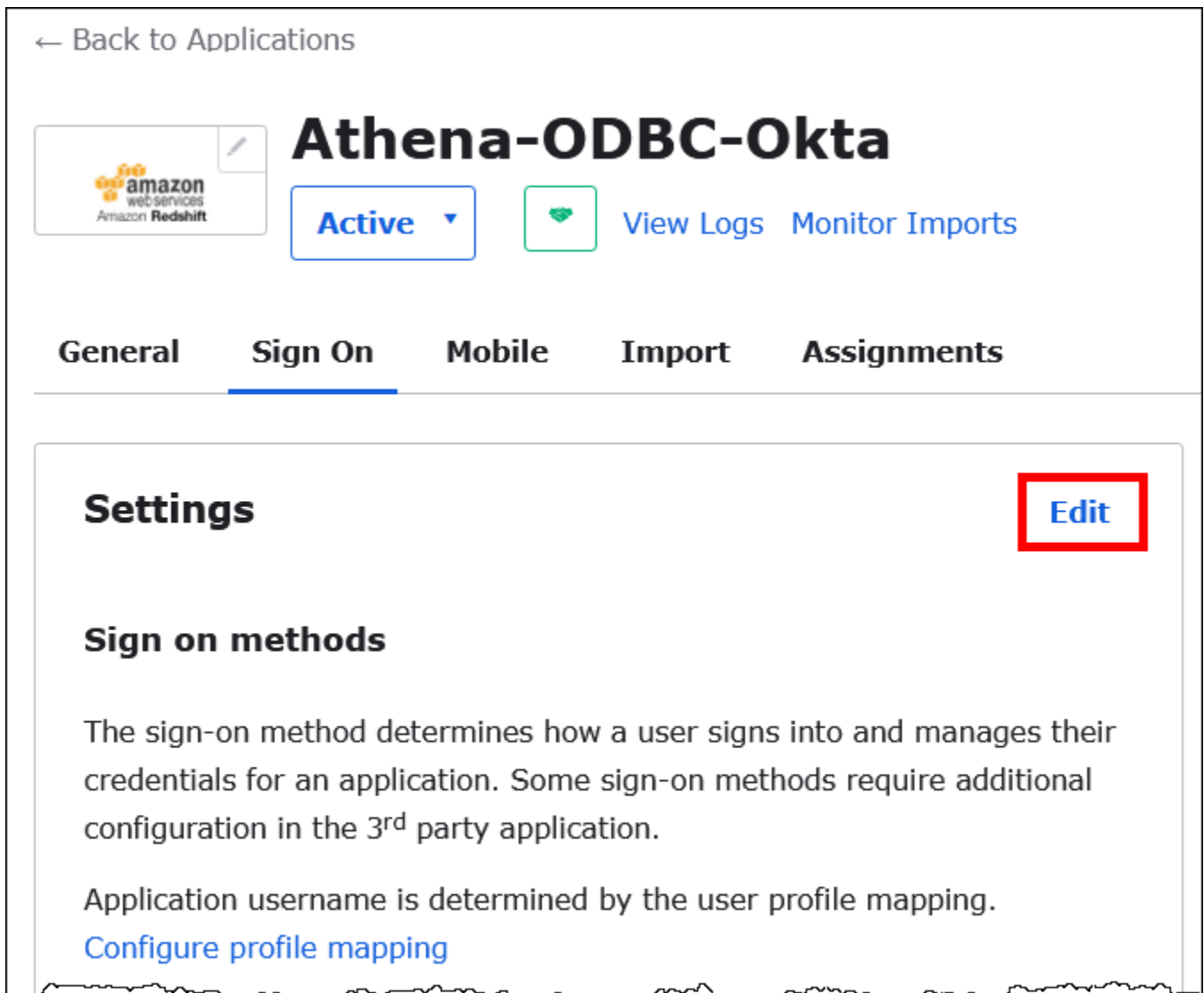
**General** **Sign On** **Mobile** **Import** **Assignments**

**Assign** **Convert assignments**


Search... **People**


Filters	Person	Type
People		
Groups		01101110 01101111 01101100 01101000 01101001 01101110 01100111
		No users found

10. En la sección Settings (Configuración), elija Edit (Editar).



← Back to Applications

 **Athena-ODBC-Okta**

Active  [View Logs](#) [Monitor Imports](#)

**General** **Sign On** **Mobile** **Import** **Assignments**

**Settings** [Edit](#)

**Sign on methods**

The sign-on method determines how a user signs into and manages their credentials for an application. Some sign-on methods require additional configuration in the 3<sup>rd</sup> party application.

Application username is determined by the user profile mapping.

[Configure profile mapping](#)

11. En la sección Advanced Sign-on Settings (Configuración avanzada de inicio de sesión), configure los siguientes valores.
  - En IdP ARN and Role ARN (ARN de IdP y ARN de rol), ingrese su ARN de IdP y ARN de rol de AWS como valores separados por comas. Para obtener información sobre el formato del rol de IAM, consulte [Configuración de aserciones SAML para la respuesta de autenticación](#) en la Guía del usuario de IAM..
  - En Session Duration (Duración de la sesión), indique un valor entre 900 y 43 200 segundos. En este tutorial, se usa el valor predeterminado de 3600 (1 hora).



## Advanced Sign-on Settings

These fields may be required for a Amazon Web Services Redshift proprietary sign-on option or general setting.

Idp ARN and Role ARN

arn:aws:iam::1234567890:saml-provid

Enter your AWS IDP ARN and Role ARN as comma separated values (e.g. "arn:aws:iam::1234567890:saml-provider/OKTA,arn:aws:iam::1234567890:role/SAML\_ROLE").

Session Duration

3600

Set the user's session duration in seconds here.

Valid range is 900 to 43200.

DB User Format (Redshift)

\${user.username}

EL expression to get DB User value (e.g. "\${user.username}", "\${user.firstName}\${user.lastName}@acme.com")

Auto Create (Redshift)



AutoCreate Redshift property (Create a new database user if one does not exist)

Allowed DB Groups (Redshift)

Comma separated list of allowed user groups. Use "\*" to allow all groups, "\" to escape comma in group name

Athena no utiliza las configuraciones DbUser Format (Formato DbUser), AutoCreate (Creación automática) y Allowed DBGroups (DBGroups permitidos). No es preciso configurarlas.

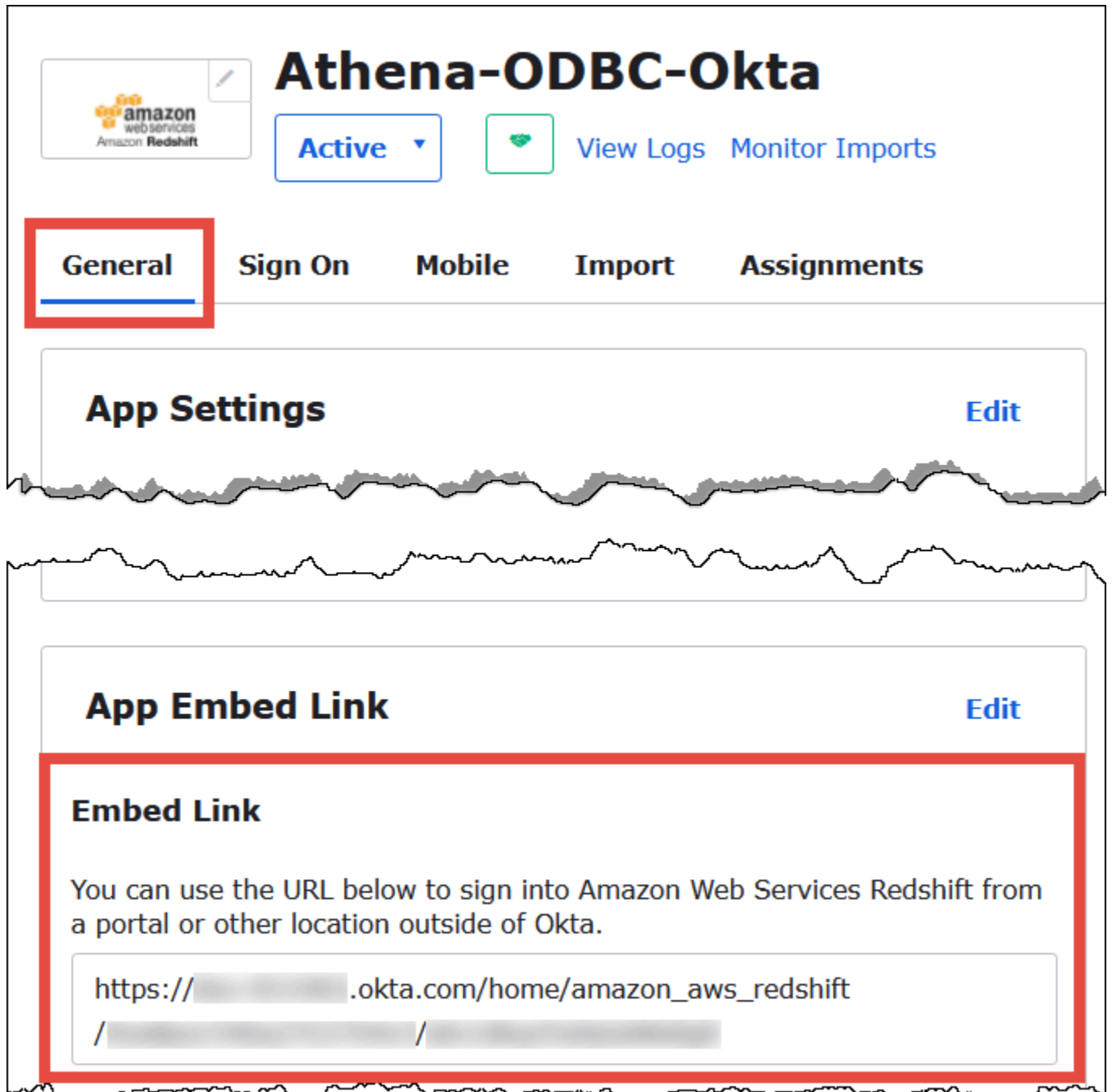
12. Elija Guardar.

## Recuperación de información de configuración de ODBC desde Okta

Ahora que ya creó la aplicación Okta, tiene todo listo para recuperar el ID de la aplicación y la URL del host del IdP. Los necesitará más adelante cuando configure ODBC para la conexión a Athena.

Para recuperar la información de configuración de ODBC desde Okta

1. Elija la pestaña General (General) de la aplicación Okta y desplácese hacia abajo hasta la sección App Embed Link (Enlace de inserción de aplicación).



**Athena-ODBC-Okta**

Active View Logs Monitor Imports

**General** Sign On Mobile Import Assignments

**App Settings** [Edit](#)

**App Embed Link** [Edit](#)

**Embed Link**

You can use the URL below to sign into Amazon Web Services Redshift from a portal or other location outside of Okta.

`https://[redacted].okta.com/home/amazon_aws_redshift/[redacted]/[redacted]`

La URL de Embed Link (Enlace de inserción) tiene el siguiente formato:

```
https://trial-1234567.okta.com/home/amazon_aws_redshift/Abc1de2fghi3J45kL678/abc1defghij2klmNo3p4
```

- Desde la URL de Embed Link (Enlace de inserción), extraiga y guarde las siguientes piezas:

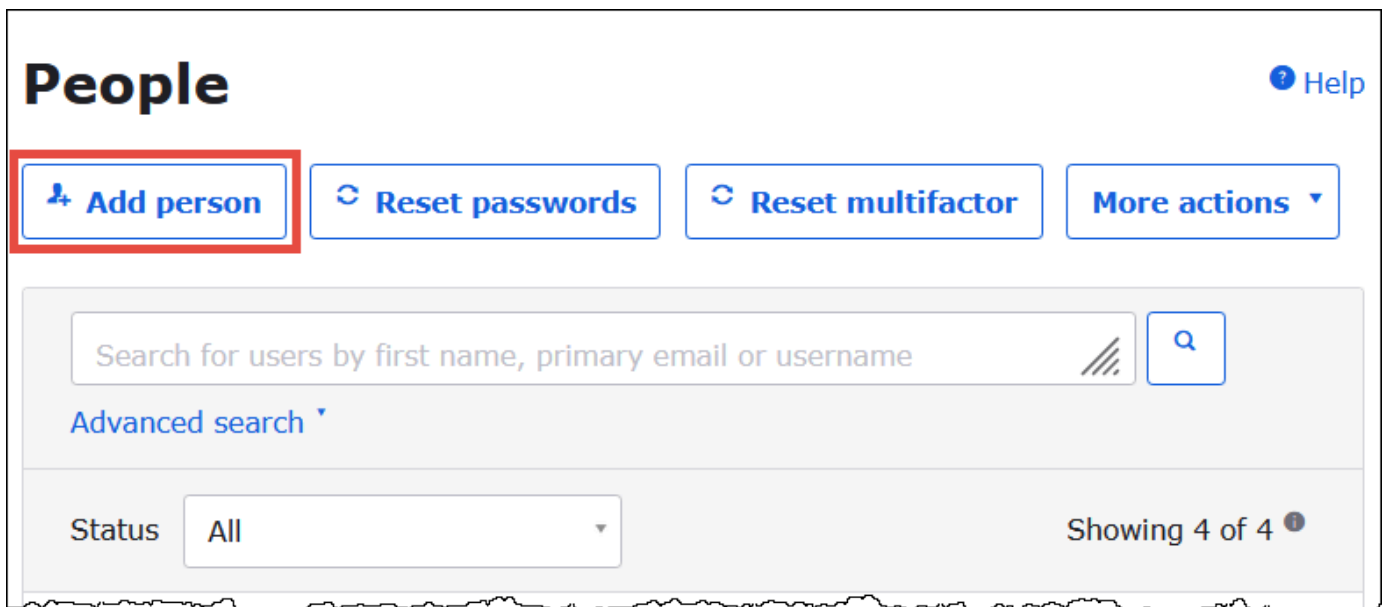
- El primer segmento después de `https://`, hasta e incluyendo `okta.com` (por ejemplo, `trial-1234567.okta.com`). Este es el host de su IdP.
- Los dos últimos segmentos de la URL, incluida la barra del medio. Los segmentos son dos cadenas de 20 caracteres con una combinación de números, y letras mayúsculas y minúsculas (por ejemplo, `Abc1de2fghi3J45kL678/abc1defghij2klmNo3p4`). Este es el ID de la aplicación.

## Adición de un usuario a la aplicación de Okta

Ya puede agregar un usuario a su aplicación Okta.

Para agregar un usuario a la aplicación de Okta

1. En el panel de navegación izquierdo, elija Directory (Directorio) y, a continuación, elija People (Personas).
2. Elija Add person (Agregar persona).



3. En el cuadro de diálogo Add person, ingrese la siguiente información.
  - En First name (Nombre) y Last name (Apellido), ingrese valores. En este tutorial se utiliza un **test user**.
  - Ingrese los valores para Username (Nombre de usuario) y Primary email (Correo electrónico principal). En este tutorial se utiliza **test@amazon.com** para ambos. Los requisitos de seguridad para las contraseñas pueden variar.

## Add Person

User type <sup>?</sup>

First name

Last name

Username

Primary email

Secondary email (optional)

Groups (optional)

Password <sup>?</sup>

Send user activation email now <sup>?</sup>

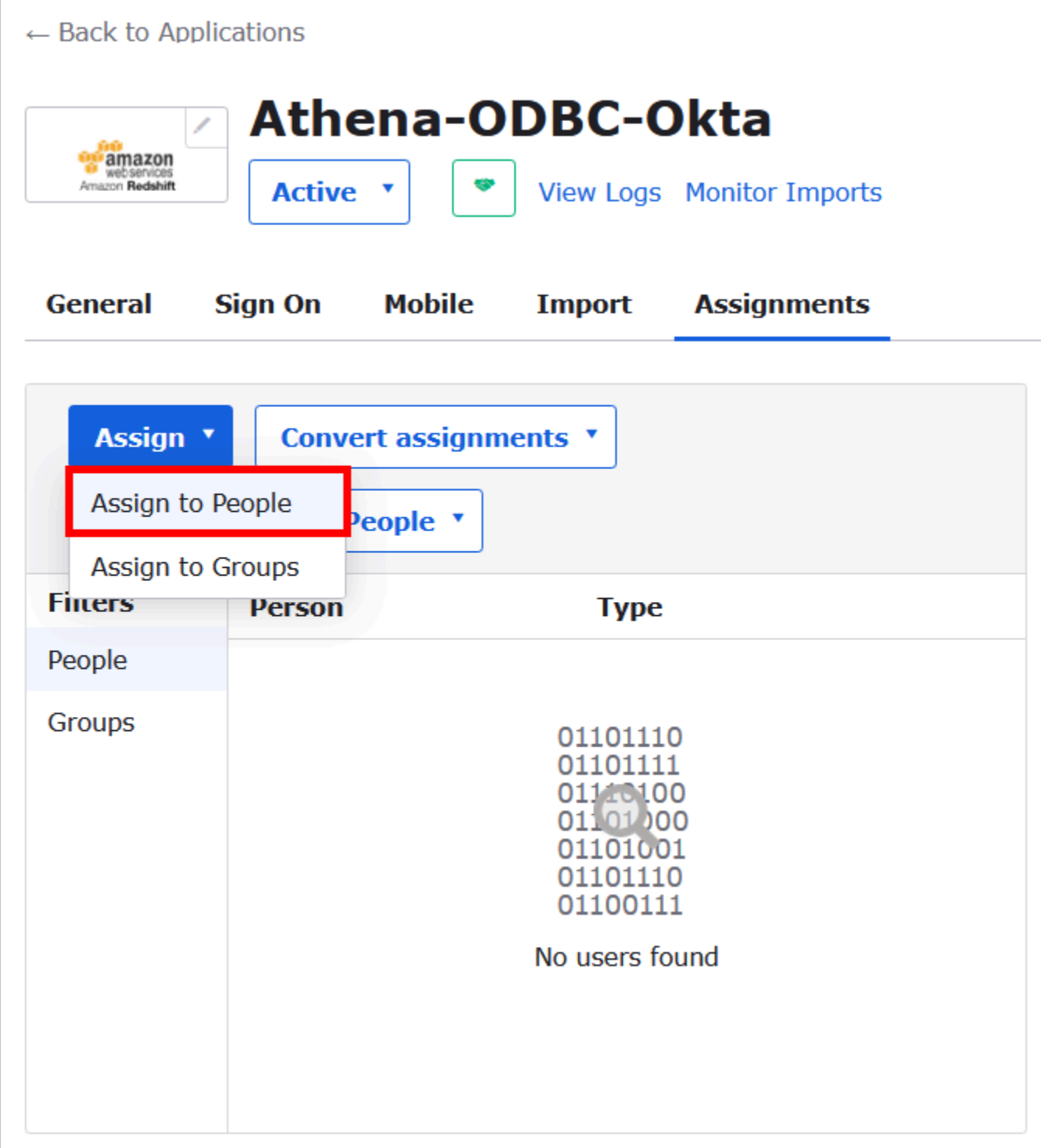
**Save** **Save and Add Another** **Cancel**

4. Elija Guardar.


Ya puede asignar el usuario que ha creado a su aplicación.


Para asignar el usuario a la aplicación:

1. En el panel de navegación, elija Applications (Aplicaciones), Applications (Aplicaciones) y, a continuación, elija el nombre de la aplicación (por ejemplo, Atena-ODBC-Okta).
2. Elija Assign (Asignar) y, luego, elija Assign to People (Asignar a personas).




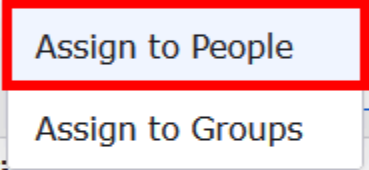
← Back to Applications

 **Athena-ODBC-Okta**

Active  View Logs Monitor Imports

General Sign On Mobile Import **Assignments**

Assign 

Assign to People 

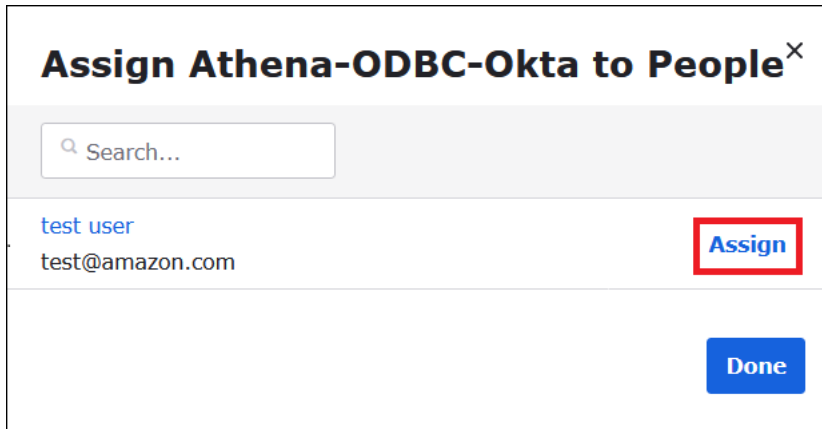
Filters Person Type

Filters	Person	Type
People		
Groups		

01101110  
01101111  
01101100  
01101000  
01101001  
01101110  
01100111

No users found

3. Elija la opción Assign (Asignar) para el usuario y, luego, seleccione Done (Listo).



4. Cuando se le solicite, elija Save and Go Back (Guardar y volver). En el cuadro de diálogo se muestra el estado del usuario como Assigned (Asignado).
5. Seleccione Listo.
6. Elija la pestaña Sign On (Iniciar sesión).
7. Desplácese hasta la sección SAML Signing Certificates (Certificados de firma de SAML).
8. Elija Acciones.
9. Abra el menú contextual (clic derecho) de View IdP metadata (Visualización de metadatos del IdP) y, a continuación, seleccione la opción del navegador para guardar el archivo.
10. Guarde el archivo con una extensión de .xml.

**SAML Signing Certificates**

[Generate new certificate](#)

Type	Type	Created	Expires	Status	Actions
SHA-2	SHA-2	Aug 2022	Aug 2032	Active	<a href="#">Actions</a> <ul style="list-style-type: none"> <li><a href="#">View IdP metadata</a></li> <li><a href="#">Download certificate</a></li> </ul>

**Sign On Policy**

[+ Add Rule](#)

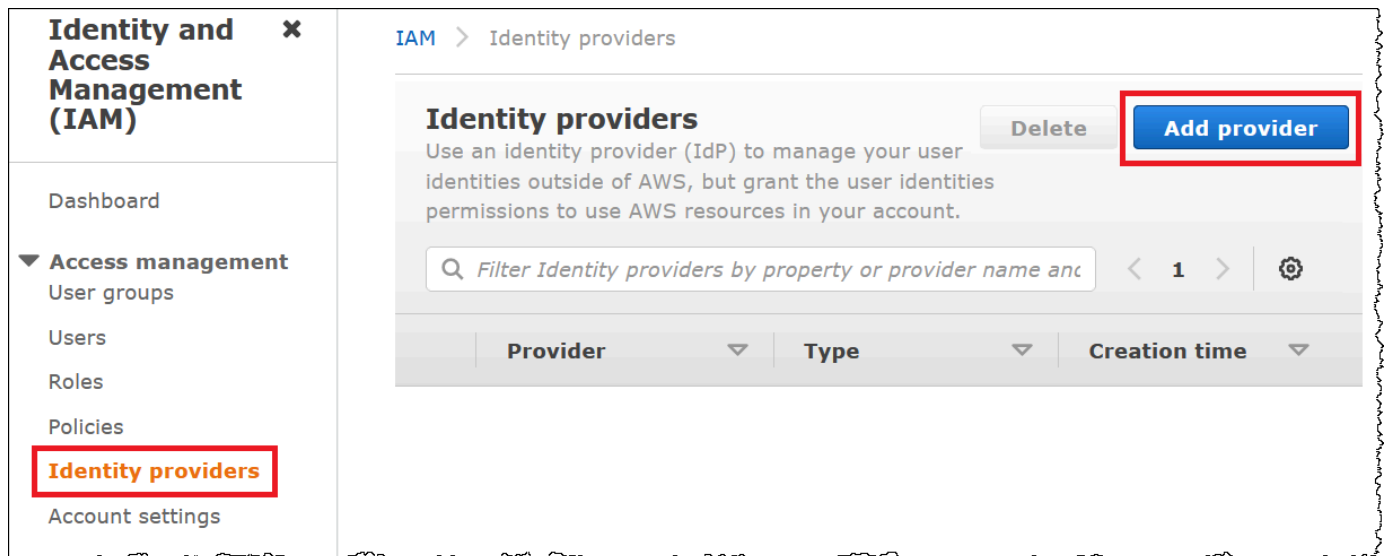
Crear un proveedor de identidad SAML y rol de AWS

Ya puede cargar el archivo XML de metadatos en la consola de IAM en AWS. Utilizará este archivo para crear un proveedor de identidad de SAML y rol de AWS. Utilice una cuenta de administrador de servicios de AWS para realizar estos pasos.

Para crear un proveedor de identidad SAML y un rol en AWS

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en <https://console.aws.amazon.com/IAM/>.
2. En el panel de navegación, elija Proveedores de identidades y, a continuación, Agregar proveedor.





3. En la página Add an Identity provider (Agregar un proveedor de identidad), en Configure provider (Configurar proveedor), ingrese la siguiente información.
  - En Provider type (Tipo de proveedor), elija SAML.
  - En Provider name (Nombre del proveedor), ingrese un nombre para el proveedor (por ejemplo, **AthenaODBCOkta**).
  - En Metadata document (Documento de metadatos), utilice la opción Choose file (Elegir archivo) para cargar el archivo XML de metadatos del proveedor de identidad (IdP) que descargó.

# Add an Identity provider

## Configure provider

Provider type

**SAML**  
Establish trust between your AWS account and a SAML 2.0 compatible Identity Provider such as Shibboleth or Active Directory Federation Services.

**OpenID Connect**  
Establish trust between your AWS account and an Identity Provider such as Google or Salesforce.

Provider name  
Enter a meaningful name to identify this provider

Maximum 128 characters. Use alphanumeric or '.', '\_' characters.

Metadata document  
This document is issued by your IdP.

File needs to be a valid UTF-8 XML document.

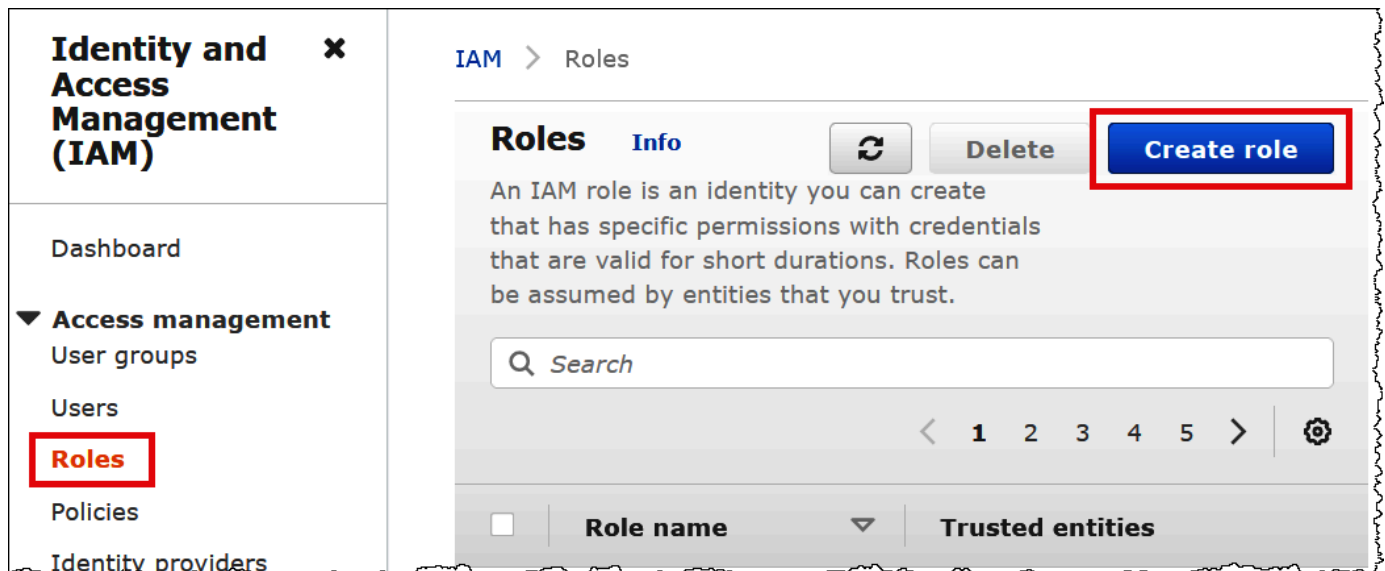
4. Elija Add Provider (Agregar proveedor).

### Creación de un rol de IAM para el acceso a Athena y Amazon S3

Ahora ya tiene todo listo para crear un rol de IAM para el acceso a Athena y Amazon S3. Asignará este rol al usuario. De esta forma, puede proporcionar al usuario acceso de inicio de sesión único a Athena.

Para crear un rol de IAM para el usuario

1. En el panel de navegación de la consola de IAM, elija Roles y, a continuación, elija Create role (Crear rol).



2. En la página Create role (Crear rol), elija las opciones siguientes:

- En Select type of trusted entity (Seleccionar tipo de entidad de confianza), elija SAML 2.0 Federation (Federación SAML 2.0).
- En SAML 2.0–based provider (Proveedor basado en SAML 2.0), seleccione el proveedor de identidad SAML que ha creado (por ejemplo, AthenaODBCOkta).
- Seleccione Permitir acceso mediante programación y mediante la AWS Management Console.

SAML 2.0 federation  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

Custom trust policy  
Create a custom trust policy to enable others to perform actions in this account.

### SAML 2.0 federation

Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

SAML 2.0-based provider

AthenaODBCOkta

Allow programmatic access only

Allow programmatic and AWS Management Console access

Attribute

SAML:aud

Value

https://signin.aws.amazon.com/saml

Condition - (optional)

3. Elija Siguiente.
4. En la página Add Permissions (Agregar permisos), para Filter policies (Filtrar políticas), ingrese **AthenaFull** y, a continuación, pulse INTRO.
5. Seleccione la política administrada de AmazonAthenaFullAccess y, a continuación, elija Next (Siguiente).

## Add permissions

**Permissions policies** (Selected 1/819)



Create policy

Choose one or more policies to attach to your new role.

1 match

< 1 >



"AthenaFull"

Clear filters



Policy name



Type



Description



AmazonAthenaFullAccess

AWS managed

Provide full access to

### ► Set permissions boundary - optional

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel

Previous

Next

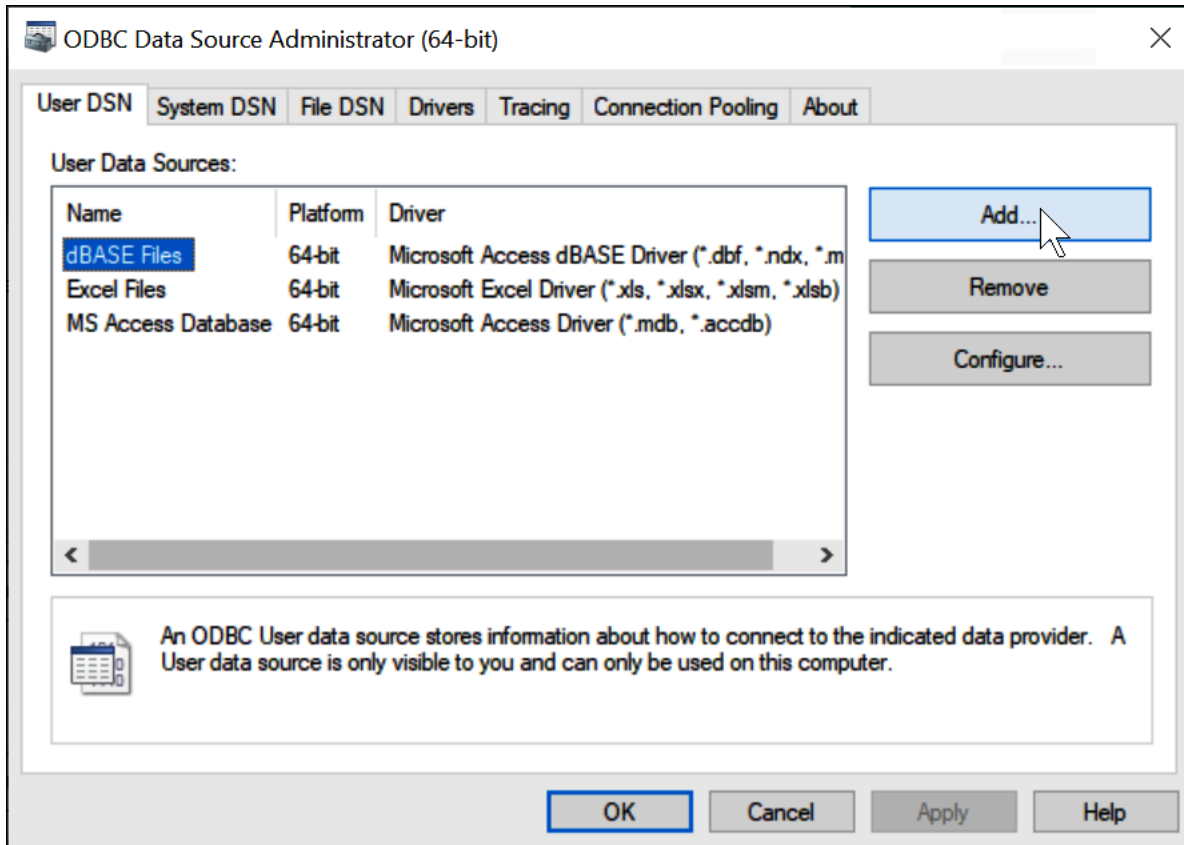
6. En la página Asignar nombre, revisar y crear, en Nombre del rol, ingrese un nombre para el rol (por ejemplo, **Athena-ODBC-OktaRole**) y luego elija Crear rol.

### Configuración de la conexión ODBC de Okta a Athena

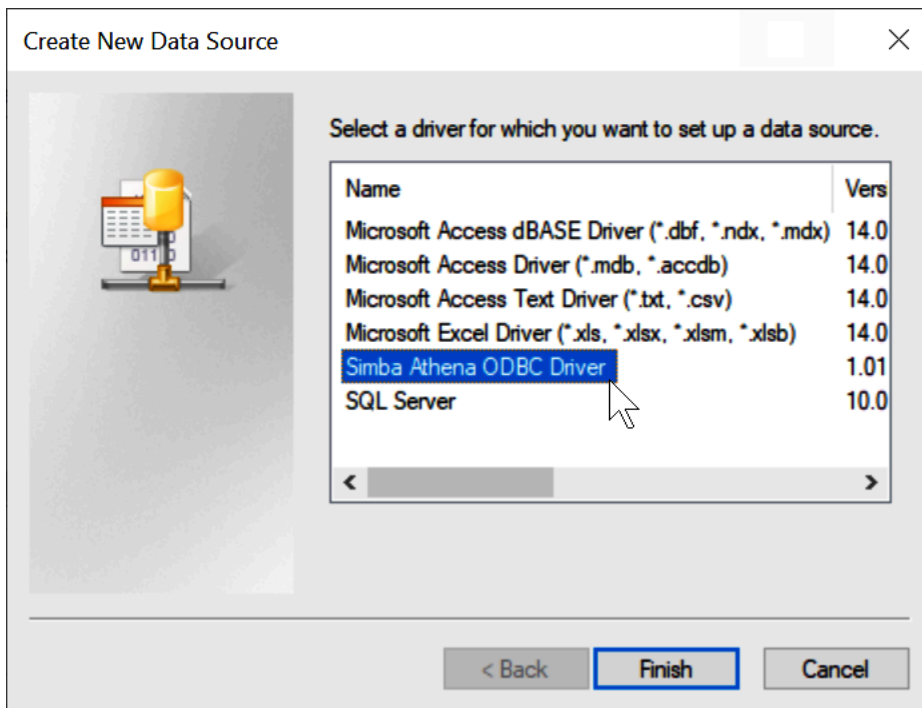
Ya tiene todo listo para configurar la conexión ODBC de Okta a Athena con el programa de orígenes de datos ODBC en Windows.

## Para configurar la conexión ODBC de Okta a Athena

1. En Windows, inicie el programa ODBC Data Sources (Orígenes de datos ODBC).
2. En el programa ODBC Data Source Administrator (Administrador de orígenes de datos ODBC), elija Add (Agregar).



3. Elija Simba Athena ODBC Driver (Controlador ODBC Simba Athena) y, a continuación, elija Finish (Finalizar).



4. En el cuadro de diálogo Simba Athena ODBC Driver DSN Setup (Configuración de DSN del controlador ODBC Simba Athena), ingrese los valores descritos.
  - En Data Source Name (Nombre de origen de datos), ingrese un nombre para el origen de datos (por ejemplo, **Athena ODBC 64**).
  - En Description (Descripción), ingrese una descripción para el origen de datos.
  - En Región de AWS, ingrese la Región de AWS que usa actualmente (por ejemplo, **us-west-1**).
  - En S3 Output Location (Ubicación de salida de S3), ingrese la ruta de Amazon S3 en la que desea que se almacenen los datos de salida.

Simba Athena ODBC Driver DSN Setup

Data Source Name: Athena ODBC 64

Description: My ODBC Data Source

AWS Region: us-west-1

Catalog: AwsDataCatalog

Schema: default

Workgroup: primary

Metadata Retrieval Method: Auto

Output Options

S3 Output Location: s3://DOC-EXAMPLE-BUCKET

Encryption Options: NOT\_SET

KMS Key:

Endpoint Override:

Streaming Endpoint Override:

Authentication Options... Advanced Options... Logging Options...

Proxy Options...

v1.1.13.1000 (64 bit) Test... OK Cancel

5. Elija Authentication Options (Opciones de autenticación).
6. En el cuadro de diálogo Authentication Options (Opciones de autenticación), elija o ingrese los siguientes valores.
  - En Authentication Type (Tipo de autenticación), elija Okta.
  - Para User (Usuario), escriba su nombre de usuario de Okta.
  - En Password (Contraseña), escriba su contraseña de Okta.
  - En IdP Host (Host de IdP), ingrese el valor que registró anteriormente (por ejemplo, **trial-1234567.okta.com**).



- En IdP Port (Puerto de IdP), ingrese **443**.
- En App ID (ID de aplicación), ingrese el valor que registró anteriormente (los dos últimos segmentos de su enlace de inserción de Okta).
- En Okta App Name (Nombre de aplicación Okta), ingrese **amazon\_aws\_redshift**.

Authentication Options

Authentication Type: Okta

User: test@amazon.com

Password: ●●●●●●●●

Password Options...

Session Token:

Preferred Role:

Session Duration:

IdP Host: trial-...okta.com

IdP Port: 443

App ID:

Okta App Name: amazon\_aws\_redshift

Okta MFA wait time:

Okta MFA Type:

Okta MFA Phone No:

Use HTTP Proxy For IdP Host  SSL Insecure

OK Cancel

7. Seleccione Aceptar.
8. Elija Test (Probar) para realizar una prueba de la conexión o OK (Aceptar) para finalizar.

## Configuración del inicio de sesión único con ODBC, SAML 2.0 y el proveedor de identidades Okta

Para conectarse a orígenes de datos puede usar Amazon Athena con proveedores de identidades (IdP) como PingOne, Okta, OneLogin y otros. A partir de la versión 1.1.13 del controlador ODBC de Athena y de la versión 2.0.25 del controlador JDBC de Athena, se incluye un complemento SAML para el navegador que se puede configurar para que funcione con cualquier proveedor de SAML 2.0. Este tema muestra cómo configurar el controlador ODBC de Amazon Athena y el complemento SAML basado en navegador para agregar la capacidad de inicio de sesión único (SSO) con el proveedor de identidades Okta.

### Requisitos previos

Para completar los pasos de este tutorial, se necesitan las siguientes versiones:

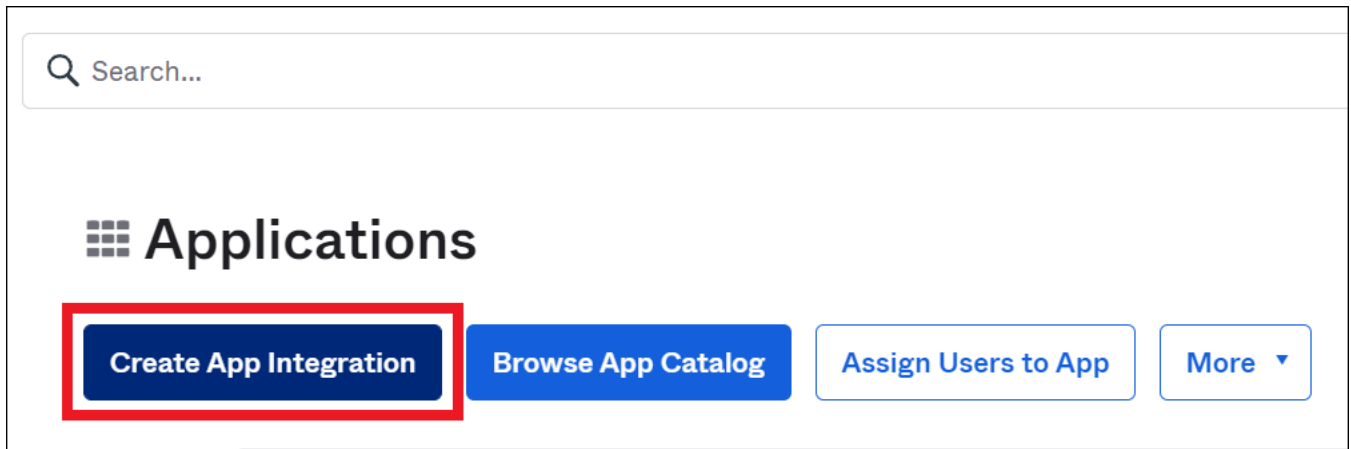
- Controlador Athena ODBC versión 1.1.13 o posterior. Las versiones 1.1.13 y posteriores incluyen la compatibilidad con SAML del navegador. Para obtener los enlaces de descarga, consulte [Conexión a Amazon Athena con ODBC](#).
- Un rol de IAM que quiera usar con SAML. Para obtener más información, consulte [Creación de un rol para una federación SAML 2.0](#) en la Guía del usuario de IAM.
- Una cuenta de Okta. Para obtener información, visite [okta.com](https://okta.com).

### Creación de una integración de aplicaciones en Okta

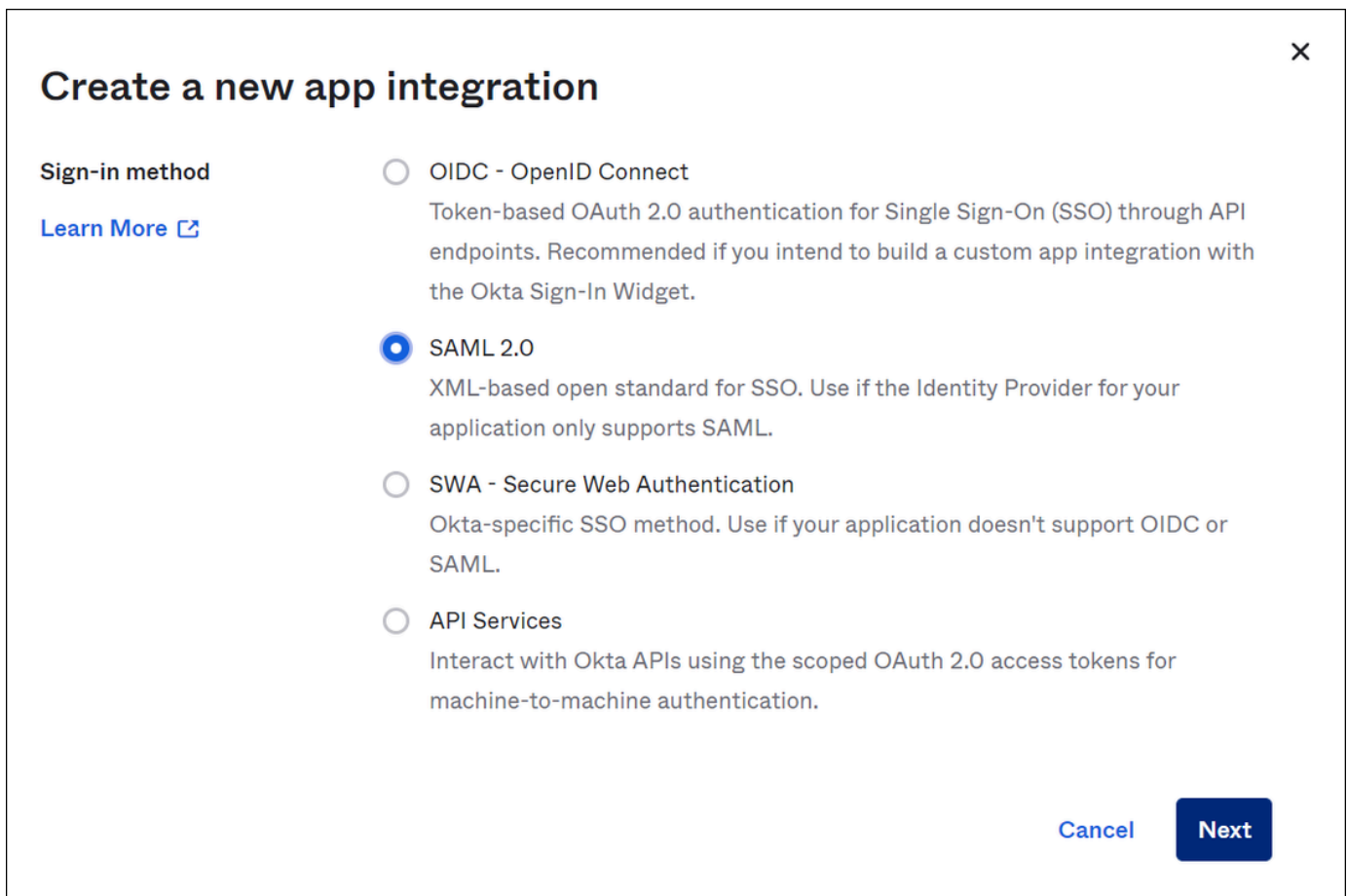
En primer lugar, use el panel de control de Okta para crear y configurar una aplicación SAML 2.0 para el inicio de sesión único en Athena.

Para usar el panel de control de Okta para configurar el inicio de sesión único para Athena

1. Inicie sesión en la página de administración de Okta en [okta.com](https://okta.com).
2. En el panel de navegación, elija Applications (Aplicaciones), Applications.
3. En la página Applications, elija Create App Integration (Crear integración de aplicaciones).






4. En el cuadro de diálogo Create a new app integration (Crear una nueva integración de aplicaciones), en Sign-in method (Método de inicio de sesión), seleccione SAML 2.0 y, a continuación, elija Next (Siguiente).




5. En la página Create SAML Integration (Crear integración SAML), en la sección General Settings (Configuración general), ingrese un nombre para la aplicación. Este tutorial usa el nombre Athena SSO.

### 1 General Settings

App name

App logo (optional)   



App visibility  Do not display application icon to users  
 Do not display application icon in the Okta Mobile app

[Cancel](#) [Next](#)

6. Elija Siguiente.

7. En la página Configure SAML (Configurar SAML), en la sección SAML Settings (Configuración de SAML), ingrese los siguientes valores:

- En Single sign on URL (URL de inicio de sesión único), ingrese **http://localhost:7890/athena**
- En Audience URI (URI de la audiencia), ingrese **urn:amazon:webservices**

## A SAML Settings

### General

Single sign on URL <sup>?</sup>

Use this for Recipient URL and Destination URL

Allow this app to request other SSO URLs

Audience URI (SP Entity ID) <sup>?</sup>

Default RelayState <sup>?</sup>

If no value is set, a blank RelayState is sent

Name ID format <sup>?</sup>

Application username <sup>?</sup>

[Show Advanced Settings](#)

### Attribute Statements (optional)

[LEARN MORE](#)

8. En Attribute Statements (Instrucciones de los atributos), ingrese los siguientes dos pares de nombre/valor. Son atributos de asignación obligatorios.

- En Name (Nombre), ingrese la siguiente URL:

**`https://aws.amazon.com/SAML/Attributes/Role`**

En Value (Valor), ingrese el nombre de su rol de IAM. Para obtener información sobre el formato del rol de IAM, consulte [Configuración de aserciones SAML para la respuesta de autenticación](#) en la Guía del usuario de IAM..

- En Name, ingrese la siguiente URL:

**`https://aws.amazon.com/SAML/Attributes/RoleSessionName`**

En Value, ingrese **`user.email`**.

### Attribute Statements (optional) [LEARN MORE](#)

Name	Name format (optional)	Value
<input type="text" value="https://aws."/>	<input type="text" value="Unspecified"/>	<input type="text" value="YOUR_ROLE"/>
<input type="text" value="https://aws."/>	<input type="text" value="Unspecified"/>	<input type="text" value="user.email"/>

9. Elija Siguiente y, a continuación, elija Finalizar.

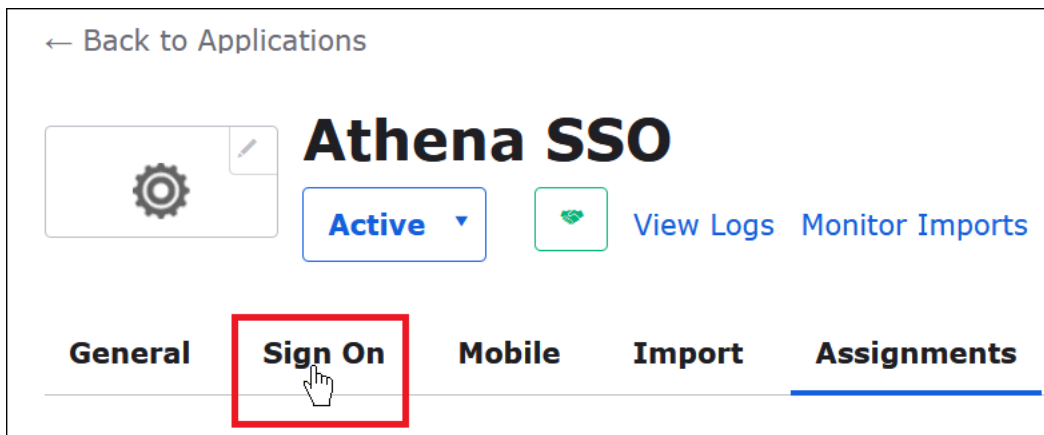
Cuando Okta crea la aplicación, también crea su URL de inicio de sesión, que recuperará a continuación.

## Obtención de la URL de inicio de sesión desde el panel de control de Okta

Ahora que se creó su aplicación, puede obtener la URL de inicio de sesión y otros metadatos desde el panel de control de Okta.

Cómo obtener la URL de inicio de sesión desde el panel de control de Okta

1. En el panel de navegación de Okta, elija Applications (Aplicaciones), Applications.
2. Elija la aplicación para la que necesita la URL de inicio de sesión (por ejemplo, AthenaSSO).
3. En la página de su aplicación, elija Sign On (Iniciar sesión).




4. Elija View Setup Instructions (Ver instrucciones de configuración).




← Back to Applications

# Athena SSO

**Active**  [View Logs](#) [Monitor Imports](#)

**General** **Sign On** **Mobile** **Import** **Assignments**

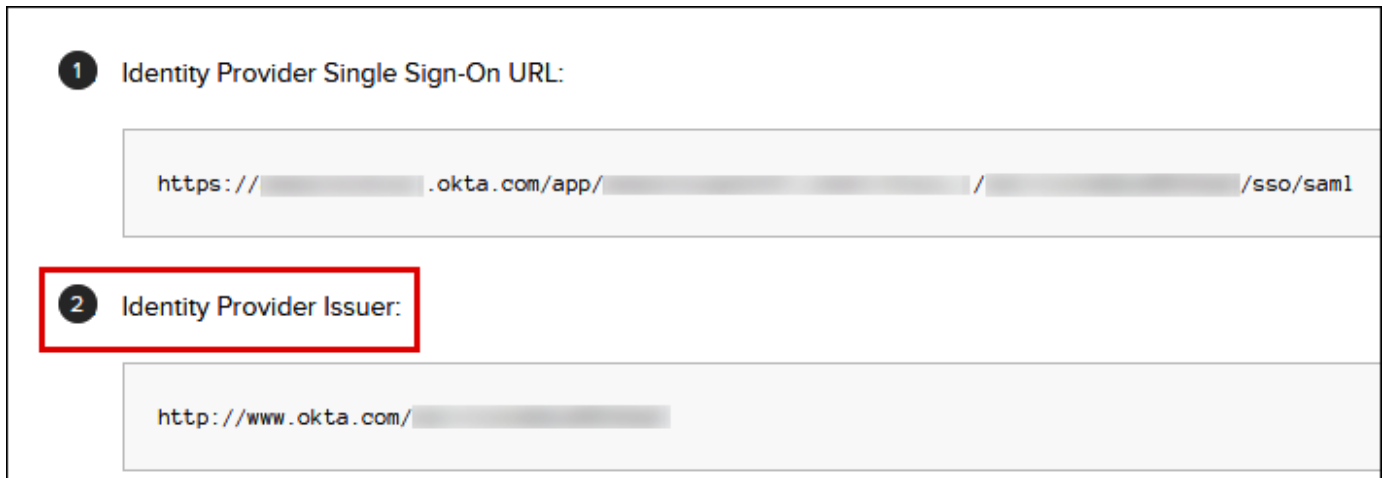
## Settings [Edit](#)

 **SAML 2.0** is not configured until you complete the setup instructions.

[View Setup Instructions](#)

[Identity Provider metadata](#) is available if this application supports dynamic configuration.

5. En la página Cómo configurar SAML 2.0 para Athena SSO, busque la URL del Identity Provider Issuer (Proveedor de identidades emisor). En algunas partes del panel de Okta hacen referencia a esta URL como SAML issuer ID (ID del emisor de SAML).



1 Identity Provider Single Sign-On URL:

`https://[redacted].okta.com/app/[redacted]/[redacted]/sso/saml`

2 Identity Provider Issuer:

`http://www.okta.com/[redacted]`

6. Copie o almacene el valor para Identity Provider Single Sign-On URL (URL de inicio de sesión único del proveedor de identidades).

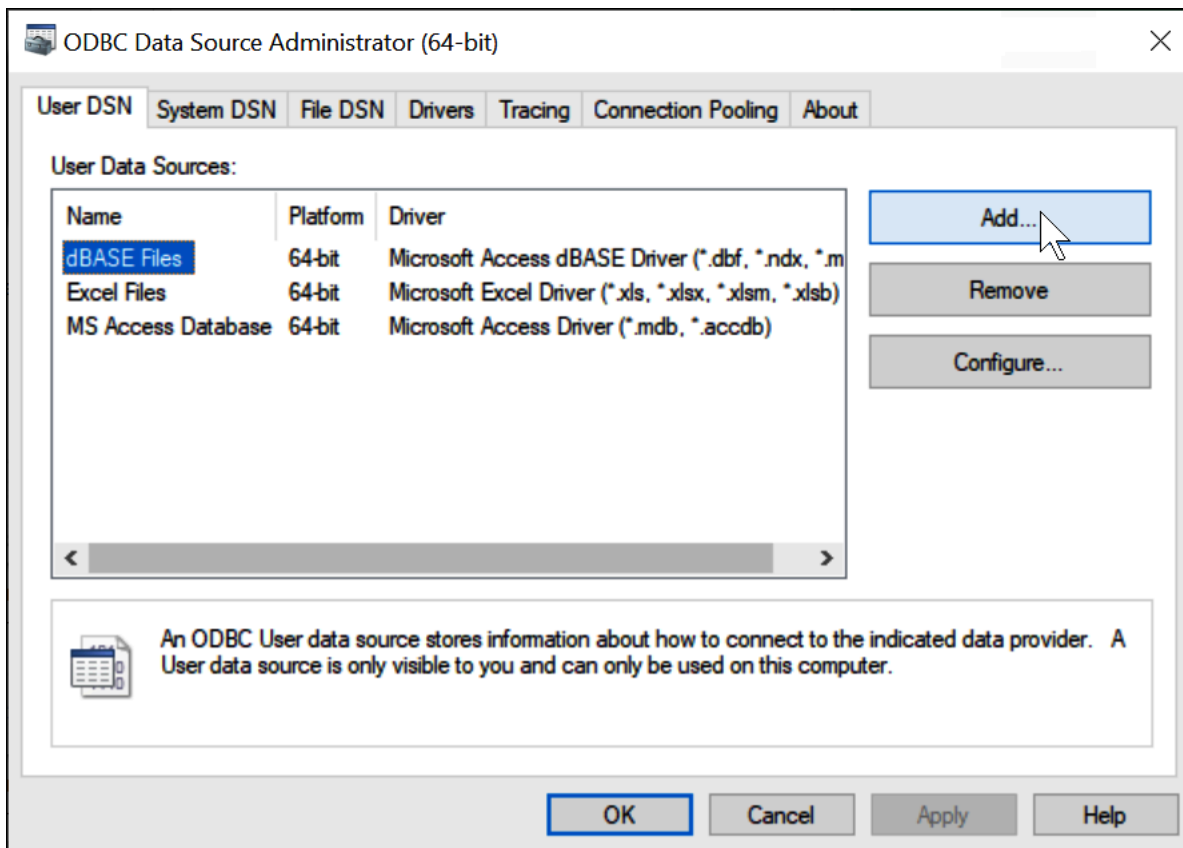
En la siguiente sección, cuando configure la conexión ODBC, proporcionará este valor como el parámetro de conexión Login URL (URL de inicio de sesión) para el complemento SAML del navegador.

#### Configuración de la conexión ODBC del navegador SAML a Athena

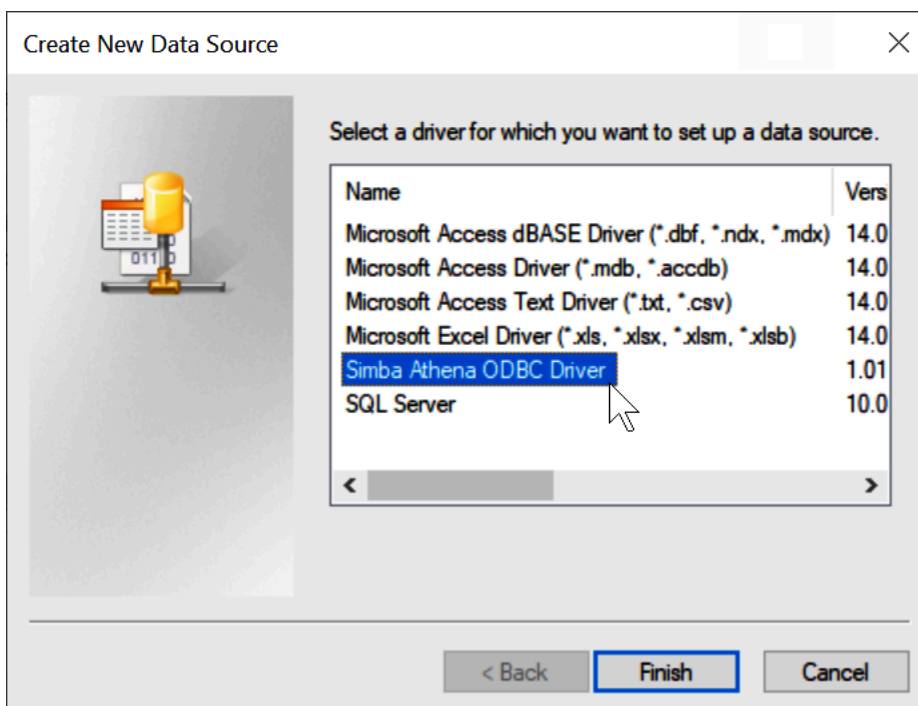
Ahora ya tiene todo listo para configurar la conexión SAML del navegador a Athena con el programa de orígenes de datos ODBC en Windows.

Para configurar la conexión SAML ODBC del navegador a Athena

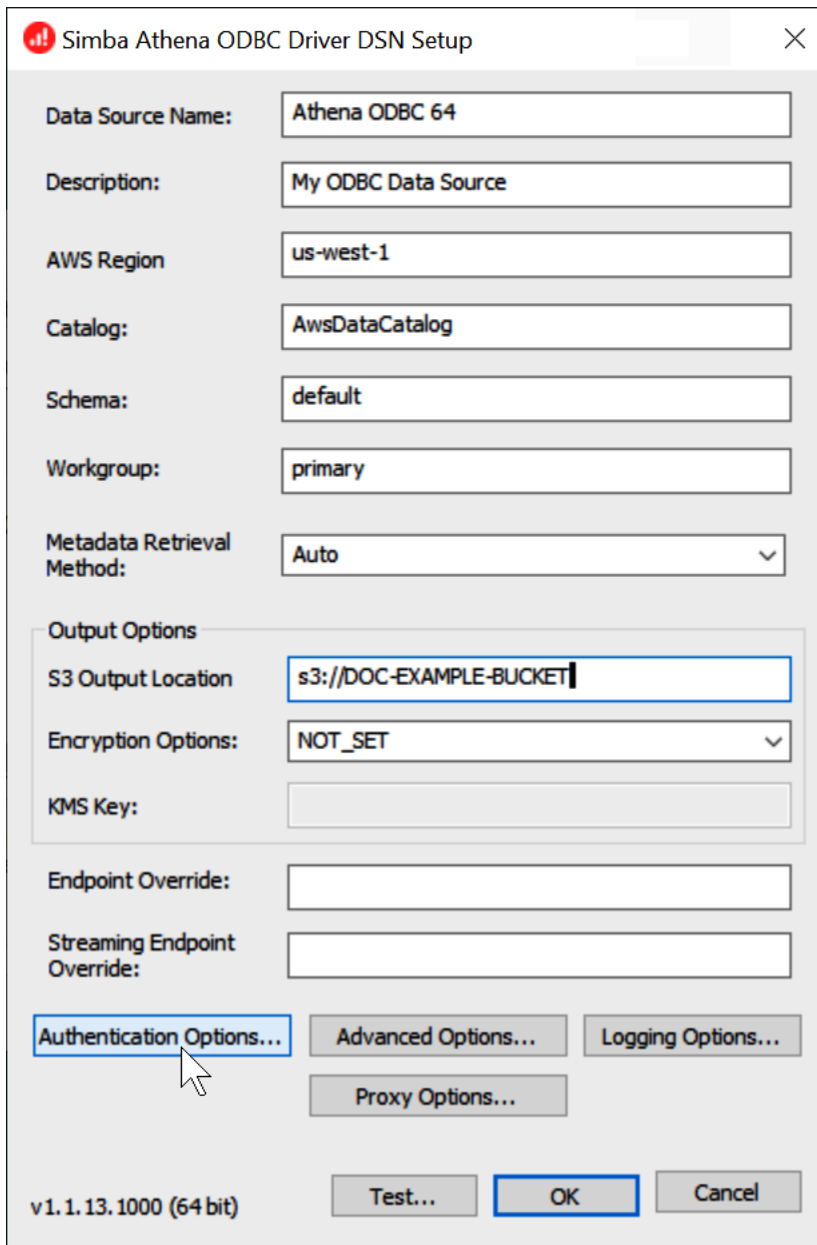
1. En Windows, inicie el programa ODBC Data Sources (Orígenes de datos ODBC).
2. En el programa ODBC Data Source Administrator (Administrador de orígenes de datos ODBC), elija Add (Agregar).



3. Elija Simba Athena ODBC Driver (Controlador ODBC Simba Athena) y, a continuación, elija Finish (Finalizar).



4. En el cuadro de diálogo Simba Athena ODBC Driver DSN Setup (Configuración de DSN del controlador ODBC Simba Athena), ingrese los valores descritos.



Simba Athena ODBC Driver DSN Setup

Data Source Name: Athena ODBC 64

Description: My ODBC Data Source

AWS Region: us-west-1

Catalog: AwsDataCatalog

Schema: default

Workgroup: primary

Metadata Retrieval Method: Auto

Output Options

S3 Output Location: s3://DOC-EXAMPLE-BUCKET

Encryption Options: NOT\_SET

KMS Key:

Endpoint Override:

Streaming Endpoint Override:

Authentication Options... Advanced Options... Logging Options...

Proxy Options...

v1.1.13.1000 (64 bit) Test... OK Cancel

- En Data Source Name (Nombre de origen de datos), ingrese un nombre para el origen de datos (por ejemplo, Athena ODBC 64).
  - En Description (Descripción), ingrese una descripción para el origen de datos.
  - En Región de AWS, ingrese la Región de AWS que usa (por ejemplo, **us-west-1**).
  - En S3 Output Location (Ubicación de salida de S3), ingrese la ruta de Amazon S3 en la que desea que se almacenen los datos de salida.
5. Elija Authentication Options (Opciones de autenticación).

6. En el cuadro de diálogo Authentication Options (Opciones de autenticación), elija o ingrese los siguientes valores.

### Authentication Options

Authentication Type:

User:

Password:

Session Token:

Preferred Role:

Session Duration:

Login URL:

Listen Port:

Timeout (sec):

Use HTTP Proxy For IdP Host       SSL Insecure

- En Authentication Type (Tipo de autenticación), elija BrowserSAML.
  - En Login URL (URL de inicio de sesión), ingrese la URL de inicio de sesión único del proveedor de identidades que obtuvo del panel de control de Okta.
  - En Listen Port (Puerto de escucha), ingrese 7890.
  - En Timeout (sec) (Tiempo de espera en segundos), ingrese un valor de tiempo de espera de conexión en segundos.
7. Elija OK (Aceptar) para cerrar las Authentication Options (Opciones de autenticación).
  8. Elija Test (Probar) para probar la conexión o OK (Aceptar) para finalizar.

## Uso del conector Power BI de Amazon Athena

En los sistemas operativos Windows, puede utilizar el conector Microsoft Power BI para Amazon Athena para analizar datos de Amazon Athena en Microsoft Power BI Desktop. Para obtener información sobre Power BI, consulte [Microsoft Power BI](#). Después de publicar contenido en el servicio Power BI, puede utilizar la versión de julio de 2021 de [puerta de enlace de Power BI](#) o una posterior para mantener el contenido actualizado mediante actualizaciones bajo demanda o programadas.

### Requisitos previos

Antes de comenzar, asegúrese de que el entorno cumple los siguientes requisitos. Se requiere el controlador ODBC de Amazon Athena.

- [Cuenta de AWS](#)
- [Permisos para utilizar Athena](#)
- [Controlador ODBC de Amazon Athena](#)
- [Power BI Desktop](#)

### Capacidades compatibles

- Import: las tablas y columnas seleccionadas se importan a Power BI Desktop para realizar consultas.
- DirectQuery: no se importan ni copian datos en Power BI Desktop. Power BI Desktop consulta directamente el origen de datos subyacente.

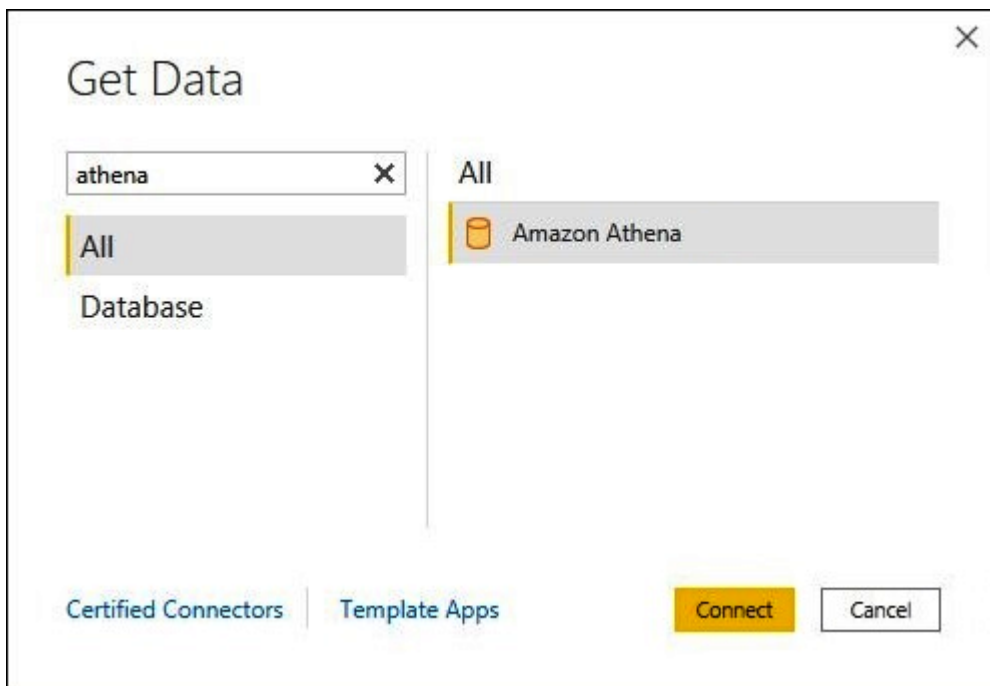
- Puerta de enlace de Power BI: una puerta de enlace de datos en las instalaciones en su Cuenta de AWS que funciona como un puente entre Microsoft Power BI Service y Athena. La gateway es necesaria para ver los datos en el servicio Microsoft Power BI.

## Conectar a Amazon Athena

Siga estos pasos para conectar el escritorio de Power BI a los datos de Amazon Athena.

Para conectarse a los datos de Athena desde Power BI Desktop

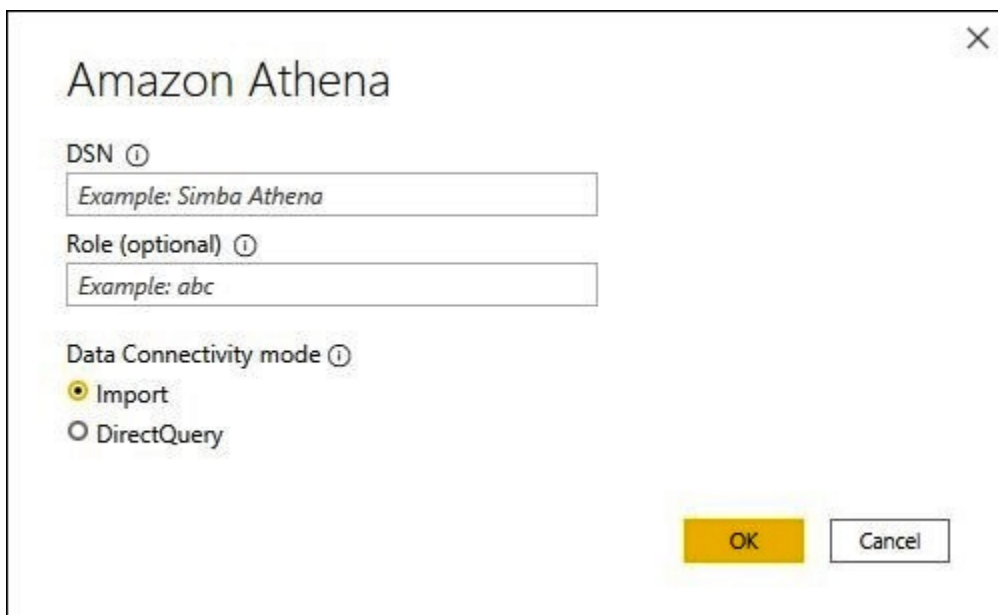
1. Inicie Power BI Desktop.
2. Realice una de las acciones siguientes:
  - Elija File (Archivo), Get Data (Obtener datos)
  - Desde la barra de herramientas de Inicio, elija Get Data (Obtener datos).
3. En el cuadro de búsqueda, ingrese Athena.
4. Seleccione Amazon Athena y luego elija Connect (Conectar).



5. En la página de conexión Amazon Athena, ingrese la siguiente información.
  - Para DSN, ingrese el nombre del DSN ODBC que desea utilizar. Para obtener instrucciones sobre cómo configurar su DSN, consulte la [Documentación del controlador ODBC](#).

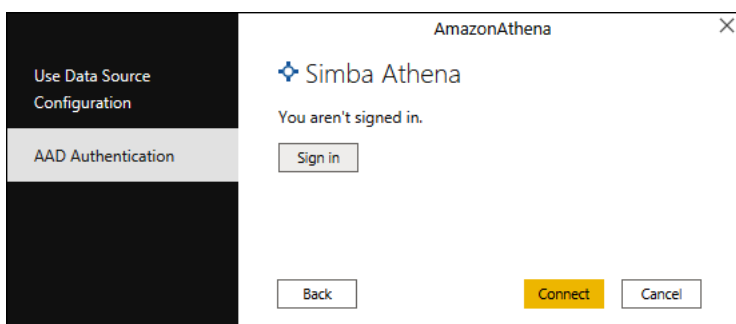


- En Modo de conectividad de datos, elija un modo adecuado para su caso de uso, siguiendo estas pautas generales:
  - Para conjuntos de datos más pequeños, elija Import (Importar). Al utilizar el modo Importar, Power BI trabaja con Athena para importar el contenido de todo el conjunto de datos para utilizarlo en las visualizaciones.
  - Para conjuntos de datos más grandes, elija DirectQuery. En el modo DirectQuery, no se descargan datos en la estación de trabajo. Mientras crea o interactúa con una visualización, Microsoft Power BI trabaja con Athena para consultar de manera dinámica el origen de datos subyacente para que siempre esté viendo los datos actuales. Para obtener más información sobre DirectQuery, consulte [Use DirectQuery in power BI desktop](#) (Utilizar DirectQuery en Power BI Desktop) en la documentación de Microsoft.



The screenshot shows a dialog box titled "Amazon Athena" with a close button (X) in the top right corner. It contains three input fields and two radio buttons. The first field is labeled "DSN" with a help icon and contains the text "Example: Simba Athena". The second field is labeled "Role (optional)" with a help icon and contains the text "Example: abc". The third section is labeled "Data Connectivity mode" with a help icon and contains two radio buttons: "Import" (which is selected) and "DirectQuery". At the bottom right, there are two buttons: "OK" (highlighted in yellow) and "Cancel".

6. Seleccione Aceptar.
7. En el mensaje para configurar la autenticación del origen de datos, elija Usar configuración de origen de datos o Autenticación AAD y, a continuación, elija Connect (Conectar).



The screenshot shows a dialog box titled "AmazonAthena" with a close button (X) in the top right corner. On the left side, there is a vertical navigation pane with two options: "Use Data Source Configuration" (highlighted in black) and "AAD Authentication" (highlighted in grey). The main area of the dialog shows the text "Simba Athena" with a blue diamond icon, followed by "You aren't signed in." and a "Sign in" button. At the bottom, there are three buttons: "Back", "Connect" (highlighted in yellow), and "Cancel".

El catálogo de datos, las bases de datos y las tablas aparecen en el cuadro de diálogo Navigator (Navegador).

Navigator

Display Options ▾

- demo-dsn [1]
  - AwsDataCatalog [3]
    - default [8]
      - demo-datasets [2]
        - iris
        - demo\_datasets
      - sampledb [5]

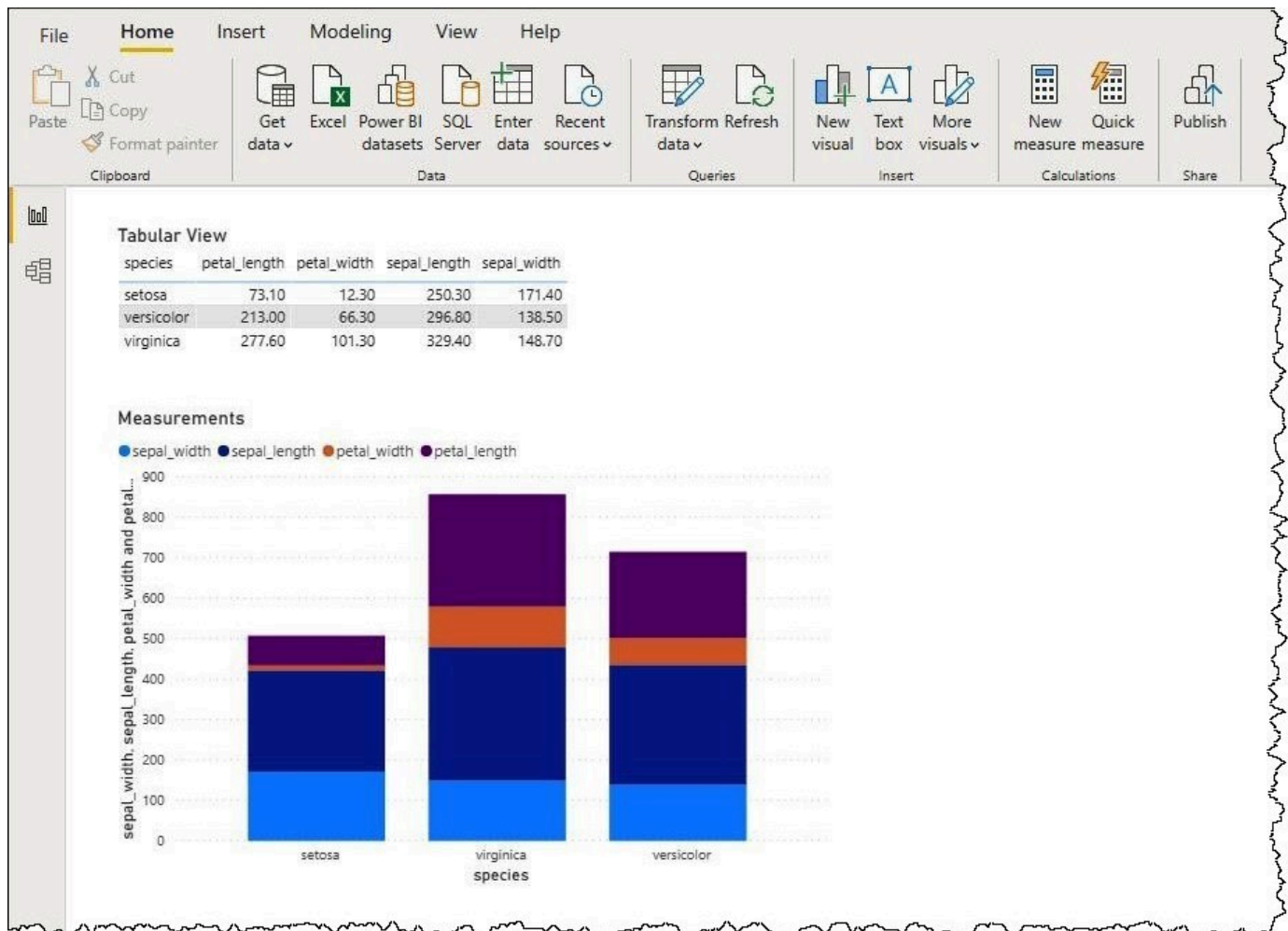
iris

Preview downloaded on Thursday

species	sepal_length	sepal_width	petal_length	petal_width
setosa	5.1	3.5	1.4	0.1
setosa	4.9	3	1.4	0.1
setosa	4.7	3.2	1.3	0.1
setosa	4.6	3.1	1.5	0.1
setosa	5	3.6	1.4	0.1
setosa	5.4	3.9	1.7	0.1
setosa	4.6	3.4	1.4	0.1
setosa	5	3.4	1.5	0.1
setosa	4.4	2.9	1.4	0.1
setosa	4.9	3.1	1.5	0.1
setosa	5.4	3.7	1.5	0.1
setosa	4.8	3.4	1.6	0.1
setosa	4.8	3	1.4	0.1
setosa	4.3	3	1.1	0.1
setosa	5.8	4	1.2	0.1
setosa	5.7	4.4	1.5	0.1
setosa	5.4	3.9	1.3	0.1
setosa	5.1	3.5	1.4	0.1
setosa	5.7	3.8	1.7	0.1
setosa	5.1	3.8	1.5	0.1
setosa	5.4	3.4	1.7	0.1
setosa	5.1	3.7	1.5	0.1

Load Transform Data Cancel

8. En el panel Display Options (Opciones de visualización), seleccione la casilla de verificación del conjunto de datos que desee utilizar.
9. Si desea transformar el conjunto de datos antes de importarlo, vaya a la parte inferior del cuadro de diálogo y elija Transform Data (Transformar datos). Esto abre el Editor de consultas de Power para que pueda filtrar y refinar el conjunto de datos que desea utilizar.
10. Elija Load (Cargar). Una vez completada la carga, puede crear visualizaciones como la de la imagen siguiente. Si seleccionó DirectQuery como modo de importación, Power BI emite una consulta a Athena para obtener la visualización solicitada.



## Configuración de una gateway en las instalaciones

Puede publicar paneles y conjuntos de datos en el servicio Power BI para que otros usuarios puedan interactuar con ellos a través de aplicaciones web, móviles e integradas. Para ver los datos en el servicio Microsoft Power BI, instale la gateway de datos en las instalaciones de Microsoft Power BI en su Cuenta de AWS. La gateway funciona como un puente entre el servicio Microsoft Power BI y Athena.

Para descargar, instalar y probar una gateway de datos en las instalaciones

1. Visite la página [Descarga de la puerta de enlace de Microsoft Power BI](#) y elija el modo personal o estándar. El modo personal es útil para probar el conector Athena de manera local. El modo estándar es adecuado en una configuración de producción multiusuario.
2. Para instalar una gateway en las instalaciones (en modo personal o estándar), consulte [Instalar una gateway de datos local](#) en la documentación de Microsoft.

3. Para probar la gateway, siga los pasos de [Usar conectores de datos personalizados con la gateway de datos en las instalaciones](#) en la documentación de Microsoft.

Para obtener más información acerca de las gateways de datos en las instalaciones, consulte los siguientes recursos de Microsoft.

- [¿Qué es una puerta de enlace de datos en las instalaciones?](#)
- [Guía para implementar una puerta de enlace de datos para Power BI](#)

Para obtener un ejemplo de configuración de la puerta de enlace de Power BI para su uso con Athena, consulte el artículo de blog de macrodatos de AWS [Creating dashboards quickly on Microsoft power BI using amazon Athena](#) (Crear paneles rápidamente en Microsoft Power BI con Amazon Athena).

## Creación de bases de datos y tablas

Amazon Athena admite un subconjunto de instrucciones del lenguaje de definición de datos (DDL) y funciones y operadores de ANSI SQL para definir y consultar tablas externas cuyos datos se encuentran en Amazon Simple Storage Service.

Al crear una base de datos y una tabla en Athena, se describe el esquema y la ubicación de los datos, lo que permite que los datos de la tabla estén listos para efectuar las consultas en tiempo real.

Para mejorar el rendimiento de las consultas y reducir los costos, se recomienda dividir los datos en particiones y utilizar formatos en columnas de código abierto para el almacenamiento en Simple Storage Service (Amazon S3), como [Apache Parquet](#) u [ORC](#).

### Temas

- [Creación de bases de datos en Athena](#)
- [Creación de tablas en Athena](#)
- [Nombres de tablas, bases de datos y columnas](#)
- [Palabras clave reservadas](#)
- [Ubicación de las tablas en Simple Storage Service \(Amazon S3\)](#)
- [Formatos de almacenamiento en columnas](#)
- [Conversión a formatos de columnas](#)

- [Particiones de datos en Athena](#)
- [Proyección de particiones con Amazon Athena](#)

## Creación de bases de datos en Athena

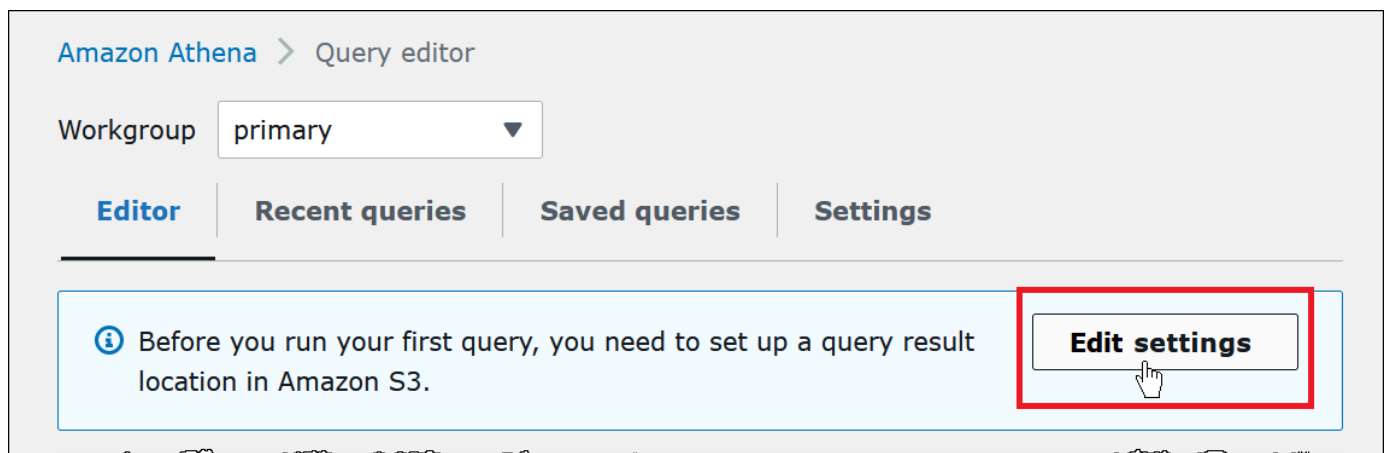
En Athena, una base de datos es una agrupación lógica de tablas creadas en él.

### Requisitos previos

Si aún no tiene configurada una ubicación de salida de consultas en Amazon S3, proceda con los siguientes pasos previos para hacerlo.

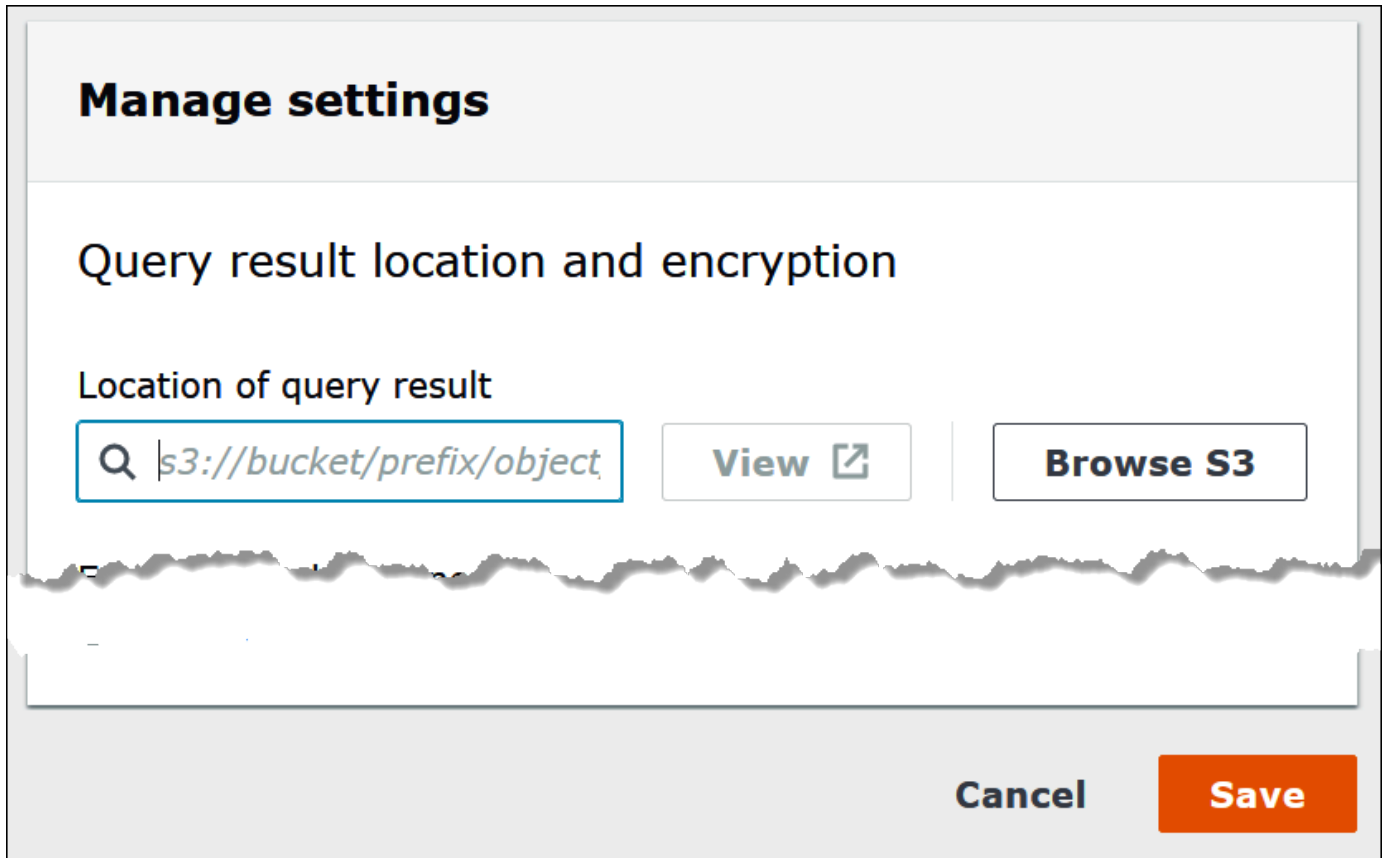
Para crear una ubicación de salida de consultas

1. Con la misma cuenta y Región de AWS que utiliza para Athena, siga los pasos (por ejemplo, con la consola de Amazon S3) para [crear un bucket en Amazon S3](#) a fin de retener los resultados de las consultas de Athena. Configuraré este bucket para que sea la ubicación de salida de consultas.
2. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
3. Si es la primera vez que visita la consola de Athena en esta Región de AWS, elija Explorar el editor de consultas para abrir el editor de consultas. De lo contrario, Athena abre la consulta en el editor de consultas.
4. Elija Edit Settings (Editar configuración) para configurar una ubicación de resultados de la consulta en Amazon S3.



5. En Manage settings (Administrar configuración), realice una de las siguientes operaciones:

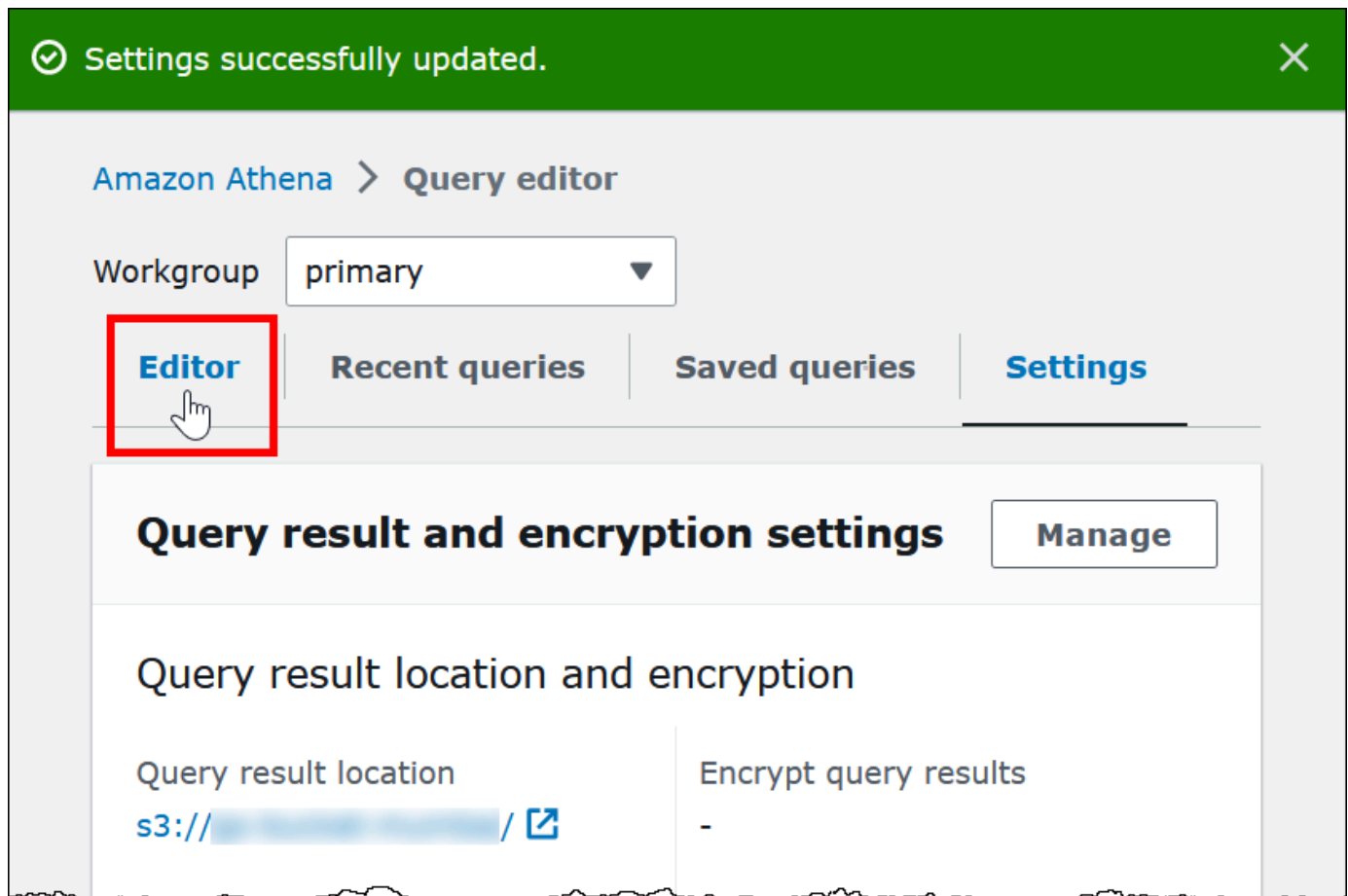
- En el cuadro Location of query result (Ubicación del resultado de la consulta), ingrese la ruta de acceso al bucket que creó en Amazon S3 para los resultados de la consulta. Prefije la ruta con `s3://`.
- Elija Browse S3 (Navegar S3), elija el bucket de Amazon S3 que creó para su región actual y, a continuación, elija Choose (Elegir).



The screenshot shows a 'Manage settings' dialog box with the following elements:

- Manage settings** (Section header)
- Query result location and encryption** (Section header)
- Location of query result** (Label)
- (Text input field with a magnifying glass icon)
- (Button with an external link icon)
- (Button)
- (Button)
- (Button)

6. Elija Guardar.
7. Elija Editor para cambiar al editor de consultas.



## Creación de una base de datos

Una vez que haya configurado una ubicación de resultados de consultas, crear una base de datos en el editor de consultas de la consola de Athena es muy sencillo.

Para crear una base de datos mediante el editor de consultas de Athena

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. En la pestaña Editor, en el editor de consultas, escriba el comando DDL (lenguaje de definición de datos) de Hive `CREATE DATABASE myDataBase`. Reemplace *myDataBase* por el nombre que quiera usar. Para obtener restricciones sobre los nombres de bases de datos, consulte [Nombres de tablas, bases de datos y columnas](#).
3. Elija Run (Ejecutar) o pulse **Ctrl+ENTER**.
4. Para convertir la base de datos en la base de datos actual, selecciónela del menú Database (Base de datos), a la izquierda del editor de consultas.

Para obtener información sobre cómo controlar permisos para bases de datos de Athena, consulte [Acceso detallado a las bases de datos y las tablas en AWS Glue Data Catalog](#).

## Creación de tablas en Athena

Puede ejecutar instrucciones DDL en la consola de Athena mediante un controlador ODBC o JDBC, o mediante el [formulario Add Table](#) (Agregar tabla) de Athena.

Cuando se usa Athena para crear un nuevo esquema de tabla, Athena lo almacena en un catálogo de datos y lo usa al ejecutar consultas.

Athena utiliza un método conocido como esquema en lectura, que implica la proyección de un esquema sobre los datos en el momento de ejecutar la consulta. Así no es necesaria la carga ni la transformación de los datos.

Athena no modifica los datos en Amazon S3.

Athena utiliza Apache Hive para definir las tablas y crear bases de datos, que son básicamente un espacio de nombres lógico de tablas.

Al crear una base de datos y una tabla en Athena, solo describe el esquema y la ubicación donde se encuentran los datos de la tabla en Amazon S3 para efectuar la consulta en el momento de la lectura. Por lo tanto, base de datos y tabla tienen un significado ligeramente distinto al habitual en los sistemas de bases de datos relacionales tradicionales, ya que los datos no se almacenan junto con la definición del esquema de la base de datos y la tabla.

Cuando se realiza la consulta, se aplica SQL estándar a la tabla y los datos se leen en ese momento. En la [documentación de Apache Hive](#), encontrará información para crear bases de datos y tablas, pero la información siguiente lo orientará específicamente para Athena.

La longitud máxima de la cadena de consulta es de 256 KB.

Hive admite diversos formatos de datos mediante el uso de bibliotecas serializadoras-deserializadoras (SerDe). También puede definir esquemas complejos con expresiones regulares. Para obtener una lista de las bibliotecas SerDe compatibles, consulte [Formatos de datos y SerDes compatibles](#).

## Consideraciones y limitaciones

A continuación, se indican algunas limitaciones y consideraciones importantes para las tablas de Athena.



## Requisitos para las tablas en Athena y los datos en Amazon S3

Al crear una tabla, debe especificar la ubicación de un bucket de Amazon S3 para los datos subyacentes con la cláusula `LOCATION`. Considere lo siguiente:

- Athena solo puede consultar la versión más reciente de datos en un bucket de Amazon S3 con control de versiones y no puede consultar versiones anteriores de los datos.
- Debe disponer de los permisos adecuados para trabajar con los datos de la ubicación de Amazon S3. Para obtener más información, consulte [Acceso a Amazon S3](#).
- Athena permite consultar los objetos almacenados con varias clases de almacenamiento en el mismo bucket especificado por la cláusula `LOCATION`. Por ejemplo, puede consultar los datos en objetos que se almacenan en diferentes clases de almacenamiento (Standard y Standard-IA e Intelligent-Tiering) en Amazon S3.
- Athena admite los [buckets de pago por solicitante](#). Para obtener información sobre cómo habilitar pagos por solicitante para buckets con datos de origen que desea consultar en Athena, consulte [Creación de un grupo de trabajo](#).
- Athena no es compatible con la consulta de los datos en las clases de almacenamiento [S3 Glacier Flexible Retrieval](#) ni S3 Glacier Deep Archive. Se ignoran los objetos de las clases de almacenamiento S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive. Como alternativa, puede utilizar la clase de almacenamiento Amazon S3 Glacier Instant Retrieval, que Athena puede consultar. Para obtener más información, consulte [Amazon S3 Glacier instant retrieval storage class](#) (Clase de almacenamiento Amazon S3 Glacier Instant Retrieval).

Para obtener información sobre las clases de almacenamiento, consulte [Clases de almacenamiento](#), [Cambio de la clase de almacenamiento de un objeto en Amazon S3](#), [Transición a la clase de almacenamiento GLACIER \(archivo de objetos\)](#) y [Buckets de pago por solicitante](#) en la Guía del usuario de Amazon Simple Storage Service.

- Si efectúa consultas de los buckets de S3 con un gran número de objetos y los datos no están divididos en particiones, dichas consultas pueden afectar a los límites de tasa de solicitudes Get en Amazon S3 y dar lugar a excepciones de Amazon S3. Para evitar errores, divida los datos. Plantéese también ajustar las tasas de solicitud de Amazon S3. Para obtener más información, consulte [Velocidad de solicitudes y consideraciones sobre el rendimiento](#).
- Si utiliza la operación de la API [CreateTable](#) de AWS Glue o la plantilla [AWS::Glue::Table](#) de AWS CloudFormation para crear una tabla para su uso en Athena sin especificar la propiedad `TableType`, y luego ejecuta una consulta DDL como `SHOW CREATE TABLE` o `MSCK REPAIR`

TABLE, puede recibir el mensaje de error FAILED: NullPointerException Name is null (ERROR: el nombre NullPointerException es nulo).

Para resolver el error, especifique un valor para el atributo `TableType` [TableInput](#) como parte de la llamada a la API `CreateTable` de AWS Glue o la [plantilla AWS CloudFormation](#). Entre los valores posibles para `TableType`, se incluyen `EXTERNAL_TABLE` o `VIRTUAL_VIEW`.

Este requisito se aplica únicamente cuando se crea una tabla mediante la operación de la API `CreateTable` de AWS Glue o la plantilla `AWS::Glue::Table`. Si crea una tabla para Athena mediante una instrucción DDL o un rastreador de AWS Glue, la propiedad `TableType` se define automáticamente.

## Funciones compatibles

Las funciones admitidas en las consultas de Athena corresponden a las de Trino y Presto. Para obtener más información sobre funciones individuales, consulte la sección de funciones y operadores en la documentación de [Trino](#) o [Presto](#).

No se admiten transformaciones de datos transaccionales

Athena no admite operaciones basadas en transacciones (como las de Hive o Presto) para los datos de una tabla. Encontrará una lista completa de palabras clave no admitidas en [Unsupported DDL](#).

Las operaciones que cambian el estado de las tablas son ACID.

Al crear, actualizar o eliminar tablas, está garantizado que tales operaciones cumplen con ACID. Por ejemplo, si varios usuarios o clientes intentan crear una tabla o modificar una existente al mismo tiempo, solo uno lo conseguirá.

Las tablas son EXTERNAL

Utilice siempre la palabra clave `EXTERNAL` excepto al crear tablas de [Iceberg](#). Si utiliza `CREATE TABLE` sin la palabra clave `EXTERNAL` para tablas que no son Iceberg, Athena emite un error. Al eliminar una tabla en Athena, solo desaparecen los metadatos de la tabla. Los datos permanecen en Amazon S3.

## Creación de tablas mediante AWS Glue o la consola de Athena

Puede crear tablas en Athena mediante AWS Glue, el formulario para agregar tablas, o ejecutando una instrucción DDL en el editor de consultas de Athena.

## Para crear una tabla con el rastreador de AWS Glue

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. En el editor de consultas, junto a Tablas y vistas, elija Crear y, a continuación, Rastreador de AWS Glue.
3. Siga los pasos de la página Add crawler (Agregar rastreador) de la consola de AWS Glue para agregar un rastreador.

Para obtener más información, consulte [Uso de rastreadores AWS Glue](#).

## Para crear una tabla mediante el formulario para crear tablas de Athena

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. En el editor de consultas, junto a Tables and views (Tablas y vistas), elija Create (Crear) y, a continuación, S3 bucket data (Datos del bucket de S3).
3. En el formulario Create Table From S3 bucket data (Crear tabla a partir de datos de buckets de S3), ingrese la información para crear la tabla y, a continuación, elija Create table (Crear tabla). Para obtener más información acerca de los campos del formulario, consulte [Adición de una tabla mediante un formulario](#).

## Para crear una tabla con DDL de Hive

1. En el menú Database (Base de datos), elija la base de datos para la que desea crear una tabla. Si no especifica una base de datos en la instrucción CREATE TABLE, la tabla se crea en la base de datos seleccionada actualmente en el editor de consultas.
2. Ingrese una instrucción como la siguiente en el editor de consultas y, a continuación, elija Run (Ejecutar) o pulse **Ctrl+ENTER**.

```
CREATE EXTERNAL TABLE IF NOT EXISTS cloudfront_logs (  
    `Date` Date,  
    Time STRING,  
    Location STRING,  
    Bytes INT,  
    RequestIP STRING,  
    Method STRING,  
    Host STRING,  
    Uri STRING,  
    Status INT,
```



## Requisitos de base de datos, tablas y columnas

- Los caracteres aceptables para los nombres de base de datos, los nombres de tablas y los nombres de columnas en AWS Glue deben ser una cadena UTF-8. La cadena debe ser 1 byte de largo como mínimo y 255 bytes como máximo. Si se supera este límite, se genera el error El valor en “nombre” no satisface la restricción: el miembro debe tener una longitud menor o igual a 255. Los caracteres que se pueden usar incluyen espacios y se definen mediante el siguiente patrón de cadena de una sola línea:

```
[\\u0020-\\uD7FF\\uE000-\\uFFFF\\uD800\\uDC00-\\uDBFF\\uDFFF\\t]*
```

- Actualmente, el patrón de expresiones regulares AWS Glue permite añadir espacios iniciales al principio de los nombres. Dado que estos espacios iniciales pueden ser difíciles de detectar y pueden provocar problemas de usabilidad tras su creación, evite crear nombres de objetos que tengan espacios iniciales.
- Si usa una plantilla [AWS::Glue::Database](#) de AWS CloudFormation para crear una base de datos de AWS Glue y no especifica ningún nombre de la base de datos, AWS Glue genera automáticamente un nombre de la base de datos con el formato *nombre\_recurso-cadena\_aleatoria* que no es compatible con Athena.
- Puede utilizar el administrador de catálogo de AWS Glue para cambiar el nombre de las columnas, pero no los nombres de las tablas ni los nombres de bases de datos. Para evitar esta limitación, debe usar una definición de la base de datos antigua para crear una base de datos con el nombre nuevo. A continuación, utilice las definiciones de las tablas de la base de datos antigua para volver a crear las tablas en la base de datos nueva. Para ello, puede utilizar la AWS CLI o el SDK de AWS Glue. Para ver los pasos, consulte [Uso de la AWS CLI para recrear una base de datos de AWS Glue y sus tablas](#).

## Utilice minúsculas para los nombres de tablas y columnas de las tablas de Athena

Athena acepta mayúsculas y minúsculas en consultas DDL y DML, pero minúsculas en los nombres cuando ejecuta la consulta. Por este motivo, evite el uso de mayúsculas y minúsculas para nombres de tablas o columnas, y no confíe en el uso de mayúsculas y minúsculas solamente en Athena para distinguir dichos nombres. Por ejemplo, si utiliza una instrucción DDL para crear una columna denominada Castle, la columna creada irá en minúsculas en castle. Si, a continuación, especifica el nombre de columna en una consulta DML como Castle o CASTLE, Athena usará minúsculas en el nombre para que ejecute la consulta, pero mostrará el encabezado de columna utilizando el modo que eligió en la consulta.

Los nombres de base de datos, tablas y columnas deben tener 255 caracteres o menos.

## Nombres que comienzan con un guion bajo

Al crear tablas, utilice acentos graves para entrecomillar los nombres de las tablas, vistas o columnas que empiecen por un guion bajo. Por ejemplo:

```
CREATE EXTERNAL TABLE IF NOT EXISTS `_myunderscoretable`(  
  `_id` string, `_index` string)  
LOCATION 's3://DOC-EXAMPLE-BUCKET/'
```

## Nombres de tabla, vista o columna que comienzan por números

Al ejecutar consultas SELECT, CTAS o VIEW, ponga comillas alrededor de identificadores como los nombres de tabla, vista o columna que comienzan por un dígito. Por ejemplo:

```
CREATE OR REPLACE VIEW "123view" AS  
SELECT "123columnone", "123columntwo"  
FROM "234table"
```

## Nombres de columna y tipos complejos

Para los tipos complejos, solo están permitidos caracteres alfanuméricos, guion bajo (\_) y período (.) en los nombres de las columnas. Para crear una tabla y mapeos para claves con caracteres restringidos, puede utilizar una instrucción DDL personalizada. Para obtener más información, consulte el artículo [Crear tablas en Amazon Athena a partir de JSON anidado y asignaciones por medio de JSONSerDe](#) en el Blog de macrodatos de AWS.

## Palabras reservadas

Ciertas palabras reservadas de Athena deben tener caracteres de escape. Para entrecomillar las palabras reservadas en instrucciones DDL, utilice acentos graves (`) como carácter de escape. Para entrecomillar palabras reservadas en instrucciones SQL SELECT y en consultas en [vistas](#), utilice dobles comillas (") como carácter de escape.

Para obtener más información, consulte [Palabras clave reservadas](#).

## Recursos adicionales

Para ver la sintaxis de creación de bases de datos y tablas completa, consulte las siguientes páginas.

- [CREATE DATABASE](#)
- [CREATE TABLE](#)

Para obtener más información sobre las bases de datos y las tablas en AWS Glue, consulte [Bases de datos](#) y [Tablas](#) en la Guía para desarrolladores de AWS Glue.

## Palabras clave reservadas

Cuando se ejecutan consultas en Athena que incluyen palabras clave reservadas, debe aplicarles escape delimitándolas con caracteres especiales. Utilice las listas de este tema para verificar qué palabras clave están reservadas en Athena.

Para entrecomillar las palabras reservadas en instrucciones DDL, utilice acentos graves (‘) como carácter de escape. Para entrecomillar palabras reservadas en instrucciones SQL SELECT y en consultas en [vistas](#), utilice dobles comillas (”) como carácter de escape.

- [Lista de palabras reservadas en instrucciones DDL](#)
- [Lista de palabras reservadas en instrucciones SQL SELECT](#)
- [Ejemplos de consultas con palabras reservadas](#)

## Lista de palabras reservadas en instrucciones DDL

Athena utiliza la siguiente lista de palabras clave reservadas en sus instrucciones DDL. Si las utiliza sin aplicarles escape, Athena emite un error. Para entrecomillarlas, utilice el acento grave (‘) como carácter de escape.

No puede utilizar palabras clave reservadas de DDL como nombres de identificadores en instrucciones DDL sin entrecomillarlas con acentos graves (‘).

```
ALL, ALTER, AND, ARRAY, AS, AUTHORIZATION, BETWEEN, BIGINT,
BINARY, BOOLEAN, BOTH, BY, CASE, CASHE, CAST, CHAR, COLUMN,
CONF, CONSTRAINT, COMMIT, CREATE, CROSS, CUBE, CURRENT,
CURRENT_DATE, CURRENT_TIMESTAMP, CURSOR, DATABASE, DATE,
DAYOFWEEK, DECIMAL, DELETE, DESCRIBE, DISTINCT, DOUBLE, DROP,
ELSE, END, EXCHANGE, EXISTS, EXTENDED, EXTERNAL, EXTRACT,
FALSE, FETCH, FLOAT, FLOOR, FOLLOWING, FOR, FOREIGN, FROM,
FULL, FUNCTION, GRANT, GROUP, GROUPING, HAVING, IF, IMPORT,
IN, INNER, INSERT, INT, INTEGER, INTERSECT, INTERVAL, INTO,
```

```
IS, JOIN, LATERAL, LEFT, LESS, LIKE, LOCAL, MACRO, MAP, MORE,
NONE, NOT, NULL, NUMERIC, OF, ON, ONLY, OR, ORDER, OUT,
OUTER, OVER, PARTIALSCAN, PARTITION, PERCENT, PRECEDING,
PRECISION, PRESERVE, PRIMARY, PROCEDURE, RANGE, READS,
REDUCE, REGEXP, REFERENCES, REVOKE, RIGHT, RLIKE, ROLLBACK,
ROLLUP, ROW, ROWS, SELECT, SET, SMALLINT, START, TABLE,
TABLESAMPLE, THEN, TIME, TIMESTAMP, TO, TRANSFORM, TRIGGER,
TRUE, TRUNCATE, UNBOUNDED, UNION, UNIQUEJOIN, UPDATE, USER,
USING, UTC_TIMESTAMP, VALUES, VARCHAR, VIEWS, WHEN, WHERE,
WINDOW, WITH
```

## Lista de palabras reservadas en instrucciones SQL SELECT

Athena utiliza la siguiente lista de palabras clave reservadas en las instrucciones SQL SELECT y en las consultas de vistas.

Si utiliza estas palabras clave como identificadores, debe entrecomillarlas con comillas dobles (") en las instrucciones de la consulta.

```
ALTER, AND, AS, BETWEEN, BY, CASE, CAST, CONSTRAINT, CREATE,
CROSS, CUBE, CURRENT_CATALOG, CURRENT_DATE, CURRENT_PATH,
CURRENT_SCHEMA, CURRENT_TIME, CURRENT_TIMESTAMP, CURRENT_USER,
DEALLOCATE, DELETE, DESCRIBE, DISTINCT, DROP, ELSE, END, ESCAPE,
EXCEPT, EXECUTE, EXISTS, EXTRACT, FALSE, FIRST, FOR, FROM,
FULL, GROUP, GROUPING, HAVING, IN, INNER, INSERT, INTERSECT,
INTO, IS, JOIN, JSON_ARRAY, JSON_EXISTS, JSON_OBJECT,
JSON_QUERY, JSON_TABLE, JSON_VALUE, LAST, LEFT, LIKE,
LISTAGG, LOCALTIME, LOCALTIMESTAMP, NATURAL, NORMALIZE,
NOT, NULL, OF, ON, OR, ORDER, OUTER, PREPARE, RECURSIVE, RIGHT,
ROLLUP, SELECT, SKIP, TABLE, THEN, TRIM, TRUE, UESCAPE, UNION,
UNNEST, USING, VALUES, WHEN, WHERE, WITH
```

## Ejemplos de consultas con palabras reservadas

La consulta del siguiente ejemplo utiliza acentos graves (`) para aplicar escape a las palabras clave reservadas relacionadas con DDL `partition` y `date` que se utilizan para un nombre de tabla y uno de los nombres de las columnas:

```
CREATE EXTERNAL TABLE `partition` (
  `date` INT,
  col2 STRING
)
```



```
PARTITIONED BY (year STRING)
STORED AS TEXTFILE
LOCATION 's3://DOC-EXAMPLE-BUCKET/test_examples/';
```

Las consultas del siguiente ejemplo incluyen un nombre de columna que contiene las palabras clave reservadas relacionadas con DDL en instrucciones ALTER TABLE ADD PARTITION y ALTER TABLE DROP PARTITION. Las palabras clave reservadas de DDL se encierran en acentos graves ('):

```
ALTER TABLE test_table
ADD PARTITION (`date` = '2018-05-14')
```

```
ALTER TABLE test_table
DROP PARTITION (`partition` = 'test_partition_value')
```

La siguiente consulta de ejemplo incluye una palabra clave reservada (fin) como identificador en una instrucción SELECT. Se aplica escape a la palabra clave con comillas dobles:

```
SELECT *
FROM TestTable
WHERE "end" != nil;
```

La siguiente consulta de ejemplo incluye una palabra clave reservada (primera) en una instrucción SELECT. Se aplica escape a la palabra clave con comillas dobles:

```
SELECT "itemId"."first"
FROM testTable
LIMIT 10;
```

## Ubicación de las tablas en Simple Storage Service (Amazon S3)

Al ejecutar una consulta CREATE TABLE en Athena, Athena registra la tabla en el catálogo de datos de AWS Glue, que es donde Athena almacena los metadatos.

Para especificar la ruta a los datos en Amazon S3, utilice la propiedad LOCATION, como se muestra en el ejemplo siguiente:

```
CREATE EXTERNAL TABLE `test_table` (
```

```

...
)
ROW FORMAT ...
STORED AS INPUTFORMAT ...
OUTPUTFORMAT ...
LOCATION s3://DOC-EXAMPLE-BUCKET/folder/

```

- Para obtener información sobre la nomenclatura de buckets, consulte [Restricciones y limitaciones de los buckets](#) en la Guía del usuario de Amazon Simple Storage Service.
- Para obtener información sobre el uso de carpetas en Simple Storage Service (Amazon S3), consulte [Uso de carpetas](#) en la Guía del usuario de Amazon Simple Storage Service.

La LOCATION en Amazon S3 especifica todos los archivos que representan la tabla.

#### Important

Athena lee todos los datos almacenados en la carpeta de Amazon S3 que especifique. Si tiene datos que no desea que Athena lea, no almacene los datos en la misma carpeta de Amazon S3 que los datos que desea que Athena lea. Si utiliza particiones, a fin de garantizar que Athena analice los datos dentro de una partición, el filtro WHERE debe incluir la partición. Para obtener más información, consulte [Ubicación y particiones de las tablas](#).

Cuando especifique LOCATION en la instrucción CREATE TABLE, utilice las siguientes directrices:

- Use una barra diagonal final.
- Puede utilizar una ruta a una carpeta de Amazon S3 o a un alias de punto de acceso de Amazon S3. Para obtener información acerca de los alias de punto de acceso de Amazon S3, consulte [Uso de un alias de estilo bucket para el punto de acceso](#) en la Guía del usuario de Amazon S3.

Uso:

```
s3://DOC-EXAMPLE-BUCKET/folder/
```

```
s3://DOC-EXAMPLE-BUCKET-metadata-s3alias/folder/
```

No utilice ninguno de los siguientes elementos para especificar la LOCATION de los datos.

- No utilice nombres de archivos, guiones bajos, comodines o patrones de glob para especificar ubicaciones de archivos.
- No agregue toda la notación HTTP como, por ejemplo, `s3.amazonaws.com`, a la ruta del bucket de Amazon S3.
- No utilice carpetas vacías como `//` en la ruta, de la siguiente manera: `S3://DOC-EXAMPLE-BUCKET/folder//folder/`. Aunque se trata de una ruta de Amazon S3 válida, Athena no lo permite y lo cambia a `s3://DOC-EXAMPLE-BUCKET/folder/folder/`, y elimina el `/` adicional.

No utilizar:

```
s3://DOC-EXAMPLE-BUCKET
s3://DOC-EXAMPLE-BUCKET/*
s3://DOC-EXAMPLE-BUCKET/mySpecialFile.dat
s3://DOC-EXAMPLE-BUCKET/prefix/filename.csv
s3://DOC-EXAMPLE-BUCKET.s3.amazonaws.com
S3://DOC-EXAMPLE-BUCKET/prefix//prefix/
arn:aws:s3:::DOC-EXAMPLE-BUCKET/prefix
s3://arn:aws:s3:<region>:<account_id>:accesspoint/<accesspointname>
https://<accesspointname>-<number>.s3-accesspoint.<region>.amazonaws.com
```

## Ubicación y particiones de las tablas

Los datos de origen pueden agruparse en carpetas de Amazon S3 denominadas particiones en función de un conjunto de columnas. Por ejemplo, estas columnas pueden representar el año, el mes y el día en particular en que se creó el registro.

Al crear una tabla, puede elegir hacerlo particionado. Cuando Athena ejecuta una consulta SQL en una tabla con particiones, utiliza la propiedad `LOCATION` de la definición de tabla como la ruta base para enumerar y, a continuación, analizar todos los archivos disponibles. Sin embargo, antes de que se pueda consultar una tabla con particiones, debe actualizar el catálogo de datos de AWS Glue con información de las particiones. Esta información representa el esquema de archivos dentro de la partición y la `LOCATION` de los archivos en Amazon S3 para la partición.

- Para obtener información sobre cómo el rastreador de AWS Glue agrega particiones, consulte [¿Cómo determina un rastreador cuándo crear particiones?](#) en la Guía para desarrolladores de AWS Glue.
- Para obtener información sobre cómo configurar el rastreador de forma que cree tablas para los datos de las particiones existentes, consulte [Uso de varios orígenes de datos con rastreadores](#).

- También puede crear particiones en una tabla directamente en Athena. Para obtener más información, consulte [Particiones de datos en Athena](#).

Cuando Athena ejecuta una consulta en una tabla con particiones, comprueba si alguna de las columnas con particiones se usa en la cláusula `WHERE` de la consulta. Si se utilizan columnas con particiones, Athena solicita al catálogo de datos de AWS Glue que devuelva la especificación de partición coincidente con las columnas con particiones especificadas. La especificación de partición incluye la propiedad `LOCATION`, que indica a Athena qué prefijo de Amazon S3 debe utilizar al leer los datos. En este caso, solo se analizan los datos almacenados en este prefijo. Si no utiliza columnas con particiones en la cláusula `WHERE`, Athena analiza todos los archivos que pertenecen a las particiones de la tabla.

Si desea ver ejemplos de cómo utilizar particiones con Athena para mejorar el rendimiento de las consultas y reducir los costos de consulta, consulte [Los 10 consejos principales para el ajuste del rendimiento de Amazon Athena](#).

## Formatos de almacenamiento en columnas

[Apache Parquet](#) y [ORC](#) son formatos de almacenamiento en columnas que están optimizados para una rápida recuperación de los datos y que se utilizan en las aplicaciones de análisis de AWS.

Los formatos de almacenamiento en columnas tienen las siguientes características que los hacen idóneos para su uso con Athena:

- Compresión por columna, con el algoritmo de compresión seleccionado para cada tipo de datos de columna para ahorrar espacio de almacenamiento en Amazon S3 y reducir el espacio de disco y las operaciones de E/S durante el procesamiento de consultas.
- La inserción de predicados en Parquet y ORC permite que las consultas de Athena solo obtengan los bloques necesarios, lo que mejora el rendimiento de las consultas. Cuando una consulta de Athena obtiene valores de columna específicos de sus datos, utiliza las estadísticas de los predicados de bloque de datos, como los valores máximos o mínimos, para determinar si se debe leer u omitir el bloque.
- La división de datos en Parquet y ORC permite a Athena dividir la lectura de los datos entre varios lectores y aumentar el paralelismo durante el procesamiento de consultas.

Para convertir sus datos sin procesar existentes de otros formatos de almacenamiento a Parquet u ORC, puede ejecutar [CREATE TABLE AS SELECT \(CTAS\)](#) en las consultas de Athena y especificar un formato de almacenamiento de datos como Parquet u ORC, o utilizar el rastreador de AWS Glue.

## Cómo elegir entre Parquet y ORC

La elección entre ORC (Optimized Row Columnar) y Parquet depende de sus requisitos de uso específicos.

Apache Parquet proporciona esquemas eficientes de compresión y codificación de datos y es ideal para ejecutar consultas complejas y procesar grandes cantidades de datos. Parquet está optimizado para su uso con [Apache Arrow](#), lo que puede resultar ventajoso si utiliza herramientas relacionadas con Arrow.

ORC proporciona una forma eficiente de almacenar los datos de Hive. Los archivos ORC suelen ser más pequeños que los archivos Parquet, y los índices ORC pueden agilizar las consultas. Además, ORC admite tipos complejos, como estructuras, mapas y listas.

Cuando elija entre Parquet y ORC, tenga en cuenta los siguientes factores:

**Rendimiento de consultas:** dado que Parquet admite una gama más amplia de tipos de consultas, Parquet podría ser una mejor opción si planea realizar consultas complejas.

**Tipos de datos complejos:** si utiliza tipos de datos complejos, ORC podría ser una mejor opción, ya que admite una gama más amplia de tipos de datos complejos.

**Tamaño de archivo:** si el espacio en disco es un problema, ORC suele producir archivos más pequeños, lo que puede reducir los costos de almacenamiento.

**Compresión:** tanto Parquet como ORC proporcionan una buena compresión, pero el mejor formato dependerá de su caso de uso específico.

**Evolución:** tanto Parquet como ORC admiten la evolución del esquema, lo que significa que puede agregar, eliminar o modificar columnas a lo largo del tiempo.

Tanto Parquet como ORC son buenas opciones para aplicaciones de macrodatos, pero tenga en cuenta los requisitos de su escenario antes de elegir. Es posible que desee realizar pruebas comparativas de sus datos y consultas para ver qué formato funciona mejor en su caso de uso.

## Conversión a formatos de columnas

El rendimiento de las consultas de Amazon Athena mejora si convierte los datos a formatos de columnas de código abierto, como [Apache Parquet](#) u [ORC](#).

Las opciones para convertir fácilmente los datos de origen, como JSON o CSV, a un formato de columnas incluyen el uso de consultas [CREATE TABLE AS](#) o ejecución de trabajos en AWS Glue.

- Puede usar consultas CREATE TABLE AS (CTAS) para convertir datos a Parquet u ORC en un solo paso. Para ver un ejemplo, consulte [Ejemplo: escritura de los resultados de la consulta en un formato diferente](#) en la página [Ejemplos de consultas CTAS](#).
- Para obtener información sobre cómo ejecutar un trabajo de AWS Glue para transformar datos en CSV a Parquet, consulte la sección “Transformar datos en formato CSV a Parquet” en la publicación del blog sobre macrodatos de AWS [Build a Data Lake Foundation with AWS Glue and Amazon S3](#). AWS Glue admite el uso de la misma técnica para convertir datos en CSV a ORC, o datos en JSON a Parquet u ORC.

## Particiones de datos en Athena

La partición de los datos le permite restringir el volumen de datos que explora cada consulta, lo que mejora el rendimiento y reduce los costos. Puede particionar datos por cualquier clave. Una práctica común consiste en particionar los datos en función del tiempo, lo que a menudo conduce a un esquema de partición de varios niveles. Por ejemplo, si un cliente recibe datos cada hora, podría elegir una partición por año, mes, día y hora. Otro cliente, que recibe datos procedentes de muchos orígenes distintos, pero que solo se cargan una vez al día, puede basar la partición en un identificador del origen de datos y en la fecha.

Athena puede utilizar particiones de estilo de Apache Hive, cuyas rutas de datos contienen pares de valores de clave conectados por signos de igual (por ejemplo, `country=us/. . .` o `year=2021/month=01/day=26/. . .`). Por lo tanto, las rutas incluyen los nombres de las claves de partición y los valores que representa cada ruta. Para cargar nuevas particiones de Hive en una tabla particionada, puede utilizar el comando [MSCK REPAIR TABLE](#), que solo funciona con particiones de estilo de Hive.

Athena también puede utilizar esquemas de partición de estilos que no son de Hive. Por ejemplo, los registros de CloudTrail y los flujos de entrega de Firehose utilizan componentes de ruta independientes para partes de fecha, como `data/2021/01/26/us/6fc7845e.json`. Para

aquellas particiones que no son del estilo de Hive, se utiliza [ALTER TABLE ADD PARTITION](#) para agregar las particiones manualmente.

## Consideraciones y limitaciones

Al usar la partición, tenga en cuenta los siguientes puntos:

- Si consulta una tabla con particiones y especifica la partición en la cláusula WHERE, Athena analiza los datos solo de esa partición. Para obtener más información, consulte [Ubicación y particiones de las tablas](#).
- Si efectúa consultas de los buckets de Amazon S3 con un gran número de objetos y los datos no están particionados, dichas consultas pueden afectar los límites de tasa de solicitudes GET en Amazon S3 y dar lugar a excepciones de Amazon S3. Para evitar errores, divida los datos. Plantéese también ajustar las tasas de solicitud de Amazon S3. Para obtener más información, consulte [Patrones de diseño de prácticas recomendadas: optimización del rendimiento de Amazon S3](#).
- Las ubicaciones de partición que se van a utilizar con Athena deben utilizar el protocolo s3 (por ejemplo, s3://DOC-EXAMPLE-BUCKET/*folder*/). En Athena, las ubicaciones que utilizan otros protocolos (por ejemplo, s3a://DOC-EXAMPLE-BUCKET/*folder*/) producirán errores de consulta cuando se ejecuten consultas MSCK REPAIR TABLE en las tablas que contienen.
- Asegúrese de que la ruta de Amazon S3 esté en minúsculas, en lugar de usar también mayúsculas en la inicial (por ejemplo, `userid` en lugar de `userId`). Si la ruta de S3 usa mayúsculas en la inicial, MSCK REPAIR TABLE no agrega las particiones a AWS Glue Data Catalog. Para obtener más información, consulte [MSCK REPAIR TABLE](#).
- Dado que MSCK REPAIR TABLE analiza una carpeta y sus subcarpetas para encontrar un esquema de partición coincidente, asegúrese de mantener los datos de tablas separadas en jerarquías de carpetas separadas. Por ejemplo, supongamos que tenemos los datos de la tabla 1 en s3://DOC-EXAMPLE-BUCKET1 y los datos de la tabla 2 en s3://DOC-EXAMPLE-BUCKET1/table-2-data. Si ambas tablas están particionadas por cadena, MSCK REPAIR TABLE añadirá las particiones de la tabla 2 a la tabla 1. Para evitarlo, utilice en su lugar estructuras de carpetas independientes, como s3://DOC-EXAMPLE-BUCKET1 y s3://DOC-EXAMPLE-BUCKET2. Tenga en cuenta que este comportamiento es coherente con Amazon EMR y Apache Hive.
- Si actualmente utiliza AWS Glue Data Catalog con Athena, consulte los [puntos de conexión y cuotas de AWS Glue](#) para ver las cuotas de servicio de las particiones por cuenta y por tabla.
  - Aunque Athena admite consultas de tablas de AWS Glue que tienen 10 millones de particiones, no puede leer más de 1 millón de particiones en un solo escaneo. En estos escenarios, la

indexación de particiones puede ser beneficiosa. Para obtener más información, consulte el artículo del Blog de macrodatos de AWS [Improve Amazon Athena query performance using AWS Glue Data Catalog partition indexes](#).

- Para solicitar un aumento de la cuota de particiones si está utilizando el AWS Glue Data Catalog, vaya a la [consola de Service Quotas de AWS Glue](#).

## Creación y carga de una tabla con datos divididos

Para crear una tabla que utiliza particiones, utilice la cláusula `PARTITIONED BY` de la instrucción [CREATE TABLE](#). La cláusula `PARTITIONED BY` define las claves en las que se llevará a cabo la partición de los datos, como en el siguiente ejemplo. La cláusula `LOCATION` especifica la ubicación raíz de los datos particionados.

```
CREATE EXTERNAL TABLE users (  
  first string,  
  last string,  
  username string  
)  
PARTITIONED BY (id string)  
STORED AS parquet  
LOCATION 's3://DOC-EXAMPLE-BUCKET'
```

Después de crear la tabla, cargará los datos en las particiones para poder consultarlos. Para las particiones de estilo Hive, se ejecuta [MSCK REPAIR TABLE](#). Para las particiones que no son del estilo de Hive, se utiliza [ALTER TABLE ADD PARTITION](#) para agregar las particiones manualmente.

## Preparación de datos de estilo de Hive y no compatibles con Hive para consultas

En las siguientes secciones, se muestra cómo preparar datos de estilo de Hive y no compatibles con Hive para consultas en Athena.

### Escenario 1: datos almacenados en Amazon S3 en formato Hive

En este escenario, las particiones están almacenadas en Amazon S3 en carpetas distintas. Por ejemplo, esta es una lista parcial de muestras de impresiones de anuncios que genera el comando [aws s3 ls](#), que enumera los objetos de S3 con un prefijo especificado:

```
aws s3 ls s3://elasticmapreduce/samples/hive-ads/tables/impressions/
```



```
PRE dt=2009-04-12-13-00/  
PRE dt=2009-04-12-13-05/  
PRE dt=2009-04-12-13-10/  
PRE dt=2009-04-12-13-15/  
PRE dt=2009-04-12-13-20/  
PRE dt=2009-04-12-14-00/  
PRE dt=2009-04-12-14-05/  
PRE dt=2009-04-12-14-10/  
PRE dt=2009-04-12-14-15/  
PRE dt=2009-04-12-14-20/  
PRE dt=2009-04-12-15-00/  
PRE dt=2009-04-12-15-05/
```

Aquí los registros se almacenan con el nombre de columna (dt) definido según incrementos de fecha, hora y minuto. Cuando se especifica DDL con la ubicación de la carpeta principal, el esquema y el nombre de la columna particionada, Athena puede consultar los datos de las subcarpetas.

## Creación de la tabla

Para obtener una tabla de estos datos, cree una partición basada en “dt”, como en la siguiente instrucción DDL de Athena:

```
CREATE EXTERNAL TABLE impressions (  
    requestBeginTime string,  
    adId string,  
    impressionId string,  
    referrer string,  
    userAgent string,  
    userCookie string,  
    ip string,  
    number string,  
    processId string,  
    browserCookie string,  
    requestEndTime string,  
    timers struct<modelLookup:string, requestTime:string>,  
    threadId string,  
    hostname string,  
    sessionId string)  
PARTITIONED BY (dt string)  
ROW FORMAT serde 'org.apache.hive.hcatalog.data.JsonSerDe'  
LOCATION 's3://elasticmapreduce/samples/hive-ads/tables/impressions/' ;
```

Esta tabla utiliza el serializador-deserializador JSON nativo de Hive para leer los datos JSON almacenados en Amazon S3. Para obtener más información acerca de los formatos admitidos, consulte [Formatos de datos y SerDes compatibles](#).

## Ejecución de MSCK REPAIR TABLE

Después de ejecutar la consulta CREATE TABLE, ejecute el comando MSCK REPAIR TABLE del editor de consultas de Athena para cargar las particiones, como en el siguiente ejemplo.

```
MSCK REPAIR TABLE impressions
```

Cuando ejecuta este comando, los datos están listos para consultarlos.

## Consulta de los datos

Consulte los datos de la tabla de impresiones utilizando la columna de partición. A continuación se muestra un ejemplo:

```
SELECT dt,impressionid FROM impressions WHERE dt<'2009-04-12-14-00' and
dt>='2009-04-12-13-00' ORDER BY dt DESC LIMIT 100
```

La consulta deberá mostrar resultados similares a los siguientes:

```
2009-04-12-13-20    ap3HcVKAWfXtgIPu6WpuUfAfL0DQEc
2009-04-12-13-20    17uchtodoS9kdeQP1x0XThK15IuRsV
2009-04-12-13-20    J0Uf1SCtRwviGw8sVcghqE5h0nkgtp
2009-04-12-13-20    NQ2XP0J0dvVbCXJ0pb4XvqJ5A4QxxH
2009-04-12-13-20    fFAItiBMsgqro9kRdIwbeX60SR0axr
2009-04-12-13-20    V4og4R9W6G3QjHHwF7gI1cSqiq5D1G
2009-04-12-13-20    hPEPtBwk45msmwWTxPVVo1kVu4v11b
2009-04-12-13-20    v0SkfxegheD90gp31UCr6Fp1nKpx6i
2009-04-12-13-20    1iD9odVg0Ii4QWkwHM0hmwTkWDFkj
2009-04-12-13-20    b31tJiIA25CK8eDHQrHnbcknfSndUk
```

## Escenario 2: Los datos no están particionados en formato Hive

En el siguiente ejemplo, el comando `aws s3 ls` muestra registros de [ELB](#) almacenados en Amazon S3. Tenga en cuenta que el diseño de datos no utiliza pares `key=value` y, por lo tanto, no está en formato Hive. (La opción `--recursive` para el comando `aws s3 ls` especifica que se enumeran todos los objetos o archivos del directorio especificado o con el prefijo indicado).

```
aws s3 ls s3://athena-examples-myregion/elb/plaintext/ --recursive
```

```
2016-11-23 17:54:46 11789573 elb/plaintext/2015/01/01/part-r-00000-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:46 8776899 elb/plaintext/2015/01/01/part-r-00001-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:46 9309800 elb/plaintext/2015/01/01/part-r-00002-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:47 9412570 elb/plaintext/2015/01/01/part-r-00003-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:47 10725938 elb/plaintext/2015/01/01/part-r-00004-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:46 9439710 elb/plaintext/2015/01/01/part-r-00005-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:47 0 elb/plaintext/2015/01/01_$folder$
2016-11-23 17:54:47 9012723 elb/plaintext/2015/01/02/part-r-00006-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:47 7571816 elb/plaintext/2015/01/02/part-r-00007-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:47 9673393 elb/plaintext/2015/01/02/part-r-00008-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 11979218 elb/plaintext/2015/01/02/part-r-00009-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 9546833 elb/plaintext/2015/01/02/part-r-00010-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 10960865 elb/plaintext/2015/01/02/part-r-00011-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 0 elb/plaintext/2015/01/02_$folder$
2016-11-23 17:54:48 11360522 elb/plaintext/2015/01/03/part-r-00012-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 11211291 elb/plaintext/2015/01/03/part-r-00013-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:48 8633768 elb/plaintext/2015/01/03/part-r-00014-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:49 11891626 elb/plaintext/2015/01/03/part-r-00015-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:49 9173813 elb/plaintext/2015/01/03/part-r-00016-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:49 11899582 elb/plaintext/2015/01/03/part-r-00017-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:49 0 elb/plaintext/2015/01/03_$folder$
2016-11-23 17:54:50 8612843 elb/plaintext/2015/01/04/part-r-00018-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:50 10731284 elb/plaintext/2015/01/04/part-r-00019-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
```

```
2016-11-23 17:54:50 9984735 elb/plaintext/2015/01/04/part-r-00020-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:50 9290089 elb/plaintext/2015/01/04/part-r-00021-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:50 7896339 elb/plaintext/2015/01/04/part-r-00022-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 8321364 elb/plaintext/2015/01/04/part-r-00023-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 0 elb/plaintext/2015/01/04_$folder$
2016-11-23 17:54:51 7641062 elb/plaintext/2015/01/05/part-r-00024-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 10253377 elb/plaintext/2015/01/05/part-r-00025-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 8502765 elb/plaintext/2015/01/05/part-r-00026-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 11518464 elb/plaintext/2015/01/05/part-r-00027-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 7945189 elb/plaintext/2015/01/05/part-r-00028-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 7864475 elb/plaintext/2015/01/05/part-r-00029-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 0 elb/plaintext/2015/01/05_$folder$
2016-11-23 17:54:51 11342140 elb/plaintext/2015/01/06/part-r-00030-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:51 8063755 elb/plaintext/2015/01/06/part-r-00031-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 9387508 elb/plaintext/2015/01/06/part-r-00032-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 9732343 elb/plaintext/2015/01/06/part-r-00033-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 11510326 elb/plaintext/2015/01/06/part-r-00034-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 9148117 elb/plaintext/2015/01/06/part-r-00035-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 0 elb/plaintext/2015/01/06_$folder$
2016-11-23 17:54:52 8402024 elb/plaintext/2015/01/07/part-r-00036-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 8282860 elb/plaintext/2015/01/07/part-r-00037-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:52 11575283 elb/plaintext/2015/01/07/part-r-00038-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:53 8149059 elb/plaintext/2015/01/07/part-r-00039-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
```

```

2016-11-23 17:54:53 10037269 elb/plaintext/2015/01/07/part-r-00040-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:53 10019678 elb/plaintext/2015/01/07/part-r-00041-ce65fca5-
d6c6-40e6-b1f9-190cc4f93814.txt
2016-11-23 17:54:53 0 elb/plaintext/2015/01/07_$folder$
2016-11-23 17:54:53 0 elb/plaintext/2015/01_$folder$
2016-11-23 17:54:53 0 elb/plaintext/2015_$folder$

```

## Ejecución de ALTER TABLE ADD PARTITION

Dado que los datos no están en formato Hive, no se puede utilizar el comando `MSCK REPAIR TABLE` para agregar las particiones a la tabla después de crearla. Como alternativa, puede usar el comando [ALTER TABLE ADD PARTITION](#) para agregar cada partición manualmente. Por ejemplo, para cargar los datos en `s3://athena-examples-myregion/elb/plaintext/2015/01/01/`, puede ejecutar la siguiente consulta. Tenga en cuenta que no se requiere una columna de partición independiente para cada carpeta de Amazon S3 y que el valor de la clave de partición puede ser diferente de la clave de Amazon S3.

```

ALTER TABLE elb_logs_raw_native_part ADD PARTITION (dt='2015-01-01') location 's3://
athena-examples-us-west-1/elb/plaintext/2015/01/01/'

```

Si ya existe una partición, recibirá el error `La partición ya existe`. Para evitar este error, puede usar la cláusula `IF NOT EXISTS`. Para obtener más información, consulte [ALTER TABLE ADD PARTITION](#). Para eliminar una partición, puede utilizar [ALTER TABLE DROP PARTITION](#).

## Proyección de particiones

Para evitar tener que administrar particiones, puede usar la proyección de particiones. La proyección de particiones es una opción para tablas divididas en muchas partes cuya estructura se conoce de antemano. En la proyección de particiones, los valores de partición y las ubicaciones se calculan a partir de las propiedades de tabla que configure, en lugar de leerlos desde un repositorio de metadatos. Dado que los cálculos hechos en la memoria son más rápidos que la búsqueda remota, el uso de la proyección de particiones puede reducir significativamente los tiempos de ejecución de las consultas.

Para obtener más información, consulte [Proyección de particiones con Amazon Athena](#).

## Recursos adicionales

- Para obtener información sobre las opciones de partición de datos de Firehose, consulte [Ejemplo de Amazon Data Firehose](#).
- También puede automatizar la adición de particiones con el [controlador JDBC](#).
- Puede utilizar CTAS e INSERT INTO para particionar un conjunto de datos. Para obtener más información, consulte [Uso de CTAS e INSERT INTO en ETL y análisis de datos](#).

## Proyección de particiones con Amazon Athena

Puede utilizar la proyección de particiones en Athena para acelerar el procesamiento de consultas de tablas altamente particionadas y automatizar la administración de particiones.

En la proyección de particiones, Athena calcula los valores de partición y las ubicaciones utilizando las propiedades de tabla que configure directamente en la tabla de AWS Glue. Las propiedades de la tabla permiten a Athena “proyectar” o determinar la información de partición necesaria en lugar de tener que realizar una búsqueda de metadatos en el AWS Glue Data Catalog que consume más tiempo. Dado que las operaciones en memoria suelen ser más rápidas que las operaciones remotas, la proyección de particiones puede reducir el tiempo de ejecución de las consultas en tablas altamente particionadas. Dependiendo de las características específicas de la consulta y los datos subyacentes, la proyección de particiones puede reducir significativamente el tiempo de ejecución de la consulta para las consultas que están restringidas en la recuperación de metadatos de partición.

### Poda y proyección para tablas altamente particionadas

La poda de partición recopila metadatos y los “poda” solo en las particiones que se aplican a su consulta. Por lo general, esto acelera las consultas. Athena utiliza la poda de particiones para todas las tablas con columnas de partición, incluidas las tablas configuradas para la proyección de particiones.

Normalmente, al procesar consultas, Athena realiza una llamada `GetPartitions` al AWS Glue Data Catalog antes de realizar la poda de partición. Si una tabla tiene un gran número de particiones, el uso de `GetPartitions` puede afectar negativamente el rendimiento. Para evitar esto, puede usar la proyección de particiones. La proyección de particiones permite a Athena evitar llamadas a `GetPartitions` porque la configuración de proyección de particiones proporciona a Athena toda la información necesaria para construir las particiones en sí.

## Uso de la proyección de particiones

Para utilizar la proyección de particiones, especifique los intervalos de valores de partición y tipos de proyección para cada columna de partición en las propiedades de tabla en el AWS Glue Data Catalog o en el [metaalmacén externo de Hive](#). Estas propiedades personalizadas en la tabla le permiten a Athena saber qué patrones de partición esperar cuando ejecuta una consulta en la tabla. Durante la ejecución de la consulta, Athena utiliza esta información para proyectar los valores de partición en lugar de recuperarlos desde el metaalmacén externo de Hive o AWS Glue Data Catalog. Esto no solo reduce el tiempo de ejecución de la consulta, sino que también automatiza la administración de particiones, ya que elimina la necesidad de crear manualmente particiones en Athena, el metaalmacén externo de Hive o AWS Glue.

### Important

Habilitar la proyección de particiones en una tabla hace que Athena ignore los metadatos de partición registrados en la tabla en el metaalmacén externo de Hive o AWS Glue Data Catalog.

## Casos de uso

Entre los escenarios en los que la proyección de particiones es útil se incluyen los siguientes:

- Las consultas contra una tabla altamente particionada no se completan tan rápido como le gustaría.
- Las particiones se agregan regularmente a las tablas a medida que se crean nuevas particiones de fecha u hora en los datos. Con la proyección de particiones, puede configurar intervalos de fechas relativos que se pueden utilizar a medida que llegan nuevos datos.
- Tiene datos altamente particionados en Amazon S3. Los datos no son prácticos para modelar en su metaalmacén AWS Glue Data Catalog o Hive, y sus consultas solo leen pequeñas partes de ellos.

## Estructuras de partición proyectables

La proyección de particiones se configura más fácilmente cuando las particiones siguen un patrón predecible como, entre otros, los siguientes:

- Enteros: cualquier secuencia continua de enteros como [1, 2, 3, 4, ..., 1000] o [0500, 0550, 0600, ..., 2500].
- Fechas: cualquier secuencia continua de fechas o fechas y horas como [20200101, 20200102, ..., 20201231] o [1-1-2020 00:00:00, 1-1-2020 01:00:00, ..., 12-31-2020 23:00:00].
- Valores enumerados: un conjunto finito de valores enumerados como códigos de aeropuertos o Regiones de AWS.
- Registros de Servicio de AWS: normalmente, los registros de Servicio de AWS tienen una estructura conocida cuyo esquema de particiones puede especificar en AWS Glue y que, por lo tanto, Athena puede utilizar para la proyección de particiones.

## Personalización de la plantilla de ruta de partición

De forma predeterminada, Athena crea ubicaciones de partición utilizando el formulario `s3://DOC-EXAMPLE-BUCKET/<table-root>/partition-col-1=<partition-col-1-val>/partition-col-2=<partition-col-2-val>/`, pero si sus datos están organizados de manera diferente, Athena ofrece un mecanismo para personalizar esta plantilla de ruta. Para ver los pasos, consulte [Especificación de ubicaciones de almacenamiento de S3 personalizadas](#).

## Consideraciones y limitaciones

Tenga en cuenta las siguientes consideraciones:

- La proyección de particiones elimina la necesidad de especificar particiones manualmente en AWS Glue o en un metaalmacén externo de Hive.
- Cuando habilita la proyección de particiones en una tabla, Athena ignora los metadatos de partición en AWS Glue Data Catalog o el metaalmacén externo de Hive de esa tabla.
- Si una partición proyectada no existe en Amazon S3, Athena seguirá proyectando la partición. Athena no arroja un error, pero no se devuelve ningún dato. Sin embargo, si hay demasiadas particiones vacías, el rendimiento puede ser más lento en comparación con las particiones de AWS Glue tradicionales. Si más de la mitad de las particiones proyectadas están vacías, se recomienda utilizar particiones tradicionales.
- Las consultas de valores que están más allá de los límites de intervalos definidos para la proyección de particiones no devuelven ningún error. En su lugar, la consulta se ejecuta, pero devuelve cero filas. Por ejemplo, si tiene datos relacionados con el tiempo que comienzan en 2020 y se definen como `'projection.timestamp.range' = '2020/01/01, NOW'`, una consulta



como `SELECT * FROM table-name WHERE timestamp = '2019/02/02'` se completará correctamente, pero devolverá cero filas.

- La proyección de particiones solo se puede utilizar cuando se consulta la tabla a través de Athena. Si se lee la misma tabla a través de otro servicio, como Amazon Redshift Spectrum, Athena para Spark o Amazon EMR, se utilizan los metadatos de partición estándar.
- Dado que la proyección de particiones es una característica de solo DML, `SHOW PARTITIONS` no enumera las particiones proyectadas por Athena pero no registradas en el catálogo de AWS Glue o en el metaalmacén externo de Hive.
- Athena no utiliza las propiedades de tabla de vistas como configuración para la proyección de particiones. Para evitar esta limitación, configure y habilite la proyección de particiones en las propiedades de tabla para las tablas a las que hacen referencia las vistas.
- Los [filtros de datos](#) de Lake Formation no se pueden usar con la proyección de particiones en Athena.

## Video

En el siguiente video se muestra cómo utilizar la proyección de particiones para mejorar el rendimiento de las consultas en Athena.

### [Proyección de particiones con Amazon Athena](#)

#### Temas

- [Configuración de la proyección de particiones](#)
- [Tipos admitidos para la proyección de particiones](#)
- [Partición de ID dinámica](#)
- [Ejemplo de Amazon Data Firehose](#)

## Configuración de la proyección de particiones

Configurar la proyección de particiones en las propiedades de una tabla es un proceso de dos pasos:

1. Especifique los intervalos de datos y los patrones relevantes para cada columna de partición, o utilice una plantilla personalizada.
2. Habilite la proyección de particiones para la tabla.

**Note**

Antes de agregar propiedades de proyección de particiones a una tabla existente, la columna de partición para la que va a configurar las propiedades de proyección de particiones ya debe existir en el esquema de la tabla. Si la columna de partición aún no existe, debe agregar una columna de partición a la tabla existente manualmente. AWS Glue no realiza este paso automáticamente.

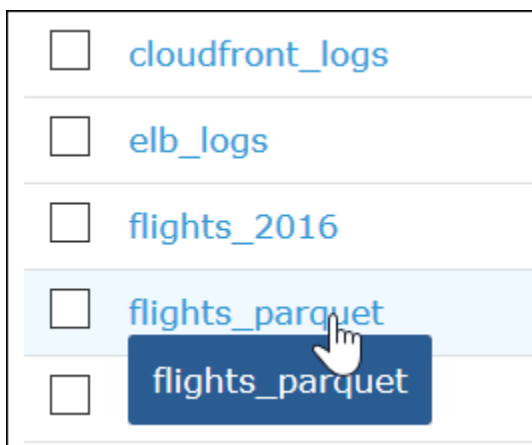
En esta sección se muestra cómo establecer estas propiedades de tabla para AWS Glue. Para configurarlas, puede utilizar la consola de AWS Glue, las consultas [CREATE TABLE](#) de Athena o las operaciones de [AWS Glue API](#). El siguiente procedimiento muestra cómo establecer las propiedades en la consola de AWS Glue.

: cómo configurar y habilitar la proyección de particiones mediante la consola de AWS Glue

1. Inicie sesión en la AWS Management Console y abra la consola de AWS Glue en <https://console.aws.amazon.com/glue/>.
2. Seleccione la pestaña Tablas.

En la pestaña Tablas puede editar tablas existentes o elegir Agregar tablas para crear otras nuevas. Para obtener información sobre cómo agregar tablas manualmente o con un rastreador, consulte [Trabajo con tablas en la consola de AWS Glue](#) en la Guía para desarrolladores de AWS Glue.

3. En la lista de tablas, elija el vínculo de la tabla que desea editar.



4. Seleccione Acciones, Editar la tabla.

5. En la página Editar la tabla, en la sección Propiedades de la tabla, agregue el siguiente par de clave-valor en cada columna particionada:
  - a. En Clave, añada `projection.columnName.type`.
  - b. En Valor, añada uno de los tipos admitidos: `enum`, `integer`, `date`, o `injected`. Para obtener más información, consulte [Tipos admitidos para la proyección de particiones](#).
6. Siguiendo las instrucciones de [Tipos admitidos para la proyección de particiones](#), añada pares clave-valor adicionales de acuerdo con sus requisitos de configuración.

La siguiente configuración de tabla de ejemplo configura la columna `year` para la proyección de particiones, lo que restringe los valores que se pueden devolver a un intervalo comprendido entre 2010 y 2016.

## Edit table details

**Table name**  
flights\_parquet


**Input format**  
org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat

**Output format**

**Table properties**

Key	Value	
last_modified_time	1582588443	×
EXTERNAL	TRUE	×
last_modified_by	hadoop	×
projection.year.type	integer	×
projection.year.range	2010,2016	×
		×


- Añada un par clave-valor para habilitar la proyección de particiones. En Clave, escriba `projection.enabled`, y en su Valor, escriba `true`.

 Note

Puede deshabilitar la proyección de particiones en esta tabla en cualquier momento estableciendo `projection.enabled` como `false`.

8. Cuando termine de actualizar las etiquetas, elija Guardar.
9. En el Editor de consultas de Athena, pruebe la consulta de las columnas que configuró para la tabla.


La siguiente consulta de ejemplo utiliza `SELECT DISTINCT` para devolver los valores únicos de la columna `year`. La base de datos contiene datos de 1987 a 2016, pero la propiedad `projection.year.range` restringe los valores devueltos a los años 2010 a 2016.


 **Query 1**

```
1 SELECT DISTINCT year FROM flights_parquet
2 ORDER BY year ASC
```

SQL Ln 2, Col 18

**Run again** Cancel Save as Clear

 **Completed**  
Time in queue: 0.25 sec Run time: 0.535 sec Data

**Results (7)**  **Copy**

year
2010
2011
2012
2013
2014
2015
2016

**Note**

Si establece `projection.enabled` como `true` pero no puede configurar una o más columnas de partición, recibirá un mensaje de error como el siguiente:

```
HIVE_METASTORE_ERROR: Table database_name.table_name is configured for partition projection, but the following partition columns are missing projection configuration: [column_name] (table database_name.table_name).
```

## Especificación de ubicaciones de almacenamiento de S3 personalizadas

Al editar propiedades de tabla en AWS Glue, también puede especificar una plantilla de ruta de Amazon S3 personalizada para las particiones proyectadas. Una plantilla personalizada permite a Athena asignar correctamente valores de partición a ubicaciones de archivos de Amazon S3 personalizadas que no siguen un patrón `.../column=value/...` típico.

El uso de una plantilla personalizada es opcional. Sin embargo, si utiliza una plantilla personalizada, la plantilla debe contener un marcador de posición para cada columna de partición. Las ubicaciones con plantilla deben terminar con una barra diagonal para que los archivos de datos particionados se alojen en una “carpeta” por partición.

Para especificar una plantilla de ubicación de partición personalizada

1. Siguiendo los pasos para [configurar y habilitar la proyección de particiones mediante la consola de AWS Glue](#), agregue un par clave-valor adicional que especifique una plantilla personalizada de la siguiente manera:
  - a. En Clave, escriba `storage.location.template`.
  - b. En Valor, especifique una ubicación que incluya un marcador de posición para cada columna de partición. Asegúrese de que cada marcador de posición (y la ruta de S3 en sí) termine con una sola barra diagonal.

En los siguientes valores de plantilla de ejemplo se asume una tabla con columnas de partición a, b y c.

```
s3://DOC-EXAMPLE-BUCKET/table_root/a=${a}/${b}/some_static_subdirectory/${c}/
```

```
s3://DOC-EXAMPLE-BUCKET/table_root/c=${c}/${b}/some_static_subdirectory/${a}/
${b}/${c}/${c}/
```

En la misma tabla, el siguiente valor de plantilla de ejemplo no es válido porque no contiene marcador de posición para la columna c.

```
s3://DOC-EXAMPLE-BUCKET/table_root/a=${a}/${b}/some_static_subdirectory/
```

## 2. Seleccione Apply.

### Tipos admitidos para la proyección de particiones

Una tabla puede tener cualquier combinación de tipos de columna de partición `enum`, `integer`, `date`, o `injected`.

#### Tipo `enum`

Utilice el tipo `enum` para las columnas de partición cuyos valores sean miembros de un conjunto enumerado (por ejemplo, códigos de aeropuerto o Regiones de AWS).

Defina las propiedades de partición en la tabla de la siguiente manera:

Nombre de la propiedad	Valores de ejemplo	Descripción
<code>projection. <i>columnName</i> e.type</code>	<code>enum</code>	Obligatorio. El tipo de proyección que se va a utilizar en la columna <code><i>columnName</i></code> . El valor debe ser <code>enum</code> (sin distinción mayúsculas y minúsculas) para indicar el uso del tipo <code>enum</code> . Se permite un espacio en blanco inicial y final.
<code>projection. <i>columnName</i> e.values</code>	<code>A,B,C,D,E,F,G,Unkn own</code>	Obligatorio. Lista separada por comas de los valores de



Nombre de la propiedad	Valores de ejemplo	Descripción
		partición enumerados para la columna <i>columnName</i> . Cualquier espacio en blanco se considera parte de un valor enum.

### Note

Como práctica recomendada, sugerimos limitar el uso de proyecciones de partición basadas en enum a unas pocas docenas o menos. Si bien no hay un límite específico para las proyecciones enum, el tamaño total de los metadatos de la tabla no puede superar el límite de AWS Glue de aproximadamente 1 MB cuando se comprime a gzip. Tenga en cuenta que este límite se comparte entre partes clave de la tabla, como nombres de columna, ubicación, formato de almacenamiento y otros. Si utiliza más de unas cuantas docenas de identificadores únicos en la proyección enum, considere un enfoque alternativo, como la asignación de buckets en un número menor de valores únicos en un campo sustituto. Al intercambiar cardinalidad, puede controlar el número de valores únicos en enum.

## Tipo entero

Utilice el tipo entero para las columnas de partición cuyos valores posibles sean interpretables como enteros dentro de un intervalo definido. Las columnas enteras proyectadas se limitan actualmente al intervalo de un Java con signo largo ( $-2^{63}$  a  $2^{63}-1$  inclusive).

Nombre de la propiedad	Valores de ejemplo	Descripción
projection. <i>columnName</i> .type	integer	Obligatorio. El tipo de proyección que se va a utilizar en la columna <i>columnName</i> . El valor debe ser integer (sin distinción entre mayúsculas y minúsculas) para indicar el uso del tipo entero. Se permite

Nombre de la propiedad	Valores de ejemplo	Descripción
		un espacio en blanco inicial y final.
<code>projection. <i>columnName</i> .range</code>	0,10 -1,8675309 0001,9999	Obligatorio. Lista separada por comas de dos elementos que proporciona los valores de intervalo mínimo y máximo que deben devolver las consultas de la columna <i>columnName</i> . Tenga en cuenta que los valores deben estar separados por una coma, no por un guion. Estos valores son inclusivos, pueden ser negativos y pueden tener ceros a la izquierda. Se permite un espacio en blanco inicial y final.
<code>projection. <i>columnName</i> .interval</code>	1 5	Opcional. Un entero positivo que especifica el intervalo entre los valores de partición sucesivos en la columna <i>columnName</i> . Por ejemplo, un valor range de "1,3" con un valor interval de "1" produce los valores 1, 2 y 3. El mismo valor range con un valor interval de "2" produce los valores 1 y 3, omitiendo 2. Se permite un espacio en blanco inicial y final. El valor predeterminado es 1.

Nombre de la propiedad	Valores de ejemplo	Descripción
<code>projection. <i>columnName</i>.digits</code>	1 5	Opcional. Un entero positivo que especifica el número de dígitos que se incluirán en la representación final del valor de partición de la columna <i>columnName</i> . Por ejemplo, un valor de <code>range</code> de "1,3" que tiene un valor de <code>digits</code> de "1" produce los valores 1, 2 y 3. El mismo valor de <code>range</code> con un valor de <code>digits</code> de "2" produce los valores 01, 02 y 03. Se permite un espacio en blanco inicial y final. Por defecto, no hay número estático de dígitos ni ceros a la izquierda.

## Tipo de fecha

Utilice el tipo de fecha para las columnas de partición cuyos valores se pueden interpretar como fechas (con horas opcionales) dentro de un rango definido.

### Important

Las columnas de fecha proyectada se generan en hora universal coordinada (UTC) en el momento de ejecución de la consulta.

Nombre de la propiedad	Valores de ejemplo	Descripción
<code>projection. <i>columnName</i>.type</code>	date	Obligatorio. El tipo de proyección que se va a utilizar en la columna <i>columnName</i> . El valor

Nombre de la propiedad	Valores de ejemplo	Descripción
		debe ser date (sin distinción entre mayúsculas y minúsculas) para indicar el uso del tipo de fecha. Se permite un espacio en blanco inicial y final.
<code>projection.<i>columnName</i>.range</code>	<p>201701,201812</p> <p>01-01-2010,12-31-2018</p> <p>NOW-3YEARS,NOW</p> <p>201801,NOW+1MONTH</p>	<p>Obligatorio. Lista separada por comas de dos elementos que proporciona los valores <code>range</code> mínimo y máximo de la columna <i>columnName</i>. Estos valores son inclusivos y pueden utilizar cualquier formato compatible con los tipos de fechas <code>java.time.*</code> de Java. Tanto los valores mínimo como máximo deben utilizar el mismo formato. El formato especificado en la propiedad <code>.format</code> debe ser el formato utilizado para estos valores.</p> <p>Esta columna también puede contener cadenas de fecha relativas, con el formato de este patrón de expresión regular:</p> <pre>\s*NOW\s*(([\+ -])\s*([0-9]+)\s*(YEARS? MONTHS? WEEKS? DAYS? HOURS? MINUTES? SECONDS?)\s*)?</pre> <p>Se permiten espacios en blanco, pero los literales de fecha se consideran parte de las cadenas de fecha.</p>

Nombre de la propiedad	Valores de ejemplo	Descripción
<code>projection.<i>columnName</i>.format</code>	<pre>yyyyMM dd-MM-yy y dd-MM-yyyy y-HH-mm-ss</pre>	<p>Obligatorio. Una cadena de formato de fecha basada en el formato de fecha Java <a href="#">DateTimeFormatter</a>. Puede ser cualquier tipo de <code>java.time.*</code> compatible.</p>
<code>projection.<i>columnName</i>.interval</code>	<pre>1 5</pre>	<p>Un entero positivo que especifica el intervalo entre los valores de partición sucesivos de la columna <i>columnName</i>. Por ejemplo, un valor de <code>range</code> de <code>2017-01, 2018-12</code> con un valor de <code>interval</code> de 1 y un valor de <code>interval.unit</code> de MONTHS produce los valores 2017-01, 2017-02, 2017-03, etc. El mismo valor de <code>range</code> con un valor de <code>interval</code> de 2 y un valor de <code>interval.unit</code> de MONTHS produce los valores 2017-01, 2017-03, 2017-05, etc. Se permite un espacio en blanco inicial y final.</p> <p>Cuando las fechas proporcionadas tienen una precisión de un solo día o de un mes, la <code>interval</code> es opcional y el valor predeterminado es 1 día o 1 mes, respectivamente. De lo contrario, se requiere el <code>interval</code>.</p>

Nombre de la propiedad	Valores de ejemplo	Descripción
<code>projection.<i>columnName</i>.interval.unit</code>	YEARS MONTHS WEEKS DAYS HOURS MINUTES SECONDS MILLIS	Palabra de unidad de tiempo que representa la forma serializada de una <a href="#">ChronoUnit</a> . Los valores posibles son YEARS, MONTHS, WEEKS, DAYS, HOURS, MINUTES, SECONDS o MILLIS. Los valores no distinguen entre mayúsculas y minúsculas.  Cuando las fechas proporcionadas tienen una precisión de un solo día o de un mes, la <code>interval.unit</code> es opcional y el valor predeterminado es 1 día o 1 mes, respectivamente. De lo contrario, se requiere la <code>interval.unit</code> .

### Tipo inyectado

Utilice el tipo inyectado para columnas de partición con valores posibles que no se pueden generar procesalmente dentro de algún intervalo lógico pero que se proporcionan en una cláusula WHERE de la consulta como un solo valor.

Es importante tener en cuenta los siguientes puntos:

- Las consultas sobre columnas inyectadas fallan si no se proporciona una expresión de filtro para cada columna inyectada.
- Las consultas con múltiples valores para una expresión de filtro en una columna inyectada solo funcionan si los valores están separados.
- Solo se admiten las columnas de tipo `string`.

Nombre de la propiedad	Valor	Descripción
<code>projection.<i>columnName</i>.type</code>	<code>injected</code>	Obligatorio. El tipo de proyección que se va a utilizar en la columna <code>columnName</code> . Solo se admite el tipo <code>string</code> . El valor especificado debe ser <code>injected</code> (sin

Nombre de la propiedad	Valor	Descripción
		distinción entre mayúsculas y minúsculas). Se permite un espacio en blanco inicial y final.

Para obtener más información, consulte [Utilización del tipo de proyección `injected`](#).

## Partición de ID dinámica

Si los datos están divididos por una propiedad con cardinalidad alta o cuando los valores no se pueden conocer de antemano, puede utilizar el tipo de proyección `injected`. Algunos ejemplos de estas propiedades son los nombres de usuario y los ID de dispositivos o productos. Cuando se utiliza el tipo de proyección `injected` para configurar una clave de partición, Athena utiliza los valores de la propia consulta para calcular el conjunto de particiones que se leerán.

Para que Athena pueda ejecutar una consulta en una tabla que tenga una clave de partición configurada con el tipo de proyección `injected`, debe cumplirse lo siguiente:

- La consulta debe incluir al menos un valor para la clave de partición.
- Los valores deben ser literales o expresiones que se puedan evaluar sin leer ningún dato.

Si no se cumple alguno de estos criterios, la consulta fallará y mostrará el siguiente error:

CONSTRAINT\_VIOLATION: En el caso de la columna de partición proyectada `injected` *column\_name*, la cláusula WHERE debe contener únicamente condiciones de igualdad estáticas, y al menos una de estas condiciones debe estar presente.

## Utilización del tipo de proyección `injected`

Imagine que tiene un conjunto de datos que consta de eventos de dispositivos de IoT, particionados en los ID de los dispositivos. Este conjunto de datos incluye las siguientes características:

- Los ID de los dispositivos se generan de forma aleatoria.
- Los dispositivos nuevos se aprovisionan con frecuencia.
- Actualmente, hay cientos de miles de dispositivos y en el futuro habrá millones.

Este conjunto de datos es difícil de administrar con los metaalmacenes tradicionales. Es difícil mantener las particiones sincronizadas entre el almacenamiento de datos y el metaalmacén, y

el filtrado de las particiones puede resultar ser lento durante la planificación de las consultas. Sin embargo, si configura una tabla para usar la proyección de particiones y usa el tipo de proyección `injected`, tiene dos ventajas: no tiene que administrar las particiones en el metaalmacén y sus consultas no tienen que buscar los metadatos de las particiones.

En el siguiente ejemplo de `CREATE TABLE`, se crea una tabla para el conjunto de datos de eventos del dispositivo que se acaba de describir. La tabla utiliza el tipo de proyección inyectada.

```
CREATE EXTERNAL TABLE device_events (  
    event_time TIMESTAMP,  
    data STRING,  
    battery_level INT  
)  
PARTITIONED BY (  
    device_id STRING  
)  
LOCATION "s3://DOC-EXAMPLE-BUCKET/prefix/"  
TBLPROPERTIES (  
    "projection.enabled" = "true",  
    "projection.device_id.type" = "injected",  
    "storage.location.template" = "s3://DOC-EXAMPLE-BUCKET/prefix/${device_id}"  
)
```

La siguiente consulta de ejemplo busca el número de eventos recibidos de tres dispositivos específicos en el transcurso de 12 horas.

```
SELECT device_id, COUNT(*) AS events  
FROM device_events  
WHERE device_id IN (  
    '4a770164-0392-4a41-8565-40ed8cec737e',  
    'f71d12cf-f01f-4877-875d-128c23cbde17',  
    '763421d8-b005-47c3-ba32-cc747ab32f9a'  
)  
AND event_time BETWEEN TIMESTAMP '2023-11-01 20:00' AND TIMESTAMP '2023-11-02 08:00'  
GROUP BY device_id
```

Al ejecutar esta consulta, Athena ve los tres valores de la clave de partición `device_id` y los usa para calcular las ubicaciones de las particiones. Athena utiliza el valor de la propiedad `storage.location.template` para generar las siguientes ubicaciones:

- `s3://DOC-EXAMPLE-BUCKET/prefix/4a770164-0392-4a41-8565-40ed8cec737e`



- `s3://DOC-EXAMPLE-BUCKET/prefix/f71d12cf-f01f-4877-875d-128c23cbde17`
- `s3://DOC-EXAMPLE-BUCKET/prefix/763421d8-b005-47c3-ba32-cc747ab32f9a`

Si omite la propiedad `storage.location.template` en la configuración de proyección de la partición, Athena utiliza la partición tipo Hive para proyectar las ubicaciones de las particiones en función del valor de `LOCATION` (por ejemplo, `s3://DOC-EXAMPLE-BUCKET/prefix/device_id=4a770164-0392-4a41-8565-40ed8cec737e`).

## Ejemplo de Amazon Data Firehose

Cuando se usa Firehose para entregar datos a Amazon S3, la configuración predeterminada escribe objetos con claves que se parecen al siguiente ejemplo:

```
s3://DOC-EXAMPLE-BUCKET/prefix/yyyy/MM/dd/HH/file.extension
```

Para crear una tabla de Athena que encuentre las particiones de forma automática en el momento de la consulta, en lugar de tener que agregarlas al catálogo de datos de AWS Glue Data Catalog a medida que llegan nuevos datos, puede usar la proyección de particiones.

En el siguiente ejemplo de `CREATE TABLE` se usa la configuración predeterminada de Firehose.

```
CREATE EXTERNAL TABLE my_ingested_data (
  ...
)
...
PARTITIONED BY (
  datehour STRING
)
LOCATION "s3://DOC-EXAMPLE-BUCKET/prefix/"
TBLPROPERTIES (
  "projection.enabled" = "true",
  "projection.datehour.type" = "date",
  "projection.datehour.format" = "yyyy/MM/dd/HH",
  "projection.datehour.range" = "2021/01/01/00,NOW",
  "projection.datehour.interval" = "1",
  "projection.datehour.interval.unit" = "HOURS",
  "storage.location.template" = "s3://DOC-EXAMPLE-BUCKET/prefix/${datehour}/"
)
```

La cláusula `TBLPROPERTIES` de la instrucción `CREATE TABLE` indica a Athena lo siguiente:

- Usar la proyección de la partición al consultar la tabla
- La clave de partición `datehour` es de tipo `date` (que incluye una hora opcional)
- Cómo se formatean las fechas
- El rango de fechas y horas. Tenga en cuenta que los valores deben estar separados por una coma, no por un guion.
- Dónde encontrar los datos en Amazon S3.

Al consultar la tabla, Athena calcula los valores de `datehour` y usa la plantilla de ubicación de almacenamiento para generar una lista de ubicaciones de partición.

### Uso del tipo `date`

Cuando se usa el tipo `date` para una clave de partición proyectada, se debe especificar un rango. Como no tiene datos para fechas anteriores a la creación del flujo de entrega de Firehose, puede usar la fecha de creación como inicio. Y como no tiene datos para fechas en el futuro, puede usar el token especial `NOW` como fin.

En el ejemplo de `CREATE TABLE`, la fecha de inicio se especifica como 1 de enero de 2021 a medianoche UTC.

#### Note

Configure un rango que se ajuste lo más posible a sus datos para que Athena busque solo las particiones existentes.

Cuando se ejecuta una consulta en la tabla de ejemplo, Athena usa las condiciones de la clave de partición `datehour` en combinación con el rango para generar valores. Analice la siguiente consulta:

```
SELECT *
FROM my_ingested_data
WHERE datehour >= '2020/12/15/00'
AND datehour < '2021/02/03/15'
```

La primera condición de la consulta `SELECT` usa una fecha que es anterior al inicio del rango de fechas especificado por la instrucción `CREATE TABLE`. Como la configuración de la proyección de la

partición específica que no hay particiones para las fechas anteriores al 1 de enero de 2021, Athena busca datos solo en las siguientes ubicaciones, e ignora las fechas anteriores a la consulta.

```
s3://DOC-EXAMPLE-BUCKET/prefix/2021/01/01/00/  
s3://DOC-EXAMPLE-BUCKET/prefix/2021/01/01/01/  
s3://DOC-EXAMPLE-BUCKET/prefix/2021/01/01/02/  
...  
s3://DOC-EXAMPLE-BUCKET/prefix/2021/02/03/12/  
s3://DOC-EXAMPLE-BUCKET/prefix/2021/02/03/13/  
s3://DOC-EXAMPLE-BUCKET/prefix/2021/02/03/14/
```

Del mismo modo, si la consulta se ejecuta en una fecha y hora anteriores al 3 de febrero de 2021 a las 15:00 h, la última partición reflejaría la fecha y hora actuales, no la fecha y hora de la condición de consulta.

Si quiere consultar los datos más recientes, puede aprovechar el hecho de que Athena no genera fechas futuras y especificar solo una `datehour` de inicio, como en el siguiente ejemplo.

```
SELECT *  
FROM my_ingested_data  
WHERE datehour >= '2021/11/09/00'
```

### Elección de claves de partición

Puede especificar cómo la proyección de particiones asigna las ubicaciones de las particiones a las claves de partición. En el ejemplo de `CREATE TABLE` de la sección anterior, la fecha y la hora se han combinado en una clave de partición denominada `datehour`, pero se pueden usar otros esquemas. Por ejemplo, también podría configurar una tabla con claves de partición separadas para el año, el mes, el día y la hora.

No obstante, si se dividen las fechas en año, mes y día no se puede utilizar el tipo de proyección de particiones `date`. Una alternativa es separar la fecha de la hora para seguir aprovechando el tipo de proyección de particiones `date`, pero realizar consultas que especifiquen intervalos de horas más fáciles de leer.

Con esto en mente, el siguiente ejemplo `CREATE TABLE` separa la fecha de la hora. Debido a que `date` es una palabra reservada en SQL, en el ejemplo se utiliza `day` como nombre para la clave de partición que representa la fecha.

```
CREATE EXTERNAL TABLE my_ingested_data2 (  
  ...  
)  
  ...  
PARTITIONED BY (  
  day STRING,  
  hour INT  
)  
LOCATION "s3://DOC-EXAMPLE-BUCKET/prefix/"  
TBLPROPERTIES (  
  "projection.enabled" = "true",  
  "projection.day.type" = "date",  
  "projection.day.format" = "yyyy/MM/dd",  
  "projection.day.range" = "2021/01/01,NOW",  
  "projection.day.interval" = "1",  
  "projection.day.interval.unit" = "DAYS",  
  "projection.hour.type" = "integer",  
  "projection.hour.range" = "0,23",  
  "projection.hour.digits" = "2",  
  "storage.location.template" = "s3://DOC-EXAMPLE-BUCKET/prefix/${day}/${hour}/"  
)
```

En la instrucción `CREATE TABLE` del ejemplo, la hora es una clave de partición independiente, configurada como un número entero. La configuración de la clave de partición de la hora especifica el rango de 0 a 23, y que la hora debe formatearse con dos dígitos cuando Athena genera las ubicaciones de la partición.

Una consulta para la tabla `my_ingested_data2` podría verse así:

```
SELECT *  
FROM my_ingested_data2  
WHERE day = '2021/11/09'  
AND hour > 3
```

## Tipos de claves de partición y tipos de proyección de particiones

Tenga en cuenta que la clave `datehour` en el primer ejemplo `CREATE TABLE` está configurada como `date` en la configuración de la proyección de particiones, pero el tipo de la clave de partición es `string`. Lo mismo ocurre con `day` en el segundo ejemplo. Los tipos de configuración de proyección de particiones solo le dicen a Athena cómo formatear los valores cuando genera las ubicaciones de las particiones. Los tipos que se especifican no cambian el tipo de la clave de partición: en las consultas, `datehour` y `day` son de tipo `string`.

Cuando una consulta incluye una condición como `day = '2021/11/09'`, Athena analiza la cadena a la derecha de la expresión con el formato de fecha especificado en la configuración de la proyección de particiones. Después de que Athena verifique que la fecha está dentro del rango configurado, usa de nuevo el formato de fecha para insertar la fecha como una cadena en la plantilla del almacén.

De forma similar, para una condición de consulta como `day > '2021/11/09'`, Athena analiza el lado derecho y genera una lista de todas las fechas coincidentes dentro del rango configurado. A continuación, usa el formato de fecha para insertar cada fecha en la plantilla de ubicación de almacenamiento para crear la lista de ubicaciones de partición.

Escribir la misma condición como `día day > '2021-11-09'` o `día day > DATE '2021-11-09'` no funciona. En el primer caso, el formato de la fecha no coincide (observe los guiones en lugar de las barras inclinadas), y en el segundo caso, los tipos de datos no coinciden.

### Uso de prefijos personalizados y particiones dinámicas

Firehose puede configurarse con [prefijos personalizados](#) y [partición dinámica](#). Mediante estas características, puede configurar las claves de Amazon S3 y establecer esquemas de partición más compatibles con su caso de uso. También puede usar la proyección de particiones con estos esquemas de partición y configurarlos en consecuencia.

Por ejemplo, puede usar la característica de prefijo personalizado para obtener claves de Amazon S3 que tengan fechas con formato ISO en lugar del esquema predeterminado `yyyy/MM/dd/HH`.

También puede combinar los prefijos personalizados con el particionamiento dinámico para extraer una propiedad como `customer_id` de los mensajes de Firehose, como en el siguiente ejemplo.

```
prefix/!{timestamp:yyyy}-!{timestamp:MM}-!{timestamp:dd}/!  
{partitionKeyFromQuery:customer_id}/
```

Con ese prefijo de Amazon S3, el flujo de entrega de Firehose escribiría objetos en claves como `s3://DOC-EXAMPLE-BUCKET/prefix/2021-11-01/customer-1234/file.extension`. Para una propiedad como `customer_id`, cuyos valores pueden no conocerse de antemano, se puede usar el tipo de proyección de particiones [injected](#) y usar una instrucción `CREATE TABLE` como la siguiente:

```
CREATE EXTERNAL TABLE my_ingested_data3 (  
  ...
```

```

)
...
PARTITIONED BY (
  day STRING,
  customer_id STRING
)
LOCATION "s3://DOC-EXAMPLE-BUCKET/prefix/"
TBLPROPERTIES (
  "projection.enabled" = "true",
  "projection.day.type" = "date",
  "projection.day.format" = "yyyy-MM-dd",
  "projection.day.range" = "2021-01-01,NOW",
  "projection.day.interval" = "1",
  "projection.day.interval.unit" = "DAYS",
  "projection.customer_id.type" = "injected",
  "storage.location.template" = "s3://DOC-EXAMPLE-BUCKET/prefix/${day}/${customer_id}/"
)

```

Cuando se consulta una tabla que tiene una clave de partición de tipo `injected`, la consulta debe incluir un valor para esa clave de partición. Una consulta para la tabla `my_ingested_data3` podría verse así:

```

SELECT *
FROM my_ingested_data3
WHERE day BETWEEN '2021-11-01' AND '2021-11-30'
AND customer_id = 'customer-1234'

```

## Fechas con formato ISO

Ya que los valores de la clave de partición `day` tienen formato ISO, también puede usar el tipo `DATE` para la clave de partición del día en lugar de `STRING`, como en el siguiente ejemplo:

```

PARTITIONED BY (day DATE, customer_id STRING)

```

Cuando se consulta, esta estrategia permite usar las funciones de fecha en la clave de partición sin analizar ni iniciar, como en el siguiente ejemplo:

```

SELECT *
FROM my_ingested_data3
WHERE day > CURRENT_DATE - INTERVAL '7' DAY
AND customer_id = 'customer-1234'

```

**Note**

Al especificar una clave de partición del tipo DATE, se supone que ha utilizado la característica de [prefijo personalizado](#) para crear claves de Amazon S3 con fechas con formato ISO. Si utiliza el formato predeterminado de Firehose de yyyy/MM/dd/HH, debe especificar la clave de partición como tipo `string` aunque la propiedad de la tabla correspondiente sea de tipo `date`, como en el siguiente ejemplo:

```
PARTITIONED BY (  
  `mydate` string)  
TBLPROPERTIES (  
  'projection.enabled'='true',  
  ...  
  'projection.mydate.type'='date',  
  'storage.location.template'='s3://DOC-EXAMPLE-BUCKET/prefix/${mydate}')
```

## Creación de una tabla a partir de los resultados de una consulta (CTAS)

Una consulta `CREATE TABLE AS SELECT` (CTAS) crea una nueva tabla en Athena a partir de los resultados de una instrucción `SELECT` de otra consulta. Athena almacena los archivos de datos creados por la instrucción CTAS en una ubicación especificada en Amazon S3. Para ver la sintaxis, consulte [CREATE TABLE AS](#).

`CREATE TABLE AS` combina una instrucción DDL `CREATE TABLE` con una instrucción DML `SELECT` y, por lo tanto, técnicamente contiene tanto DDL como DML. Sin embargo, tenga en cuenta que, a efectos de Service Quotas, las consultas CTAS en Athena se tratan como DML. Para obtener información acerca de Service Quotas de Athena, consulte [Service Quotas](#).

Utilice consultas CTAS para:

- Crear tablas a partir de los resultados de una consulta en un solo paso, sin tener que consultar una y otra vez conjuntos de datos sin formato. De ese modo, resulta más sencillo trabajar con conjuntos de datos sin formato.
- Transformar los resultados de las consultas y migrar las tablas a otros formatos de tabla, como Apache Iceberg. De esta manera, se mejora el rendimiento de las consultas y se reducen los

costos de las consultas en Athena. Para obtener más información, consulte [Creación de tablas de Iceberg](#).

- Transformar los resultados de las consultas en formatos de almacenamiento, como Parquet y ORC. De esta manera, se mejora el rendimiento de las consultas y se reducen los costos de las consultas en Athena. Para obtener más información, consulte [Formatos de almacenamiento en columnas](#).
- Crear copias de tablas existentes que contengan solo los datos que necesita.

## Temas

- [Consideraciones y limitaciones de las consultas CTAS](#)
- [Ejecución de consultas CTAS en la consola](#)
- [Creación de particiones y asignación de buckets en Athena](#)
- [Ejemplos de consultas CTAS](#)
- [Uso de CTAS e INSERT INTO en ETL y análisis de datos](#)
- [Uso de CTAS e INSERT INTO para evitar el límite de 100 particiones](#)

## Consideraciones y limitaciones de las consultas CTAS

En las siguientes secciones se detallan las consideraciones y limitaciones que se deben tener en cuenta cuando se utilicen consultas `CREATE TABLE AS SELECT` (CTAS) en Athena.

### Sintaxis de las consultas CTAS

La sintaxis de las consultas CTAS es diferente de la sintaxis de `CREATE [EXTERNAL] TABLE` utilizada para la creación de tablas. Consulte [CREATE TABLE AS](#).

### Consultas CTAS y vistas

Las consultas CTAS escriben nuevos datos en una ubicación especificada en Amazon S3, mientras que las vistas no escriben ningún dato.

### Ubicación de los resultados de las consultas CTAS

Si el grupo de trabajo [anula la configuración del lado del cliente](#) para la ubicación de resultados de consulta, Athena crea la tabla en la ubicación `s3://DOC-EXAMPLE-BUCKET/tables/<query-id>/`. Para ver la ubicación de los resultados de la consulta especificada para el grupo de trabajo, [consulte los detalles del grupo de trabajo](#).



Si el grupo de trabajo no reemplaza la ubicación de los resultados de la consulta, puede utilizar la sintaxis `WITH (external_location = 's3://DOC-EXAMPLE-BUCKET/')` de la consulta CTAS para especificar dónde se almacenan los resultados de la consulta CTAS.

#### Note

La propiedad `external_location` debe especificar una ubicación vacía. Una consulta CTAS comprueba que la ubicación de ruta (prefijo) del bucket esté vacía y nunca sobrescribe los datos si la ubicación ya contiene datos. Para volver a utilizar la misma ubicación, elimine los datos de la ubicación del prefijo de clave en el bucket.

Si omite la sintaxis `external_location` y no utiliza la configuración del grupo de trabajo, Athena utiliza la [configuración del lado del cliente](#) para la ubicación de resultados de la consulta y crea la tabla en la ubicación `s3://DOC-EXAMPLE-BUCKET/<Unsaved-or-query-name>/<year>/<month>/<date>/tables/<query-id>/`.

## Localización de archivos huérfanos

Si una instrucción `INSERT INTO` o CTAS produce un error, es posible que queden datos huérfanos en la ubicación de datos. Debido a que en algunos casos Athena no elimina datos o datos parciales del bucket, es posible que se puedan leer esos datos parciales en consultas posteriores. Para localizar archivos huérfanos para su inspección o eliminación, puede utilizar el archivo de manifiesto de datos que Athena proporciona para realizar un seguimiento de la lista de archivos que se van a escribir. Para obtener más información, consulte [Identificación de archivos de salida de consultas](#) y [DataManifestLocation](#).

## Cláusulas `ORDER BY` ignoradas

En una consulta CTAS, Athena ignora las cláusulas `ORDER BY` en la parte `SELECT` de la consulta.

Según la especificación SQL (ISO 9075, parte 2), el orden de las filas de una tabla especificada por una expresión de consulta solo está garantizado para la expresión de consulta que contenga inmediatamente la cláusula `ORDER BY`. En cualquier caso, las tablas de SQL están intrínsecamente desordenadas y, si se implementa `ORDER BY` en las cláusulas de subconsulta, la consulta tiene un rendimiento deficiente y no se obtiene un resultado ordenado. Por lo tanto, en las consultas CTAS de Athena, no hay garantía de que el orden especificado en la cláusula `ORDER BY` se conserve cuando se escriban los datos.

## Formatos para almacenar los resultados de las consultas

Los resultados de las consultas CTAS se almacenan en Parquet de forma predeterminada, si no especifica un formato de almacenamiento de datos. Puede almacenar los resultados de las consultas CTAS en PARQUET, ORC, AVRO, JSON y TEXTFILE. Los delimitadores de múltiples caracteres no son compatibles con el formato TEXTFILE CTAS. Las consultas CTAS no requieren que se especifique un SerDe para interpretar las transformaciones de datos. Consulte [Example: Writing query results to a different format](#).

## Formatos de compresión

La compresión GZIP se utiliza para los resultados de las consultas CTAS en los formatos JSON y TEXTFILE. Para Parquet, puede utilizar GZIP o SNAPPY; el valor predeterminado es GZIP. Para Parquet, puede utilizar LZ4, SNAPPY, ZLIB o ZSTD; el valor predeterminado es ZLIB. Para obtener ejemplos de CTAS que especifican compresión, consulte [Example: Specifying data storage and compression formats](#). Para obtener más información sobre la compresión en Athena, consulte [Compatibilidad con la compresión de Athena](#).

## Límites de partición y bucket

Puede particionar y guardar en buckets los datos de los resultados de una consulta CTAS. Para obtener más información, consulte [Creación de particiones y asignación de buckets en Athena](#). Al crear una tabla particionada mediante CTAS, Athena tiene un límite de escritura de 100 particiones.

Incluir predicados de particiones y buckets al final de la cláusula WITH que especifica las propiedades de la tabla de destino. Para obtener más información, consulte [Example: Creating bucketed and partitioned tables](#) y [Creación de particiones y asignación de buckets en Athena](#).

Para obtener información sobre cómo evitar la limitación de 100 particiones, consulte [Uso de CTAS e INSERT INTO para evitar el límite de 100 particiones](#).

## Cifrado

Puede cifrar los resultados de las consultas CTAS en Amazon S3, del mismo modo que cifra los resultados de otras consultas en Athena. Para obtener más información, consulte [Cifrado de los resultados de las consultas de Athena en Amazon S3](#).

## Propietario del bucket esperado

Para las instrucciones CTAS, la configuración del propietario del bucket esperado no se aplica a la ubicación de la tabla de destino en Amazon S3. La configuración del propietario esperado del

bucket se aplica solo a la ubicación de salida de Amazon S3 especificada para los resultados de las consultas de Athena. Para obtener más información, consulte [Especificación de una ubicación de resultados de consulta mediante la consola de Athena](#).

## Tipos de datos

Los tipos de datos de columna de una consulta CTAS son los mismos que los que se especifican para la consulta original.

## Ejecución de consultas CTAS en la consola

En la consola de Athena, se puede crear una consulta CTAS a partir de otra consulta.

Para crear una consulta CTAS a partir de otra consulta

1. Ejecute la consulta en el editor de consultas de la consola de Athena.
2. En la parte inferior del editor de consultas, elija la opción Create (Crear) y, a continuación, elija Table from query (Tabla a partir de consulta).
3. En el formulario Create table as select (Crear tabla basada en la selección), complete los campos como se indica a continuación:
  - a. En Table name (Nombre de la tabla), ingrese el nombre de la nueva tabla. Utilice únicamente minúsculas y caracteres de subrayado, como `my_select_query_parquet`.
  - b. En Database configuration (Configuración de la base de datos), utilice las opciones para elegir una base de datos existente o crear una.
  - c. (Opcional) En Result configuration (Configuración de resultados), para Location of CTAS query results (Ubicación de los resultados de la consulta de CTAS), si la configuración de ubicación de los resultados de la consulta de grupo de trabajo no anula esta opción, haga una de las siguientes acciones:
    - Ingrese la ruta a una ubicación de S3 existente en el cuadro de búsqueda o seleccione Browse S3 (Examinar S3) para elegir una ubicación de una lista.
    - Elija View (Ver) para abrir la página Buckets (Buckets) de la consola de Amazon S3, donde podrá ver más información sobre sus buckets existentes y elegir o crear un bucket con su propia configuración.

Debería especificar una ubicación vacía en Amazon S3 en la que se generarán los datos. Si ya existen datos en la ubicación especificada, se produce un error en la consulta.

Si el grupo de trabajo anula la configuración del cliente para la ubicación de resultados de consulta, Athena crea la tabla en la ubicación `s3://DOC-EXAMPLE-BUCKET/tables/query_id/`.

- d. En Data format (Formato de datos), especifique el formato en el que se encuentran los datos.
  - Table type (Tipo de tabla): el tipo de tabla predeterminado en Athena es Apache Hive.
  - File format (Formato de archivo): elija entre opciones como CSV, TSV, JSON, Parquet u ORC. Para obtener más información sobre los formatos Parquet y ORC, consulte [Formatos de almacenamiento en columnas](#).
  - Write compression (Compresión de escritura): (opcional) elija un formato de compresión. Athena admite una variedad de formatos de compresión para leer y escribir datos, como la lectura de una tabla que utiliza varios formatos de compresión. Por ejemplo, Athena puede leer correctamente los datos de una tabla que utiliza el formato de archivo Parquet cuando algunos archivos Parquet se comprimen con Snappy y otros archivos Parquet se comprimen con GZIP. El mismo principio se aplica a los formatos de almacenamiento ORC, archivo de texto y JSON. Para obtener más información, consulte [Compatibilidad con la compresión de Athena](#).
  - Partitions (Particiones): (opcional) seleccione las columnas que quiera particionar. La partición de los datos restringe el volumen de datos que explora cada consulta, lo que mejora el rendimiento y reduce los costos. Puede particionar datos por cualquier clave. Para obtener más información, consulte [Particiones de datos en Athena](#).
  - Buckets (Buckets): (opcional) seleccione las columnas que quiera agrupar en buckets. La agrupación en buckets es una técnica que agrupa los datos en función de columnas específicas dentro de una sola partición. Estas columnas se conocen como claves de bucket. Al agrupar los datos relacionados en un solo bucket (un archivo dentro de una partición), se reduce significativamente la cantidad de datos escaneados por Athena, lo que mejora el rendimiento de las consultas y reduce los costos. Para obtener más información, consulte [Creación de particiones y asignación de buckets en Athena](#).
- e. En Preview table query (Vista previa de consulta de tablas), revise la consulta. Para ver la sintaxis de la consulta, consulte [CREATE TABLE AS](#).
- f. Elija Crear tabla.

## Para crear una consulta CTAS con una plantilla de SQL

Utilice la plantilla `CREATE TABLE AS SELECT` para crear una consulta CTAS en el editor de consultas.

1. En la consola de Athena, junto a **Tables and views** (Tablas y vistas), elija **Create table** (Crear tabla) y, a continuación, elija **CREATE TABLE AS SELECT**. De este modo, se rellena el editor de consultas con una consulta CTAS con valores de marcador de posición.
2. En el editor de consultas, edite la consulta según sea necesario. Para ver la sintaxis de la consulta, consulte [CREATE TABLE AS](#).
3. Elija **Ejecutar**.

Para ver ejemplos, consulte [Ejemplos de consultas CTAS](#).

## Creación de particiones y asignación de buckets en Athena

La creación de particiones y la asignación de buckets son dos formas de reducir la cantidad de datos que Athena debe analizar al ejecutar una consulta. La creación de particiones y la asignación de buckets son complementarias y se pueden utilizar juntas. Reducir la cantidad de datos analizados mejora el rendimiento y reduce los costos. Para obtener pautas generales sobre el rendimiento de las consultas de Athena, consulte [Los 10 mejores consejos para ajustar el rendimiento de Amazon Athena](#).

### ¿Qué es la creación de particiones?

Crear particiones significa organizar los datos en directorios (o “prefijos”) en Amazon S3 en función de una propiedad concreta de los datos. Estas propiedades se denominan claves de partición. Una clave de partición común es la fecha o alguna otra unidad de tiempo, como el año o el mes. Sin embargo, un conjunto de datos se puede particionar en más de una clave. Por ejemplo, los datos sobre las ventas de productos pueden particionarse por fecha, categoría de producto y mercado.

### Decidir cómo crear particiones

Las propiedades que se utilizan siempre o con frecuencia en las consultas y que tienen una cardinalidad baja son buenas candidatas para las claves de partición. Hay una disyuntiva entre tener demasiadas particiones y tener muy pocas. Con demasiadas particiones, el aumento del número de archivos genera una sobrecarga. El filtrado de las propias particiones también supone una sobrecarga. Con muy pocas particiones, las consultas suelen tener que analizar más datos.

## Creación de una tabla particionada

Cuando un conjunto de datos está particionado, puede crear una tabla particionada en Athena. Una tabla particionada es una tabla que tiene claves de partición. Cuando se utiliza `CREATE TABLE`, se agregan particiones a la tabla. Cuando se utiliza `CREATE TABLE AS`, las particiones que se crean en Amazon S3 se agregan automáticamente a la tabla.

En una instrucción `CREATE TABLE`, se especifican las claves de partición en la cláusula `PARTITIONED BY` (*column\_name data\_type*). En una instrucción `CREATE TABLE AS`, se especifican las claves de partición en una cláusula `WITH` (`partitioned_by = ARRAY['partition_key']`) o `WITH` (`partitioning = ARRAY['partition_key']`) en las tablas de Iceberg. Por motivos de rendimiento, las claves de partición siempre deben ser del tipo `STRING`. Para obtener más información, consulte [Uso de una cadena como tipo de datos para las claves de partición](#).

Para obtener información adicional sobre la sintaxis de `CREATE TABLE` y `CREATE TABLE AS`, consulte [CREATE TABLE](#) y [Propiedades de la tabla CTAS](#).

## Consulta de tablas particionadas

Cuando se consulta una tabla particionada, Athena usa los predicados de la consulta para filtrar la lista de particiones. A continuación, utiliza las ubicaciones de las particiones coincidentes para procesar los archivos encontrados. Athena puede reducir de manera eficiente la cantidad de datos analizados simplemente al no leer los datos de las particiones que no coinciden con los predicados de la consulta.

## Ejemplos

Supongamos que tiene una tabla particionada en `sales_date` y `product_category` y quiere saber los ingresos totales de una semana en una categoría específica. Debe incluir predicados en las columnas `sales_date` y `product_category` para garantizar que Athena analice solo la cantidad mínima de datos, como en el siguiente ejemplo.

```
SELECT SUM(amount) AS total_revenue
FROM sales
WHERE sales_date BETWEEN '2023-02-27' AND '2023-03-05'
AND product_category = 'Toys'
```

Supongamos que tiene un conjunto de datos que está particionado por fecha, pero que también tiene una marca de tiempo detallada.

Con las tablas de Iceberg, puede declarar que una clave de partición tiene una relación con una columna, pero con las tablas de Hive, el motor de consultas no conoce las relaciones entre las columnas y las claves de partición. Por este motivo, debe incluir un predicado tanto en la columna como en la clave de partición de la consulta para asegurarse de que la consulta no analice más datos de los necesarios.

Por ejemplo, supongamos que la tabla `sales` del ejemplo anterior también tiene una columna `sold_at` del tipo de datos `TIMESTAMP`. Si desea obtener los ingresos solo para un intervalo de tiempo específico, debe escribir la consulta de la siguiente manera:

```
SELECT SUM(amount) AS total_revenue
FROM sales
WHERE sales_date = '2023-02-28'
AND sold_at BETWEEN TIMESTAMP '2023-02-28 10:00:00' AND TIMESTAMP '2023-02-28
  12:00:00'
AND product_category = 'Toys'
```

Para obtener más información sobre esta diferencia entre las consultas de las tablas de Hive e Iceberg, consulte [Cómo escribir consultas para campos de marca de tiempo que también se encuentren particionados por tiempo](#).

## ¿Qué es la asignación de buckets?

La asignación de buckets es una forma de organizar los registros de un conjunto de datos en categorías denominadas buckets.

Este significado de bucket y creación de buckets es diferente del de bucket de Amazon S3 y no debe confundirse con este. En la asignación de buckets para datos, los registros que tienen el mismo valor para una propiedad se incluyen en el mismo bucket. Los registros se distribuyen de la forma más uniforme posible entre los buckets, de modo que cada uno de ellos tenga aproximadamente la misma cantidad de datos.

En la práctica, los buckets son archivos y una función hash determina el bucket al que se asigna un registro. Un conjunto de datos agrupado en buckets tendrá uno o más archivos por bucket y partición. El bucket al que pertenece un archivo está codificado en el nombre del archivo.

## Beneficios de la asignación de buckets

La asignación de buckets es útil cuando un conjunto de datos está agrupado en buckets según una propiedad específica y se desean recuperar registros en los que esa propiedad tiene un valor

determinado. Como los datos están agrupados en buckets, Athena puede utilizar el valor para determinar los archivos que se van a examinar. Por ejemplo, supongamos que un conjunto de datos está agrupado en buckets por `customer_id` y que usted desea buscar todos los registros de un cliente específico. Athena determina el bucket que contiene esos registros y solo lee los archivos de ese bucket.

Las columnas que presentan una alta cardinalidad (es decir, tienen muchos valores distintos), están distribuidas de manera uniforme y se consultan en busca de valores específicos con frecuencia, se consideran buenas candidatas para la asignación de datos.

#### Note

Athena no admite el uso de `INSERT INTO` para agregar nuevos registros a tablas agrupadas en buckets.

### Tipos de datos admitidos para filtrado en columnas en buckets

Puede agregar filtros en columnas agrupadas en buckets con determinados tipos de datos. Athena admite el filtrado en columnas agrupadas en buckets con los siguientes tipos de datos:

- BOOLEAN
- BYTE
- FECHA
- DOBLE
- FLOAT
- INT
- LONG
- SHORT
- STRING
- VARCHAR

### Soporte para Hive y Spark

La versión 2 del motor de Athena admite conjuntos de datos agrupados en buckets mediante el algoritmo de bucket Hive, y la versión 3 del motor de Athena también admite el algoritmo de bucket Apache Spark. Hive es el bucket predeterminado. Si el conjunto de datos está agrupado en



buckets mediante el algoritmo Spark, use la cláusula `TBLPROPERTIES` para establecer el valor de la propiedad `bucketing_format` en spark.

### Note

Athena tiene un límite de 100 particiones por consulta `CREATE TABLE AS SELECT (CTAS)`. Del mismo modo, solo puede agregar un máximo de 100 particiones a una tabla de destino con una instrucción `INSERT INTO`. Este límite de 100 solo se aplica cuando la tabla está agrupada en buckets además de particionada.

Si supera esta limitación, es posible que reciba el mensaje de error

`HIVE_TOO_MANY_OPEN_PARTITIONS: Exceeded limit of 100 open writers for partitions/buckets (HIVE_TOO_MANY_OPEN_PARTITIONS: Se ha superado el límite de 100 autores abiertos para particiones/buckets)`. Para evitar esta limitación, puede utilizar una instrucción `CTAS` y una serie de instrucciones `INSERT INTO` que crean o insertan hasta 100 particiones cada una. Para obtener más información, consulte [Uso de CTAS e INSERT INTO para evitar el límite de 100 particiones](#).

### Ejemplo de asignación de buckets con CREATE TABLE

Si desea crear una tabla para un conjunto de datos agrupado en buckets existente, use la cláusula `CLUSTERED BY (column)` seguida de la cláusula `INTO N BUCKETS`. La cláusula `INTO N BUCKETS` especifica el número de buckets en los que se agrupan los datos.

En el siguiente ejemplo `CREATE TABLE`, el conjunto de datos `sales` se agrupa por `customer_id` en 8 buckets mediante el algoritmo Spark. La instrucción `CREATE TABLE` usa las cláusulas `CLUSTERED BY` y `TBLPROPERTIES` para establecer las propiedades en consecuencia.

```
CREATE EXTERNAL TABLE sales (...)  
...  
CLUSTERED BY (`customer_id`) INTO 8 BUCKETS  
...  
TBLPROPERTIES (  
  'bucketing_format' = 'spark'  
)
```

### Ejemplo de asignación de buckets con CREATE TABLE AS (CTAS)

Para especificar la asignación de buckets con `CREATE TABLE AS`, utilice los parámetros `bucketed_by` y `bucket_count`, como en el siguiente ejemplo.

```
CREATE TABLE sales
WITH (
  ...
  bucketed_by = ARRAY['customer_id'],
  bucket_count = 8
)
AS SELECT ...
```

## Ejemplo de consulta de asignación de buckets

En el siguiente ejemplo de consulta, se buscan los nombres de los productos que un cliente específico ha comprado en el transcurso de una semana.

```
SELECT DISTINCT product_name
FROM sales
WHERE sales_date BETWEEN '2023-02-27' AND '2023-03-05'
AND customer_id = 'c123'
```

Si esta tabla está particionada por `sales_date` y agrupada en buckets por `customer_id`, Athena puede calcular el bucket en el que se encuentran los registros del cliente. Como máximo, Athena lee un archivo por partición.

## Recursos adicionales de

Para ver un ejemplo de `CREATE TABLE AS` en el que se crean tablas agrupadas en buckets y particionadas, consulte [Ejemplo: creación de tablas agrupadas en buckets y particionadas](#).

## Ejemplos de consultas CTAS

Utilice los siguientes ejemplos para crear consultas CTAS. Para obtener información sobre la sintaxis CTAS, consulte [CREATE TABLE AS](#).

En esta sección:

- [Example: Duplicating a table by selecting all columns](#)
- [Example: Selecting specific columns from one or more tables](#)
- [Example: Creating an empty copy of an existing table](#)
- [Example: Specifying data storage and compression formats](#)
- [Example: Writing query results to a different format](#)

- [Example: Creating unpartitioned tables](#)
- [Example: Creating partitioned tables](#)
- [Example: Creating bucketed and partitioned tables](#)
- [Example: Creating an Iceberg table with Parquet data](#)
- [Example: Creating an Iceberg table with Avro data](#)

Example Ejemplo : duplicación de una tabla mediante la selección de todas las columnas

En el siguiente ejemplo se crea una tabla copiando todas las columnas de una tabla:

```
CREATE TABLE new_table AS
SELECT *
FROM old_table;
```

En la siguiente variante del ejemplo anterior, la instrucción SELECT incluye también una cláusula WHERE. En este caso, la consulta solo selecciona las filas de la tabla que satisfacen la cláusula WHERE:

```
CREATE TABLE new_table AS
SELECT *
FROM old_table
WHERE condition;
```

Example Ejemplo : selección de columnas específicas de una o más tablas

En el siguiente ejemplo se crea una nueva consulta que se ejecuta en un conjunto de columnas de otra tabla:

```
CREATE TABLE new_table AS
SELECT column_1, column_2, ... column_n
FROM old_table;
```

Esta variante del mismo ejemplo crea una nueva tabla a partir de columnas específicas de varias tablas:

```
CREATE TABLE new_table AS
SELECT column_1, column_2, ... column_n
FROM old_table_1, old_table_2, ... old_table_n;
```

## Example Ejemplo : creación de una copia vacía de una tabla existente

En el siguiente ejemplo se utiliza `WITH NO DATA` para crear una nueva tabla, que está vacía y tiene el mismo esquema que la tabla original:

```
CREATE TABLE new_table
AS SELECT *
FROM old_table
WITH NO DATA;
```

## Example Ejemplo : especificación de los formatos de almacenamiento y compresión de datos

Con CTAS, puede utilizar una tabla de origen en un formato de almacenamiento para crear otra tabla en un formato de almacenamiento diferente.

Utilice la propiedad `format` para especificar ORC, PARQUET, AVRO, JSON o TEXTFILE como formato de almacenamiento para la nueva tabla.

Para los formatos de almacenamiento PARQUET, ORC, TEXTFILE y JSON, utilice la propiedad `write_compression` para especificar el formato de compresión para los datos de la nueva tabla. Para obtener información sobre los formatos de compresión que admite cada formato de archivo, consulte [Compatibilidad con la compresión de Athena](#).

En el siguiente ejemplo se especifica que los datos de la tabla `new_table` se almacenen en formato Parquet y utilicen la compresión Snappy. La compresión predeterminada de Parquet es GZIP.

```
CREATE TABLE new_table
WITH (
    format = 'Parquet',
    write_compression = 'SNAPPY')
AS SELECT *
FROM old_table;
```

En el siguiente ejemplo se especifica que los datos de la tabla `new_table` se almacenen en formato ORC y utilicen la compresión Snappy. La compresión predeterminada de ORC es ZLIB.

```
CREATE TABLE new_table
WITH (format = 'ORC',
    write_compression = 'SNAPPY')
AS SELECT *
FROM old_table ;
```

En el siguiente ejemplo se especifica que los datos de la tabla `new_table` se almacenen en formato de documento de texto y utilicen la compresión Snappy. La compresión predeterminada de los formatos de documento de texto y JSON es GZIP.

```
CREATE TABLE new_table
WITH (format = 'TEXTFILE',
      write_compression = 'SNAPPY')
AS SELECT *
FROM old_table ;
```

Example Ejemplo : escritura de los resultados de la consulta en un formato diferente

La siguiente consulta CTAS selecciona todos los registros de `old_table`, que podrían estar almacenados en CSV u otro formato, y crea una nueva tabla con los datos subyacentes guardados en Amazon S3 en formato ORC:

```
CREATE TABLE my_orc_ctas_table
WITH (
      external_location = 's3://DOC-EXAMPLE-BUCKET/my_orc_stas_table/',
      format = 'ORC')
AS SELECT *
FROM old_table;
```

Example Ejemplo : creación de tablas no particionadas

Los siguientes ejemplos crean tablas que no están particionadas. Los datos de las tablas se almacenan en diferentes formatos. Algunos de estos ejemplos especifican la ubicación externa.

En el siguiente ejemplo se crea una consulta CTAS que almacena los resultados como un archivo de texto:

```
CREATE TABLE ctas_csv_unpartitioned
WITH (
      format = 'TEXTFILE',
      external_location = 's3://DOC-EXAMPLE-BUCKET/ctas_csv_unpartitioned/')
AS SELECT key1, name1, address1, comment1
FROM table1;
```

En el siguiente ejemplo, los resultados se almacenan en Parquet y se utiliza la ubicación predeterminada para los resultados:

```
CREATE TABLE ctas_parquet_unpartitioned
WITH (format = 'PARQUET')
AS SELECT key1, name1, comment1
FROM table1;
```

En la siguiente consulta, la tabla se almacena en JSON y se seleccionan columnas específicas de los resultados de la tabla original:

```
CREATE TABLE ctas_json_unpartitioned
WITH (
    format = 'JSON',
    external_location = 's3://DOC-EXAMPLE-BUCKET/ctas_json_unpartitioned/')
AS SELECT key1, name1, address1, comment1
FROM table1;
```

En el siguiente ejemplo, el formato es ORC:

```
CREATE TABLE ctas_orc_unpartitioned
WITH (
    format = 'ORC')
AS SELECT key1, name1, comment1
FROM table1;
```

En el siguiente ejemplo, el formato es Avro:

```
CREATE TABLE ctas_avro_unpartitioned
WITH (
    format = 'AVRO',
    external_location = 's3://DOC-EXAMPLE-BUCKET/ctas_avro_unpartitioned/')
AS SELECT key1, name1, comment1
FROM table1;
```

### Example Ejemplo : creación de tablas particionadas

Los siguientes ejemplos muestran consultas CREATE TABLE AS SELECT para tablas con particiones en diferentes formatos de almacenamiento, que utilizan `partitioned_by` y otras propiedades en la cláusula WITH. Para ver la sintaxis, consulte [Propiedades de la tabla CTAS](#). Para obtener más información sobre cómo elegir las columnas para crear particiones, consulte [Creación de particiones y asignación de buckets en Athena](#).

**Note**

Indique las columnas particionadas al final de la lista de columnas en la instrucción SELECT. Puede particionar por varias columnas y tener hasta 100 combinaciones únicas de partición y bucket. Por ejemplo, puede tener 100 particiones si no se especifican buckets.

```
CREATE TABLE ctas_csv_partitioned
WITH (
  format = 'TEXTFILE',
  external_location = 's3://DOC-EXAMPLE-BUCKET/ctas_csv_partitioned/',
  partitioned_by = ARRAY['key1'])
AS SELECT name1, address1, comment1, key1
FROM tables1;
```

```
CREATE TABLE ctas_json_partitioned
WITH (
  format = 'JSON',
  external_location = 's3://DOC-EXAMPLE-BUCKET/ctas_json_partitioned/',
  partitioned_by = ARRAY['key1'])
AS select name1, address1, comment1, key1
FROM table1;
```

**Example Ejemplo : creación de tablas particionadas y en buckets**

En el siguiente ejemplo se muestra una consulta CREATE TABLE AS SELECT que utiliza particiones y buckets para almacenar los resultados en Amazon S3. Los resultados de las tablas están particionados y distribuidos en buckets por columnas diferentes. Athena admite un máximo de 100 combinaciones únicas de partición y bucket. Por ejemplo, si crea una tabla con cinco buckets, se admiten 20 particiones con cinco buckets cada una. Para ver la sintaxis, consulte [Propiedades de la tabla CTAS](#).

Para obtener información sobre cómo elegir las columnas para la asignación de buckets, consulte [Creación de particiones y asignación de buckets en Athena](#).

```
CREATE TABLE ctas_avro_bucketed
WITH (
  format = 'AVRO',
  external_location = 's3://DOC-EXAMPLE-BUCKET/ctas_avro_bucketed/',
  partitioned_by = ARRAY['nationkey'],
```

```
        bucketed_by = ARRAY['mktsegment'],
        bucket_count = 3)
AS SELECT key1, name1, address1, phone1, acctbal, mktsegment, comment1, nationkey
FROM table1;
```

### Example Ejemplo: creación de una tabla de Iceberg con datos de Parquet

En el siguiente ejemplo, se crea una tabla de Iceberg con archivos de datos de Parquet. Los archivos se particionan por mes mediante la columna `dt` de `table1`. El ejemplo actualiza las propiedades de retención de la tabla para que se retengan 10 instantáneas de forma predeterminada en cada ramificación de la tabla. También se retienen las instantáneas de los últimos 7 días. Para obtener más información sobre las propiedades de las tablas de Iceberg, consulte [Propiedades de la tabla](#).

```
CREATE TABLE ctas_iceberg_parquet
WITH (table_type = 'ICEBERG',
      format = 'PARQUET',
      location = 's3://DOC-EXAMPLE-BUCKET/ctas_iceberg_parquet/',
      is_external = false,
      partitioning = ARRAY['month(dt)'],
      vacuum_min_snapshots_to_keep = 10,
      vacuum_max_snapshot_age_seconds = 604800
)
AS SELECT key1, name1, dt FROM table1;
```

### Example Ejemplo: creación de una tabla de Iceberg con datos de Avro

En el siguiente ejemplo, se crea una tabla de Iceberg con archivos de datos de Avro particionados por `key1`.

```
CREATE TABLE ctas_iceberg_avro
WITH ( format = 'AVRO',
      location = 's3://DOC-EXAMPLE-BUCKET/ctas_iceberg_avro/',
      is_external = false,
      table_type = 'ICEBERG',
      partitioning = ARRAY['key1'])
AS SELECT key1, name1, date FROM table1;
```

## Uso de CTAS e INSERT INTO en ETL y análisis de datos

Utilice las instrucciones Create Table as Select ([CTAS](#)) e [INSERT INTO](#) en Athena para extraer, transformar y cargar (ETL) datos en Amazon S3 para el procesamiento de datos. En este tema se



muestra cómo utilizar estas instrucciones para crear particiones y convertir un conjunto de datos al formato de datos en columnas a fin de optimizarlo para el análisis de datos.

Las instrucciones CTAS utilizan consultas [SELECT](#) estándar para crear nuevas tablas. Puede utilizar una instrucción CTAS para crear un subconjunto de datos para su análisis. En una instrucción CTAS, puede particionar los datos, especificar su compresión y convertir los datos en un formato de columnas como Apache Parquet o Apache ORC. Cuando ejecuta la consulta CTAS, las tablas y particiones que crea se añaden automáticamente a [AWS Glue Data Catalog](#). Esto hace que las nuevas tablas y particiones que crea estén disponibles inmediatamente para consultas posteriores.

Las instrucciones INSERT INTO insertan nuevas filas en una tabla de destino basándose en una instrucción de consulta SELECT que se ejecuta en una tabla de origen. Puede utilizar instrucciones INSERT INTO para transformar y cargar datos de tabla de origen en formato CSV en datos de tabla de destino utilizando todas las transformaciones compatibles con CTAS.

## Información general

En Athena, utilice una instrucción CTAS para realizar una conversión inicial por lotes de los datos. A continuación, utilice varias instrucciones INSERT INTO para realizar actualizaciones incrementales en la tabla creada por la instrucción CTAS.

### Pasos

- [Paso 1: Crear una tabla basada en el conjunto de datos original](#)
- [Paso 2: Utilizar CTAS para particionar, convertir y comprimir los datos](#)
- [Paso 3: Utilizar INSERT INTO para agregar datos](#)
- [Paso 4: Medir las diferencias de rendimiento y costo](#)

### Paso 1: Crear una tabla basada en el conjunto de datos original

En el ejemplo de este tema, se utiliza un subconjunto legible de Simple Storage Service (Amazon S3) del conjunto de datos [Diario de la Red Global de Climatología Histórica de NOAA \(GHCN-D\)](#) disponible de forma pública. Los datos de Amazon S3 tienen las siguientes características.

```
Location: s3://aws-bigdata-blog/artifacts/athena-ctas-insert-into-blog/  
Total objects: 41727  
Size of CSV dataset: 11.3 GB  
Region: us-east-1
```

Los datos originales se almacenan en Amazon S3 sin particiones. Los datos están en formato CSV en archivos como el siguiente.

```
2019-10-31 13:06:57 413.1 KiB artifacts/athena-ctas-insert-into-blog/2010.csv0000
2019-10-31 13:06:57 412.0 KiB artifacts/athena-ctas-insert-into-blog/2010.csv0001
2019-10-31 13:06:57 34.4 KiB artifacts/athena-ctas-insert-into-blog/2010.csv0002
2019-10-31 13:06:57 412.2 KiB artifacts/athena-ctas-insert-into-blog/2010.csv0100
2019-10-31 13:06:57 412.7 KiB artifacts/athena-ctas-insert-into-blog/2010.csv0101
```

Los tamaños de archivo de esta muestra son relativamente pequeños. Al fusionarlos en archivos más grandes, puede reducir el número total de archivos, lo que permite ejecutar mejor las consultas. Puede utilizar instrucciones CTAS e INSERT INTO para mejorar el rendimiento de la consulta.

Para crear una base de datos y una tabla basadas en el conjunto de datos de ejemplo

1. En la consola de Athena, elija US East (N. Virginia) (Este de EE. UU. [Norte de Virginia]) como Región de AWS. Asegúrese de ejecutar todas las consultas de este tutorial en us-east-1.
2. En el Editor de consultas de Athena, ejecute el comando [CREATE DATABASE](#) para crear una base de datos.

```
CREATE DATABASE blogdb
```

3. Ejecute la siguiente instrucción para [crear una tabla](#).

```
CREATE EXTERNAL TABLE `blogdb`.`original_csv` (  
  `id` string,  
  `date` string,  
  `element` string,  
  `datavalue` bigint,  
  `mflag` string,  
  `qflag` string,  
  `sflag` string,  
  `obstime` bigint)  
ROW FORMAT DELIMITED  
  FIELDS TERMINATED BY ','  
STORED AS INPUTFORMAT  
  'org.apache.hadoop.mapred.TextInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat'  
LOCATION  
  's3://aws-bigdata-blog/artifacts/athena-ctas-insert-into-blog/'
```

## Paso 2: Utilizar CTAS para particionar, convertir y comprimir los datos

Después de crear una tabla, puede utilizar una sola instrucción [CTAS](#) para convertir los datos al formato Parquet con compresión Snappy y para particionar los datos por año.

La tabla que creó en el paso 1 tiene un campo `date` con formato de fecha `YYYYMMDD` (por ejemplo, `20100104`). Dado que la nueva tabla se particionará en `year`, la instrucción de ejemplo del procedimiento siguiente utiliza la función Presto `substr("date", 1, 4)` para extraer el valor `year` del campo `date`.

Para convertir los datos al formato Parquet con compresión Snappy haciendo particiones por año

- Ejecute la siguiente instrucción CTAS, reemplazando *your-bucket* por la ubicación de su bucket de Amazon S3.

```
CREATE table new_parquet
WITH (format='PARQUET',
parquet_compression='SNAPPY',
partitioned_by=array['year'],
external_location = 's3://DOC-EXAMPLE-BUCKET/optimized-data/')
AS
SELECT id,
       date,
       element,
       datavalue,
       mflag,
       qflag,
       sflag,
       obstime,
       substr("date",1,4) AS year
FROM original_csv
WHERE cast(substr("date",1,4) AS bigint) >= 2015
      AND cast(substr("date",1,4) AS bigint) <= 2019
```

### Note

En este ejemplo, la tabla que ha creado solo incluye los datos de 2015 a 2019. En el paso 3, añada nuevos datos a esta tabla mediante el comando `INSERT INTO`.

Cuando finaliza la consulta, se utiliza el siguiente procedimiento para comprobar el resultado en la ubicación de Amazon S3 especificada en la instrucción CTAS.

Para ver las particiones y los archivos de parquet creados por la instrucción CTAS

1. Para mostrar las particiones creadas, ejecute el siguiente comando en la AWS CLI. Asegúrese de incluir la barra diagonal (/) al final.

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/optimized-data/
```

La salida muestra las particiones.

```
PRE year=2015/  
PRE year=2016/  
PRE year=2017/  
PRE year=2018/  
PRE year=2019/
```

2. Para ver los archivos Parquet, ejecute el siguiente comando. Tenga en cuenta que la opción `| head -5`, que restringe la salida a los cinco primeros resultados, no está disponible para Windows.

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/optimized-data/ --recursive --human-readable |  
head -5
```

La salida se parece a la siguiente.

```
2019-10-31 14:51:05    7.3 MiB optimized-data/  
year=2015/20191031_215021_00001_3f42d_1be48df2-3154-438b-b61d-8fb23809679d  
2019-10-31 14:51:05    7.0 MiB optimized-data/  
year=2015/20191031_215021_00001_3f42d_2a57f4e2-ffa0-4be3-9c3f-28b16d86ed5a  
2019-10-31 14:51:05    9.9 MiB optimized-data/  
year=2015/20191031_215021_00001_3f42d_34381db1-00ca-4092-bd65-ab04e06dc799  
2019-10-31 14:51:05    7.5 MiB optimized-data/  
year=2015/20191031_215021_00001_3f42d_354a2bc1-345f-4996-9073-096cb863308d  
2019-10-31 14:51:05    6.9 MiB optimized-data/  
year=2015/20191031_215021_00001_3f42d_42da4cfd-6e21-40a1-8152-0b902da385a1
```

### Paso 3: Utilizar INSERT INTO para agregar datos

En el paso 2, utilizó CTAS para crear una tabla con particiones para los años 2015 a 2019. Sin embargo, el conjunto de datos original también contiene datos para los años 2010 a 2014. Ahora añade esos datos usando una instrucción [INSERT INTO](#) .

Para añadir datos a la tabla mediante una o más instrucciones INSERT INTO

1. Ejecute el siguiente comando INSERT INTO, especificando los años anteriores a 2015 en la cláusula WHERE.

```
INSERT INTO new_parquet
SELECT id,
       date,
       element,
       datavalue,
       mflag,
       qflag,
       sflag,
       obstime,
       substr("date",1,4) AS year
FROM original_csv
WHERE cast(substr("date",1,4) AS bigint) < 2015
```

2. Vuelva a ejecutar el comando `aws s3 ls`, utilizando la sintaxis siguiente.

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/optimized-data/
```

La salida muestra las nuevas particiones.

```
PRE year=2010/
PRE year=2011/
PRE year=2012/
PRE year=2013/
PRE year=2014/
PRE year=2015/
PRE year=2016/
PRE year=2017/
PRE year=2018/
PRE year=2019/
```

- Para ver la reducción en el tamaño del conjunto de datos obtenido mediante compresión y almacenamiento en columnas en formato Parquet, ejecute el siguiente comando.

```
aws s3 ls s3://DOC-EXAMPLE-BUCKET/optimized-data/ --recursive --human-readable --summarize
```

Los siguientes resultados muestran que el tamaño del conjunto de datos después de Parquet con compresión Snappy es de 1,2 GB.

```
...
2020-01-22 18:12:02 2.8 MiB optimized-data/
year=2019/20200122_181132_00003_nja5r_f0182e6c-38f4-4245-afa2-9f5bfa8d6d8f
2020-01-22 18:11:59 3.7 MiB optimized-data/
year=2019/20200122_181132_00003_nja5r_fd9906b7-06cf-4055-a05b-f050e139946e
Total Objects: 300
Total Size: 1.2 GiB
```

- Si se añaden más datos CSV a la tabla original, puede añadir esos datos a la tabla Parquet mediante instrucciones INSERT INTO. Por ejemplo, si quiere insertar nuevos datos para el año 2020, puede ejecutar la siguiente instrucción INSERT INTO. La instrucción añade los datos y la partición relevante a la tabla new\_parquet.

```
INSERT INTO new_parquet
SELECT id,
       date,
       element,
       datavalue,
       mflag,
       qflag,
       sflag,
       obstime,
       substr("date",1,4) AS year
FROM original_csv
WHERE cast(substr("date",1,4) AS bigint) = 2020
```

#### Note

La instrucción INSERT INTO admite escribir un máximo de 100 particiones en la tabla de destino. Sin embargo, para agregar más de 100 particiones, puede ejecutar varias

instrucciones INSERT INTO. Para obtener más información, consulte [Uso de CTAS e INSERT INTO para evitar el límite de 100 particiones](#).

## Paso 4: Medir las diferencias de rendimiento y costo

Después de transformar los datos, puede medir las ganancias de rendimiento y el ahorro de costes ejecutando las mismas consultas en las tablas nuevas y antiguas y comparando los resultados.

### Note

Para obtener información sobre los costos por consulta de Athena, consulte [Precios de Amazon Athena](#).

Para medir las ganancias de rendimiento y las diferencias de costos

1. Ejecute la siguiente consulta en la tabla original. La consulta busca el número de identificadores distintos para cada valor del año.

```
SELECT substr("date",1,4) as year,  
       COUNT(DISTINCT id)  
FROM original_csv  
GROUP BY 1 ORDER BY 1 DESC
```

2. Apunte la hora a la que se ejecutó la consulta y la cantidad de datos analizados.
3. Ejecute la misma consulta en la nueva tabla, y anote el tiempo de ejecución de la consulta y la cantidad de datos analizados.

```
SELECT year,  
       COUNT(DISTINCT id)  
FROM new_parquet  
GROUP BY 1 ORDER BY 1 DESC
```

4. Compare los resultados y calcule la diferencia de rendimiento y costo. Los siguientes resultados de ejemplo muestran que la consulta de prueba de la nueva tabla era más rápida y barata que la consulta de la tabla anterior.

Tabla	Tiempo de ejecución	Datos escaneados
Original	16,88 segundos	11,35 GB
New	3,79 segundos	428,05 MB

5. Ejecute la siguiente consulta de ejemplo en la tabla original. La consulta calcula la temperatura máxima promedio (Celsius), la temperatura mínima promedio (Celsius) y la precipitación media (mm) de la Tierra en 2018.

```
SELECT element, round(avg(CAST(datavalue AS real)/10),2) AS value
FROM original_csv
WHERE element IN ('TMIN', 'TMAX', 'PRCP') AND substr("date",1,4) = '2018'
GROUP BY 1
```

6. Apunte la hora a la que se ejecutó la consulta y la cantidad de datos analizados.
7. Ejecute la misma consulta en la nueva tabla, y anote el tiempo de ejecución de la consulta y la cantidad de datos analizados.

```
SELECT element, round(avg(CAST(datavalue AS real)/10),2) AS value
FROM new_parquet
WHERE element IN ('TMIN', 'TMAX', 'PRCP') and year = '2018'
GROUP BY 1
```

8. Compare los resultados y calcule la diferencia de rendimiento y costo. Los siguientes resultados de ejemplo muestran que la consulta de prueba de la nueva tabla era más rápida y barata que la consulta de la tabla anterior.

Tabla	Tiempo de ejecución	Datos escaneados
Original	18,65 segundos	11,35 GB
New	1,92 segundos	68 MB

## Resumen

En este tema se muestra cómo realizar operaciones ETL mediante instrucciones CTAS e INSERT INTO en Athena. Ha realizado el primer conjunto de transformaciones mediante una instrucción



CTAS que ha convertido los datos al formato Parquet con compresión Snappy. La instrucción CTAS también convirtió el conjunto de datos de no particionado a particionado. Esto redujo su tamaño y redujo los costos de ejecución de las consultas. Cuando haya nuevos datos disponibles, puede utilizar una instrucción `INSERT INTO` para transformar y cargar los datos en la tabla que creó con la instrucción CTAS.

## Uso de CTAS e `INSERT INTO` para evitar el límite de 100 particiones

Athena tiene un límite de 100 particiones por consulta `CREATE TABLE AS SELECT` ([CTAS](#)). Del mismo modo, puede añadir un máximo de 100 particiones a una tabla de destino con una instrucción [INSERT INTO](#).

Si supera esta limitación, es posible que reciba el mensaje de error `HIVE_TOO_MANY_OPEN_PARTITIONS: Exceeded limit of 100 open writers for partitions/buckets` (`HIVE_TOO_MANY_OPEN_PARTITIONS: Se ha superado el límite de 100 autores abiertos para particiones/buckets`). Para evitar esta limitación, puede utilizar una instrucción CTAS y una serie de instrucciones `INSERT INTO` que crean o insertan hasta 100 particiones cada una.

El ejemplo de este tema utiliza una base de datos llamada `tpch100` cuyos datos residen en la ubicación del bucket de Amazon S3 `s3://DOC-EXAMPLE-BUCKET/`.

Para utilizar CTAS e `INSERT INTO` para crear una tabla de más de 100 particiones

1. Utilice una instrucción `CREATE EXTERNAL TABLE` para crear una tabla con particiones en el campo que desee.

La siguiente instrucción de ejemplo divide los datos mediante la columna `l_shipdate`. La tabla tiene 2525 particiones.

```
CREATE EXTERNAL TABLE `tpch100.lineitem_parq_partitioned`(  
  `l_orderkey` int,  
  `l_partkey` int,  
  `l_suppkey` int,  
  `l_linenumber` int,  
  `l_quantity` double,  
  `l_extendedprice` double,  
  `l_discount` double,  
  `l_tax` double,  
  `l_returnflag` string,  
  `l_linestatus` string,  
  `l_commitdate` string,
```

```

`l_receiptdate` string,
`l_shipinstruct` string,
`l_comment` string)
PARTITIONED BY (
  `l_shipdate` string)
ROW FORMAT SERDE
  'org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe' STORED AS
INPUTFORMAT
  'org.apache.hadoop.hive.q1.io.parquet.MapredParquetInputFormat' OUTPUTFORMAT
  'org.apache.hadoop.hive.q1.io.parquet.MapredParquetOutputFormat' LOCATION
's3://DOC-EXAMPLE-BUCKET/lineitem/'

```

2. Ejecute un comando `SHOW PARTITIONS <table_name>` como el siguiente para enumerar las particiones.

```
SHOW PARTITIONS lineitem_parq_partitioned
```

Los siguientes son resultados parciales de la muestra.

```

/*
l_shipdate=1992-01-02
l_shipdate=1992-01-03
l_shipdate=1992-01-04
l_shipdate=1992-01-05
l_shipdate=1992-01-06

...

l_shipdate=1998-11-24
l_shipdate=1998-11-25
l_shipdate=1998-11-26
l_shipdate=1998-11-27
l_shipdate=1998-11-28
l_shipdate=1998-11-29
l_shipdate=1998-11-30
l_shipdate=1998-12-01
*/

```

3. Ejecute una consulta CTAS para crear una tabla con particiones.

En el siguiente ejemplo se crea una tabla llamada `my_lineitem_parq_partitioned` y se utiliza la cláusula `WHERE` para restringir la `DATE` a antes de `1992-02-01`. Dado que el conjunto de datos de muestra comienza en enero de 1992, solo se crean particiones para enero de 1992.

```
CREATE table my_lineitem_parq_partitioned
WITH (partitioned_by = ARRAY['l_shipdate']) AS
SELECT l_orderkey,
       l_partkey,
       l_suppkey,
       l_linenumber,
       l_quantity,
       l_extendedprice,
       l_discount,
       l_tax,
       l_returnflag,
       l_linestatus,
       l_commitdate,
       l_receiptdate,
       l_shipinstruct,
       l_comment,
       l_shipdate
FROM tpch100.lineitem_parq_partitioned
WHERE cast(l_shipdate as timestamp) < DATE ('1992-02-01');
```

4. Ejecute el comando `SHOW PARTITIONS` para comprobar que la tabla contiene las particiones que desea.

```
SHOW PARTITIONS my_lineitem_parq_partitioned;
```

Las particiones del ejemplo son de enero de 1992.

```
/*
l_shipdate=1992-01-02
l_shipdate=1992-01-03
l_shipdate=1992-01-04
l_shipdate=1992-01-05
l_shipdate=1992-01-06
l_shipdate=1992-01-07
l_shipdate=1992-01-08
l_shipdate=1992-01-09
l_shipdate=1992-01-10
```

```
l_shipdate=1992-01-11
l_shipdate=1992-01-12
l_shipdate=1992-01-13
l_shipdate=1992-01-14
l_shipdate=1992-01-15
l_shipdate=1992-01-16
l_shipdate=1992-01-17
l_shipdate=1992-01-18
l_shipdate=1992-01-19
l_shipdate=1992-01-20
l_shipdate=1992-01-21
l_shipdate=1992-01-22
l_shipdate=1992-01-23
l_shipdate=1992-01-24
l_shipdate=1992-01-25
l_shipdate=1992-01-26
l_shipdate=1992-01-27
l_shipdate=1992-01-28
l_shipdate=1992-01-29
l_shipdate=1992-01-30
l_shipdate=1992-01-31
*/
```

5. Utilice una instrucción `INSERT INTO` para añadir particiones a la tabla.

En el siguiente ejemplo se añaden particiones para las fechas del mes de febrero de 1992.

```
INSERT INTO my_lineitem_parq_partitioned
SELECT l_orderkey,
       l_partkey,
       l_suppkey,
       l_linenum,
       l_quantity,
       l_extendedprice,
       l_discount,
       l_tax,
       l_returnflag,
       l_linestatus,
       l_commitdate,
       l_receiptdate,
       l_shipinstruct,
       l_comment,
       l_shipdate
FROM tpch100.lineitem_parq_partitioned
```

```
WHERE cast(l_shipdate as timestamp) >= DATE ('1992-02-01')
AND cast(l_shipdate as timestamp) < DATE ('1992-03-01');
```


6. Vuelva a ejecutar `SHOW PARTITIONS`.

```
SHOW PARTITIONS my_lineitem_parq_partitioned;
```

La tabla de muestra tiene ahora particiones tanto de enero como de febrero de 1992.

```
/*
l_shipdate=1992-01-02
l_shipdate=1992-01-03
l_shipdate=1992-01-04
l_shipdate=1992-01-05
l_shipdate=1992-01-06
...
l_shipdate=1992-02-20
l_shipdate=1992-02-21
l_shipdate=1992-02-22
l_shipdate=1992-02-23
l_shipdate=1992-02-24
l_shipdate=1992-02-25
l_shipdate=1992-02-26
l_shipdate=1992-02-27
l_shipdate=1992-02-28
l_shipdate=1992-02-29
*/
```

7. Siga utilizando instrucciones `INSERT INTO` que lean y no agreguen más de 100 particiones cada una. Continúe hasta que alcance el número de particiones necesario.

 Important

Al establecer la condición `WHERE`, asegúrese de que las consultas no se superpongan. De lo contrario, algunas particiones podrían tener datos duplicados.

# Referencia del SerDe

Athena admite varias bibliotecas SerDe para analizar datos de diferentes formatos de datos, como CSV, JSON, Parquet y ORC. Athena no admite SerDes personalizados.

## Temas

- [Uso de un SerDe](#)
- [Formatos de datos y SerDes compatibles](#)

## Uso de un SerDe

Un SerDe (serializador/deserializador) es un método que Athena emplea para interactuar con datos en varios formatos.

Es el SerDe especificado, y no el DDL, el que define el esquema de la tabla. En otras palabras, el SerDe puede anular la configuración de DDL que ha especificado en Athena al crear la tabla.

## Para utilizar un SerDe en consultas

Para utilizar un SerDe al crear una tabla en Athena, utilice uno de los métodos a continuación:

- Especifique `ROW FORMAT DELIMITED` y, a continuación, utilice instrucciones DDL para especificar delimitadores de campo, como en el ejemplo siguiente. Cuando especifique `ROW FORMAT DELIMITED`, Athena utiliza `LazySimpleSerDe` de forma predeterminada.

```
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
ESCAPED BY '\\\'
COLLECTION ITEMS TERMINATED BY '|'
MAP KEYS TERMINATED BY ':'
```

Para ver ejemplos de `ROW FORMAT DELIMITED`, consulte los siguientes temas:

[LazySimpleSerDe para CSV, TSV y archivos con delimitación personalizada](#)

[Consultas de registros de Amazon CloudFront](#)

[Consultas de los registros de Amazon EMR](#)

[Consulta de los registros de flujo de Amazon VPC](#)

## [Uso de CTAS e INSERT INTO en ETL y análisis de datos](#)

- Utilice ROW FORMAT SERDE para especificar explícitamente el tipo de SerDe que Athena debe usar al leer y escribir datos en la tabla. En el ejemplo siguiente, se especifica LazySimpleSerDe. Para especificar los delimitadores, utilice WITH SERDEPROPERTIES. Las propiedades especificadas por WITH SERDEPROPERTIES corresponden a las declaraciones separadas (como FIELDS TERMINATED BY) en el ejemplo ROW FORMAT DELIMITED.

```
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'  
WITH SERDEPROPERTIES (  
  'serialization.format' = ',',  
  'field.delim' = ',',  
  'collection.delim' = '|',  
  'mapkey.delim' = ':',  
  'escape.delim' = '\\'  
)
```

Para ver ejemplos de ROW FORMAT SERDE, consulte los siguientes temas:

[El SerDe de Avro](#)

[El SerDe de Grok](#)

[Bibliotecas de SerDe JSON](#)

[OpenCSVSerDe para el procesamiento de CSV](#)

[SerDe de Regex](#)

## Formatos de datos y SerDes compatibles

Athena permite crear tablas y consultar datos de archivos en formato CSV, TSV, con delimitación personalizada, y JSON; datos en formatos relacionados con Hadoop: ORC, Apache Avro y Parquet; registros de Logstash, registros de AWS CloudTrail y registros de Apache WebServer.

### Note

Athena utiliza los formatos que se indican en esta sección para leer los datos. Para obtener información sobre los formatos que utiliza Athena para escribir datos cuando ejecuta

consultas CTAS, consulte [Creación de una tabla a partir de los resultados de una consulta \(CTAS\)](#).

Para crear tablas y consultar datos de archivos con estos formatos en Athena, especifique una clase de serializador-deserializador (SerDe) para que Athena sepa qué formato se utiliza y cómo analizar los datos.

Esta tabla enumera los formatos de datos compatibles con Athena y sus bibliotecas SerDe correspondientes.

Un SerDe es una biblioteca personalizada que indica al catálogo de datos que utiliza Athena cómo gestionar los datos. El tipo de SerDe se especifica indicándolo explícitamente en la parte ROW FORMAT de la instrucción CREATE TABLE en Athena. En algunos casos, puede omitir el nombre del SerDe, porque Athena utiliza algunos tipos de SerDe de forma predeterminada para ciertos tipos de formatos de datos.

#### Formatos de datos compatibles y SerDes

Formato de los datos	Descripción	Tipos de SerDe admitidos en Athena
Amazon Ion	Amazon Ion es un formato de datos altamente codificado y autodescriptivo que constituye un superconjunto de JSON, de código abierto y desarrollado por Amazon.	Utilice <a href="#">SerDe de Amazon Ion Hive</a> .
Apache Avro	Formato de almacenamiento de datos en Hadoop que utiliza esquemas basados en JSON para registrar valores.	Utilice <a href="#">El SerDe de Avro</a> .
Apache Parquet	Formato para almacenamiento de datos en columnas en Hadoop.	Utilice la compresión <a href="#">El SerDe de Parquet</a> y SNAPPY.



Formato de los datos	Descripción	Tipos de SerDe admitidos en Athena
Registros de Apache WebServer	Formato para almacenar registros de Apache WebServer.	Use <a href="#">El SerDe de Grok</a> o <a href="#">SerDe de Regex</a> .
Registros de CloudTrail	Formato para almacenar registros de CloudTrail.	<ul style="list-style-type: none"> <li>Utilice la <a href="#">El SerDe JSON de Hive</a>. Para obtener más información, consulte <a href="#">Consulta de registros de AWS CloudTrail</a>.</li> </ul>
CSV (valores separados por comas)	En los datos con formato CSV, cada línea representa un registro de datos y cada registro se compone de uno o varios campos separados por comas.	<ul style="list-style-type: none"> <li>Utilice <a href="#">LazySimpleSerDe para CSV, TSV y archivos con delimitación personalizada</a> si los datos no incluyen valores entre comillas o si utiliza el formato <code>java.sql.Timestamp</code>.</li> <li>Utilice <a href="#">OpenCSVSerDe para el procesamiento de CSV</a> cuando los datos incluyan comillas en valores o utilicen el formato numérico de UNIX para <code>TIMESTAMP</code> (por ejemplo, <code>1564610311</code>).</li> </ul>
Delimitación personalizada	En los datos con este formato, cada línea representa un registro de datos y los registros están separados por un delimitadores de caracteres únicos personalizado.	Utilice <a href="#">LazySimpleSerDe para CSV, TSV y archivos con delimitación personalizada</a> y especifique un delimitador o personalizado de un solo carácter.

Formato de los datos	Descripción	Tipos de SerDe admitidos en Athena
JSON (JavaScript Object Notation)	En los datos JSON, cada línea representa un registro de datos y cada registro se compone de pares y matrices atributo-valor separados por comas.	<ul style="list-style-type: none"> <li>Utilice <a href="#">El SerDe JSON de Hive</a>.</li> <li>Utilice <a href="#">El SerDe JSON de OpenX</a>.</li> </ul>
Registros de Logstash	Formato para almacenar registros de Logstash.	Utilice <a href="#">El SerDe de Grok</a> .
ORC (Almacenamiento de filas en columnas optimizado)	Formato para almacenamiento en columnas optimizado de datos de Hive.	Utilice la compresión <a href="#">El SerDe de ORC</a> y ZLIB.
TSV (valores separados por tabulaciones)	En los datos con formato TSV, cada línea representa un registro de datos y cada registro se compone de uno o varios campos separados por tabuladores.	Utilice <a href="#">LazySimpleSerDe para CSV, TSV y archivos con delimitación personalizada</a> y especifique el carácter separador como FIELDS TERMINATED BY '\t' .

## Temas

- [SerDe de Amazon Ion Hive](#)
- [El SerDe de Avro](#)
- [El SerDe de Grok](#)
- [Bibliotecas de SerDe JSON](#)
- [LazySimpleSerDe para CSV, TSV y archivos con delimitación personalizada](#)
- [OpenCSVSerDe para el procesamiento de CSV](#)
- [El SerDe de ORC](#)
- [El SerDe de Parquet](#)
- [SerDe de Regex](#)

## SerDe de Amazon Ion Hive

Puede utilizar el SerDe de Amazon Ion Hive para consultar datos almacenados en formato [Amazon Ion](#). Amazon Ion es un formato de datos altamente codificado, autodescriptivo y de código abierto. El formato Amazon Ion se utiliza con servicios como [Amazon Quantum Ledger Database](#) (Amazon QLDB) y en el lenguaje de consulta SQL de código abierto [PartiQL](#).

Amazon Ion tiene formatos binarios y de texto que son intercambiables. Esta característica combina la facilidad de uso del texto con la eficacia de la codificación binaria.

Para consultar los datos de Amazon Ion desde Athena, puede utilizar el [SerDe de Amazon Ion Hive](#), que serializa y deserializa los datos de Amazon Ion. La deserialización permite ejecutar consultas sobre los datos de Amazon Ion o leerlos para escribir en un formato diferente como Parquet u ORC. La serialización permite generar datos en formato Amazon Ion mediante consultas CREATE TABLE AS SELECT (CTAS) o INSERT INTO para copiar datos de tablas existentes.

### Note

Dado que Amazon Ion es un superconjunto de JSON, puede utilizar el SerDe de Amazon Ion Hive para consultar conjuntos de datos JSON que no sean de Amazon Ion. A diferencia de otras [bibliotecas de SerDe JSON](#), el SerDe de Amazon Ion no espera que cada fila de datos esté en una sola línea. Esta característica resulta útil si desea consultar conjuntos de datos JSON que presentan “un buen formato de impresión” o dividir los campos en una fila con caracteres de nueva línea.

Para obtener información adicional y ejemplos de consultas de Amazon Ion con Athena, consulte [Analyze Amazon Ion datasets using Amazon Athena](#) (Analizar conjuntos de datos de Amazon Ion con Amazon Athena).

### Nombre del SerDe

- [com.amazon.ionhiveserde.IonHiveSerDe](#)

### Consideraciones y limitaciones

- Campos duplicados: las estructuras de Amazon Ion están ordenadas y admiten campos duplicados, mientras que STRUCT<> y MAP<> de Hive no lo hacen. Por lo tanto, cuando se

deserializa un campo duplicado de una estructura de Amazon Ion, se elige un solo valor de forma no determinista y los demás se ignoran.

- Tablas de símbolos externas no compatibles: en la actualidad, Athena no admite tablas de símbolos externas ni las siguientes propiedades del SerDe de Amazon Ion Hive:
  - `ion.catalog.class`
  - `ion.catalog.file`
  - `ion.catalog.url`
  - `ion.symbol_table_imports`
- Extensiones de archivo: Amazon Ion utiliza extensiones de archivo para determinar qué códec de compresión se debe utilizar para deserializar archivos de Amazon Ion. Por lo tanto, los archivos comprimidos deben tener la extensión de archivo que corresponde al algoritmo de compresión utilizado. Por ejemplo, si se utiliza ZSTD, los archivos correspondientes deben tener la extensión `.zst`.
- Datos homogéneos: Amazon Ion no tiene restricciones sobre los tipos de datos que se pueden utilizar para valores en campos determinados. Por ejemplo, dos documentos de Amazon Ion diferentes pueden tener un campo con el mismo nombre, pero con tipos de datos diferentes. Sin embargo, dado que Hive utiliza un esquema, todos los valores que se extraen en una sola columna de Hive deben tener el mismo tipo de datos.
- Restricciones del tipo de clave de asignación: cuando serialice datos de otro formato en Amazon Ion, asegúrese de que el tipo de clave de asignación sea uno de los siguientes: `STRING`, `VARCHAR`, o `CHAR`. Si bien Hive permite utilizar cualquier tipo de datos primitivo como clave de asignación, los [símbolos de Amazon Ion](#) deben ser de tipo cadena.
- Tipo de unión: Athena no admite actualmente el [tipo de unión](#) de Hive.
- Tipo de datos double: Amazon Ion no admite actualmente el tipo de datos `double`.

## Temas

- [Uso de CREATE TABLE para crear tablas de Amazon Ion](#)
- [Uso de CTAS e INSERT INTO para crear tablas de Amazon Ion](#)
- [Uso de las propiedades del SerDe de Amazon Ion](#)
- [Uso de extractores de rutas](#)

## Uso de CREATE TABLE para crear tablas de Amazon Ion

Para crear una tabla en Athena a partir de los datos almacenados en formato Amazon Ion, puede utilizar una de las siguientes técnicas en una instrucción CREATE TABLE:

- Especifique STORED AS ION. En este uso, no es necesario que especifique el SerDe de Amazon Ion Hive de forma explícita. Esta elección es la opción más sencilla.
- Especifique las rutas de clases de Amazon Ion en los campos ROW FORMAT SERDE, INPUTFORMAT, y OUTPUTFORMAT.

También puede utilizar las instrucciones CREATE TABLE AS SELECT (CTAS) para crear tablas de Amazon Ion en Athena. Para obtener más información, consulte [Uso de CTAS e INSERT INTO para crear tablas de Amazon Ion](#).

### Especificación de STORED AS ION

La próxima instrucción CREATE TABLE de ejemplo utiliza STORED AS ION antes de la cláusula LOCATION para crear una tabla basada en datos de vuelos en formato Amazon Ion. La cláusula LOCATION especifica el bucket o la carpeta donde se encuentran los archivos de entrada en formato Ion. Se analizan todos los archivos de la ubicación especificada.

```
CREATE EXTERNAL TABLE flights_ion (  
  yr INT,  
  quarter INT,  
  month INT,  
  dayofmonth INT,  
  dayofweek INT,  
  flightdate STRING,  
  uniquecarrier STRING,  
  airlineid INT,  
)  
STORED AS ION  
LOCATION 's3://DOC-EXAMPLE-BUCKET/'
```

### Especificación de las rutas de clases de Amazon Ion

En lugar de utilizar la sintaxis STORED AS ION, puede especificar de forma explícita los valores de la ruta de clase de Ion para las cláusulas ROW FORMAT SERDE, INPUTFORMAT, y OUTPUTFORMAT de la siguiente manera.

Parámetro	Ruta de clase de Ion
ROW FORMAT SERDE	'com.amazon.ionhiveserde.IonHiveSerDe'
STORED AS INPUTFORMAT	'com.amazon.ionhiveserde.formats.IonInputFormat'
OUTPUTFORMAT	'com.amazon.ionhiveserde.formats.IonOutputFormat'

La siguiente consulta DDL utiliza esta técnica para crear la misma tabla externa que en el ejemplo anterior.

```
CREATE EXTERNAL TABLE flights_ion (
  yr INT,
  quarter INT,
  month INT,
  dayofmonth INT,
  dayofweek INT,
  flightdate STRING,
  uniquecarrier STRING,
  airlineid INT,
)
ROW FORMAT SERDE
  'com.amazon.ionhiveserde.IonHiveSerDe'
STORED AS INPUTFORMAT
  'com.amazon.ionhiveserde.formats.IonInputFormat'
OUTPUTFORMAT
  'com.amazon.ionhiveserde.formats.IonOutputFormat'
LOCATION 's3://DOC-EXAMPLE-BUCKET/'
```

Para obtener información sobre las propiedades del SerDe para instrucciones CREATE TABLE en Athena, consulte [Uso de las propiedades del SerDe de Amazon Ion](#).

### Uso de CTAS e INSERT INTO para crear tablas de Amazon Ion

Puede utilizar las instrucciones CREATE TABLE AS SELECT (CTAS) y INSERT INTO para copiar o insertar los datos de una tabla en una tabla nueva en formato de Amazon Ion en Athena.

En una consulta CTAS, especifique `format=' ION '` en la cláusula `WITH`, como en el siguiente ejemplo.

```
CREATE TABLE new_table
WITH (format=' ION ')
AS SELECT * from existing_table
```

Athena serializa de forma predeterminada los resultados de Amazon Ion en [formato binario Ion](#), pero también se puede utilizar el formato de texto. Para utilizar el formato de texto, especifique `ion_encoding = 'TEXT'` en la cláusula `WITH` de la CTAS, como en el siguiente ejemplo.

```
CREATE TABLE new_table
WITH (format=' ION ', ion_encoding = 'TEXT')
AS SELECT * from existing_table
```

Para obtener más información sobre las propiedades específicas de Amazon Ion en la cláusula `WITH` de la CTAS, consulte la siguiente sección.

### CTAS WITH clause Amazon Ion properties (Propiedades de Amazon Ion para la cláusula WITH de la CTAS)

En una consulta CTAS, puede utilizar la cláusula `WITH` para especificar el formato de Amazon Ion y, opcionalmente, especificar la codificación o el algoritmo de compresión de escritura de Amazon Ion que se utilizará.

#### format

Puede especificar la palabra clave `ION` como opción de formato en la cláusula `WITH` de una consulta CTAS. Cuando hace esto, la tabla que crea utiliza el formato que especifica para `IonInputFormat` para las lecturas, y serializa los datos en el formato que especifica para `IonOutputFormat`.

En el siguiente ejemplo, se especifica que la consulta CTAS utiliza el formato de Amazon Ion.

```
WITH (format=' ION ')
```

#### ion\_encoding

Opcional

Valor predeterminado: BINARY

Valores: BINARY, TEXT

Especifica si los datos se serializan en formato binario de Amazon Ion o en formato de texto de Amazon Ion. En el siguiente ejemplo, se especifica el formato de texto de Amazon Ion.

```
WITH (format='ION', ion_encoding='TEXT')
```

write\_compression

Opcional

Valor predeterminado: GZIP

Valores: GZIP, ZSTD, BZIP2, SNAPPY, NONE

Especifica el algoritmo de compresión que se utilizará para comprimir los archivos de salida.

En el siguiente ejemplo, se especifica que la consulta CTAS escribe su salida en formato de Amazon Ion mediante el algoritmo de compresión [Zstandard](#).

```
WITH (format='ION', write_compression = 'ZSTD')
```

Para obtener información sobre el uso de la compresión en Athena, consulte [Compatibilidad con la compresión de Athena](#).

Para obtener información sobre otras propiedades de CTAS en Athena, consulte [Propiedades de la tabla CTAS](#).

Uso de las propiedades del SerDe de Amazon Ion

Este tema contiene información sobre las propiedades del SerDe para instrucciones CREATE TABLE en Athena. Si desea obtener más información y ejemplos del uso de las propiedades del SerDe de Amazon Ion, consulte [SerDe properties](#) (Propiedades del SerDe) en la documentación de Amazon Ion Hive SerDe en [GitHub](#).

Especificación de las propiedades del SerDe de Amazon Ion

Para especificar las propiedades del SerDe de Amazon Ion Hive en la instrucción CREATE TABLE, utilice la cláusula WITH SERDEPROPERTIES. Dado que WITH SERDEPROPERTIES es un subcampo



de la cláusula `ROW FORMAT SERDE`, primero debe especificar `ROW FORMAT SERDE` y la ruta de clase del SerDe de Amazon Ion Hive, como se muestra en la siguiente sintaxis.

```
...
ROW FORMAT SERDE
  'com.amazon.ionhiveserde.IonHiveSerDe'
WITH SERDEPROPERTIES (
  'property' = 'value',
  'property' = 'value',
  ...
)
```

Tenga en cuenta que si bien la cláusula `ROW FORMAT SERDE` es necesaria si desea utilizar `WITH SERDEPROPERTIES`, puede utilizar `STORED AS ION` o las sintaxis más largas `INPUTFORMAT` y `OUTPUTFORMAT` para especificar el formato Amazon Ion.

### Propiedades del SerDe de Amazon Ion

A continuación, se presentan las propiedades del SerDe de Amazon Ion que se pueden utilizar en instrucciones `CREATE TABLE` en Athena.

#### ion.encoding

Opcional

Valor predeterminado: `BINARY`

Valores: `BINARY`, `TEXT`

Esta propiedad declara si los valores nuevos agregados se serializan como formato [binario Amazon Ion](#) o formato de texto Amazon Ion.

En la siguiente propiedad del SerDe de ejemplo se especifica el formato de texto Amazon Ion.

```
'ion.encoding' = 'TEXT'
```

#### ion.fail\_on\_overflow

Opcional

Valor predeterminado: `true`

Valores: `true`, `false`

Amazon Ion permite tipos numéricos arbitrariamente grandes, mientras que Hive no lo permite. El SerDe falla de forma predeterminada si el valor de Amazon Ion no se ajusta a la columna de Hive. Sin embargo, usted puede utilizar la opción de configuración `fail_on_overflow` para permitir que el valor se desborde en lugar de fallar.

Esta propiedad se puede configurar a nivel de tabla o de columna. Para configurarla a nivel de tabla, especifique `ion.fail_on_overflow` como en el siguiente ejemplo. Esto configura el comportamiento predeterminado de todas las columnas.

```
'ion.fail_on_overflow' = 'true'
```

Para controlar una columna específica, especifique el nombre de la columna entre `ion` y `fail_on_overflow`, delimitado por puntos, como en el siguiente ejemplo.

```
'ion.<column>.fail_on_overflow' = 'false'
```

## `ion.path_extractor.case_sensitive`

Opcional

Valor predeterminado: `false`

Valores: `true`, `false`

Determina si se debe hacer distinción entre mayúsculas y minúsculas en los nombres de los campos de Amazon Ion. Cuando es `false`, el SerDe ignora el análisis de mayúsculas y minúsculas de los nombres de los campos de Amazon Ion.

Por ejemplo, supongamos que usted tiene un esquema de tabla de Hive que define un campo `alias` en minúsculas y un documento de Amazon Ion con un campo `alias` y un campo `ALIAS`, como en el siguiente ejemplo.

```
-- Hive Table Schema
alias: STRING

-- Amazon Ion Document
{ 'ALIAS': 'value1' }
{ 'alias': 'value2' }
```

En el siguiente ejemplo, se muestran las propiedades del SerDe y la tabla resultante que se extrae cuando la distinción entre mayúsculas y minúsculas se establece en `false`:

```
-- Serde properties
'ion.alias.path_extractor' = '(alias)'
'ion.path_extractor.case_sensitive' = 'false'

--Extracted Table
| alias      |
|-----|
| "value1"  |
| "value2"  |
```

En el siguiente ejemplo, se muestran las propiedades del SerDe y la tabla resultante que se extrae cuando la distinción entre mayúsculas y minúsculas se establece en `true`:

```
-- Serde properties
'ion.alias.path_extractor' = '(alias)'
'ion.path_extractor.case_sensitive' = 'true'

--Extracted Table
| alias      |
|-----|
| "value2"  |
```

En el segundo caso, `value1` para el campo `ALIAS` se ignora cuando la distinción entre mayúsculas y minúsculas se establece en `true` y el extractor de rutas se especifica como `alias`.

ion.<column>.path\_extractor

Opcional

Predeterminado: N/A

Valores: cadena con ruta de búsqueda

Crea un extractor de rutas con la ruta de búsqueda especificada para la columna determinada. Los extractores de rutas asignan campos de Amazon Ion a las columnas de Hive. Si no se especifica ningún extractor de rutas, Athena crea de forma dinámica extractores de rutas en tiempo de ejecución en función de los nombres de las columnas.

En el siguiente extractor de rutas de ejemplo se asigna el `example_ion_field` a la `example_hive_column`.

```
'ion.example_hive_column.path_extractor' = '(example_ion_field)'
```

Si desea obtener más información sobre los extractores de rutas y las rutas de búsqueda, consulte [Uso de extractores de rutas](#).

`ion.timestamp.serialization_offset`

Opcional

Valor predeterminado: 'Z'

Valores: OFFSET, donde OFFSET está representado como `<signal>hh:mm`. Valores de ejemplo: `01:00`, `+01:00`, `-09:30`, `Z` (UTC, igual que `00:00`)

A diferencia de las [marcas temporales](#) de Apache Hive, que no tienen zona horaria integrada y se almacenan como una diferencia horaria de la fecha de inicio de UNIX, las marcas temporales de Amazon Ion sí tienen una diferencia horaria. Utilice esta propiedad para especificar la diferencia horaria cuando serialice en Amazon Ion.

En el ejemplo siguiente, se agrega una diferencia horaria de una hora.

```
'ion.timestamp.serialization_offset' = '+01:00'
```

`ion.serialize_null`

Opcional

Valor predeterminado: OMIT

Valores: OMIT, UNTYPED, TYPED

El SerDe de Amazon Ion se puede configurar para que serialice u omita columnas que tengan valores nulos. Puede elegir escribir valores nulos con tipos sólidos (TYPED) o valores nulos sin tipo alguno (UNTYPED). Los valores nulos con tipos sólidos se determinan en función de la asignación de tipos predeterminada de Amazon Ion para Hive.

En el siguiente ejemplo, se especifican valores nulos con tipos sólidos.

```
'ion.serialize_null'='TYPED'
```

## ion.ignore\_malformed

Opcional

Valor predeterminado: `false`

Valores: `true`, `false`

Cuando es `true`, se ignoran las entradas con formato incorrecto o se ignora el archivo completo si el SerDe no puede leerlo. Para obtener más información, consulte [Ignore malformed](#) (Ignorar elementos con formato incorrecto) en la documentación de GitHub.

## ion.<column>.serialize\_as

Opcional

Predeterminado: tipo predeterminado para la columna.

Valores: cadena que contiene el tipo de Amazon Ion

Determina el tipo de datos de Amazon Ion en el que se serializa un valor. Dado que los tipos de Amazon Ion y Hive no siempre tienen una asignación directa, algunos tipos de Hive tienen varios tipos de datos válidos para la serialización. Para serializar los datos como un tipo de datos no predeterminado, utilice esta propiedad. Si desea obtener más información sobre la asignación de tipos, consulte la página [Type mapping](#) (Asignación de tipos) de Amazon Ion en GitHub.

Las columnas binarias de Hive se serializan de forma predeterminada como blobs de Amazon Ion, pero también se pueden serializar como un [clob de Amazon Ion](#), es decir, un objeto grande de caracteres. En el siguiente ejemplo, se serializa la columna `example_hive_binary_column` como un clob.

```
'ion.example_hive_binary_column.serialize_as' = 'clob'
```

## Uso de extractores de rutas

Amazon Ion es un formato de archivo de estilo de documento. Por otro lado, Apache Hive es un formato de columnas plano. Puede utilizar las propiedades especiales del SerDe de Amazon Ion denominadas `path extractors` para asignar entre los dos formatos. Los extractores de rutas

aplanan el formato Amazon Ion jerárquico, asignan valores de Amazon Ion a columnas de Hive y se pueden utilizar para cambiar los nombres de los campos.

Athena puede generar los extractores por usted; sin embargo, usted también puede definir sus propios extractores si es necesario.

### Extractores de rutas generados

Athena busca de forma predeterminada valores de Amazon Ion de nivel superior que coincidan con los nombres de las columnas de Hive y crea extractores de rutas en tiempo de ejecución en función de estos valores coincidentes. Si el formato de datos de Amazon Ion coincide con el esquema de tabla de Hive, Athena genera de forma dinámica los extractores por usted y usted no necesita agregar ningún extractor de ruta adicional. Estos extractores de rutas predeterminados no se almacenan en los metadatos de la tabla.

En el siguiente ejemplo, se muestra cómo Athena genera extractores en función del nombre de la columna.

```
-- Example Amazon Ion Document
{
  identification: {
    name: "John Smith",
    driver_license: "XXXX"
  },

  alias: "Johnny"
}

-- Example DDL
CREATE EXTERNAL TABLE example_schema2 (
  identification MAP<STRING, STRING>,
  alias STRING
)
STORED AS ION
LOCATION 's3://DOC-EXAMPLE-BUCKET/path_extraction1/'
```

Athena genera los siguientes extractores de ejemplo. El primero extrae el campo `identification` a la columna `identification` y el segundo extrae el campo `alias` a la columna `alias`.

```
'ion.identification.path_extractor' = '(identification)'
'ion.alias.path_extractor' = '(alias)'
```

En el siguiente ejemplo, se muestra la tabla que se extrajo.

```
|          identification          | alias |
|-----|-----|
| [{"name", "driver_license"}, ["John Smith", "XXXX"]] | "Johnny" |
```

### Especificación de sus propios extractores de rutas

Si los campos de Amazon Ion no se asignan de forma perfecta a las columnas de Hive, puede especificar sus propios extractores de ruta. En la cláusula WITH SERDEPROPERTIES de la instrucción CREATE TABLE, utilice la siguiente sintaxis.

```
WITH SERDEPROPERTIES (
  "ion.path_extractor.case_sensitive" = "<Boolean>",
  "ion.<column_name>.path_extractor" = "<path_extractor_expression>"
)
```

#### Note

Los extractores de rutas distinguen entre mayúsculas y minúsculas de forma predeterminada. Para anular esta configuración, configure la propiedad del SerDe [ion.path\\_extractor.case\\_sensitive](#) en true.

### Uso de rutas de búsqueda en extractores de rutas

La sintaxis de la propiedad del SerDe para el extractor de ruta contiene *<path\_extractor\_expression>*:

```
"ion.<column_name>.path_extractor" = "<path_extractor_expression>"
```

Puede utilizar *<path\_extractor\_expression>* para especificar una ruta de búsqueda que analice el documento de Amazon Ion y encuentre datos coincidentes. La ruta de búsqueda está entre paréntesis y puede contener uno o más de los siguientes componentes separados por espacios.

- Comodín: coincide con todos los valores.
- Índice: coincide con el valor del índice numérico especificado. Los índices están basados en cero.
- Texto: coincide con todos los valores cuya coincidencia de nombres de campo son equivalentes al texto especificado.

- **Anotaciones:** coincide con los valores especificados por un componente de ruta encapsulado que tiene las anotaciones especificadas.

En el siguiente ejemplo, se muestra un documento de Amazon Ion y algunas rutas de búsqueda de ejemplo.

```
-- Amazon Ion document
{
  foo: ["foo1", "foo2"] ,
  bar: "myBarValue",
  bar: A::"annotatedValue"
}

-- Example search paths
(foo 0)      # matches "foo1"
(1)         # matches "myBarValue"
(*)         # matches ["foo1", "foo2"], "myBarValue" and A::"annotatedValue"
()          # matches {foo: ["foo1", "foo2"] , bar: "myBarValue", bar:
A::"annotatedValue"}
(bar)       # matches "myBarValue" and A::"annotatedValue"
(A::bar)    # matches A::"annotatedValue"
```

## Ejemplos de extractores

### Aplanamiento y cambio de los nombres de los campos

En el siguiente ejemplo, se muestra un conjunto de rutas de búsqueda que aplanan y cambian los nombres de los campos. En el ejemplo, se utilizan las rutas de búsqueda para hacer lo siguiente:

- asignar la columna `nickname` al campo `alias` y
- asignar la columna `name` al subcampo `name` ubicado en la estructura `identification`.

A continuación, se presenta el documento de Amazon Ion de ejemplo.

```
-- Example Amazon Ion Document
{
  identification: {
    name: "John Smith",
    driver_license: "XXXX"
  },
}
```



```
alias: "Johnny"
}
```

A continuación, se muestra la instrucción CREATE TABLE de ejemplo que define los extractores de rutas.

```
-- Example DDL Query
CREATE EXTERNAL TABLE example_schema2 (
  name STRING,
  nickname STRING
)
ROW FORMAT SERDE
  'com.amazon.ionhiveserde.IonHiveSerDe'
WITH SERDEPROPERTIES (
  'ion.nickname.path_extractor' = '(alias)',
  'ion.name.path_extractor' = '(identification name)'
)
STORED AS ION
LOCATION 's3://DOC-EXAMPLE-BUCKET/path_extraction2/'
```

En el siguiente ejemplo, se muestran los datos extraídos.

```
-- Extracted Table
| name          | nickname      |
|-----|-----|
| "John Smith" | "Johnny"     |
```

Para obtener más información sobre las rutas de búsqueda y ejemplos de rutas de búsqueda adicionales, consulte [Extracción de rutas Ion Java](#) en GitHub.

### Extracción de datos de vuelo en formato de texto

En el siguiente ejemplo, la consulta CREATE TABLE utiliza WITH SERDEPROPERTIES para agregar extractores de rutas con el objetivo de extraer datos de vuelo y especificar la codificación de salida como texto de Amazon Ion. En el ejemplo se utiliza la sintaxis STORED AS ION.

```
CREATE EXTERNAL TABLE flights_ion (
  yr INT,
  quarter INT,
  month INT,
  dayofmonth INT,
  dayofweek INT,
```

```
    flightdate STRING,  
    uniquecarrier STRING,  
    airlineid INT,  
  )  
ROW FORMAT SERDE  
  'com.amazon.ionhiveserde.IonHiveSerDe'  
WITH SERDEPROPERTIES (  
  'ion.encoding' = 'TEXT',  
  'ion.yr.path_extractor'='(year)',  
  'ion.quarter.path_extractor'='(results quarter)',  
  'ion.month.path_extractor'='(date month)')  
STORED AS ION  
LOCATION 's3://DOC-EXAMPLE-BUCKET/'
```

## El SerDe de Avro

Nombre del SerDe

### [El SerDe de Avro](#)

Nombre de la biblioteca

### [org.apache.hadoop.hive.serde2.avro.AvroSerDe](#)

## Ejemplos

Por motivos de seguridad, Athena no es compatible con el uso de `avro.schema.url` para especificar el esquema de la tabla. Utilice `avro.schema.literal`. Para extraer el esquema de datos en formato Avro, puede utilizar el `avro-tools-<version>.jar` de Apache con el parámetro `getschema`. Esto devuelve un esquema que puede utilizar en su instrucción `WITH SERDEPROPERTIES`. Por ejemplo:

```
java -jar avro-tools-1.8.2.jar getschema my_data.avro
```

El archivo `avro-tools-<version>.jar` se encuentra en el subdirectorio `java` de la versión de Avro que tiene instalada. Para descargar Avro, consulte la página que contiene las [versiones de Apache Avro](#). Para descargar Apache Avro Tools directamente, consulte la página del [repositorio Maven de Apache Avro Tools](#).

Una vez obtenido el esquema, utilice una instrucción `CREATE TABLE` para crear una tabla de Athena basada en los datos de Avro subyacentes almacenados en Amazon S3. Para especificar el SerDe de Avro, utilice `ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.avro.AvroSerDe'`. Como se demuestra en el ejemplo siguiente, debe especificar el esquema mediante la cláusula

WITH SERDEPROPERTIES además de especificar los nombres de columna y los tipos de datos correspondientes para la tabla.

### Note

Reemplace *myregion* en `s3://athena-examples-myregion/path/to/data/` por el identificador de región donde se ejecuta Athena, por ejemplo, `s3://athena-examples-us-west-1/path/to/data/`.

```
CREATE EXTERNAL TABLE flights_avro_example (
  yr INT,
  flightdate STRING,
  uniquecarrier STRING,
  airlineid INT,
  carrier STRING,
  flightnum STRING,
  origin STRING,
  dest STRING,
  depdelay INT,
  carrierdelay INT,
  weatherdelay INT
)
PARTITIONED BY (year STRING)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.avro.AvroSerDe'
WITH SERDEPROPERTIES ('avro.schema.literal'='
{
  "type" : "record",
  "name" : "flights_avro_subset",
  "namespace" : "default",
  "fields" : [ {
    "name" : "yr",
    "type" : [ "null", "int" ],
    "default" : null
  }, {
    "name" : "flightdate",
    "type" : [ "null", "string" ],
    "default" : null
  }, {
    "name" : "uniquecarrier",
    "type" : [ "null", "string" ],
    "default" : null
```

```

}, {
  "name" : "airlineid",
  "type" : [ "null", "int" ],
  "default" : null
}, {
  "name" : "carrier",
  "type" : [ "null", "string" ],
  "default" : null
}, {
  "name" : "flightnum",
  "type" : [ "null", "string" ],
  "default" : null
}, {
  "name" : "origin",
  "type" : [ "null", "string" ],
  "default" : null
}, {
  "name" : "dest",
  "type" : [ "null", "string" ],
  "default" : null
}, {
  "name" : "depdelay",
  "type" : [ "null", "int" ],
  "default" : null
}, {
  "name" : "carrierdelay",
  "type" : [ "null", "int" ],
  "default" : null
}, {
  "name" : "weatherdelay",
  "type" : [ "null", "int" ],
  "default" : null
} ]
}
')
STORED AS AVRO
LOCATION 's3://athena-examples-myregion/flight/avro/';

```

Ejecute la instrucción `MSCK REPAIR TABLE` en la tabla para actualizar los metadatos de partición.

```
MSCK REPAIR TABLE flights_avro_example;
```

Consulte las 10 primeras ciudades de salida en función del número total de salidas.

```
SELECT origin, count(*) AS total_departures
FROM flights_avro_example
WHERE year >= '2000'
GROUP BY origin
ORDER BY total_departures DESC
LIMIT 10;
```

### Note

Los datos de la tabla de vuelos provienen de [Vuelos](#) proporcionados por la [Oficina de Estadísticas del Transporte](#) del Departamento de Transporte de los Estados Unidos. Desaturado del original.

## El SerDe de Grok

El SerDe de Grok Logstash es una biblioteca con un conjunto de patrones especializados para la deserialización de archivos de texto sin estructurar, normalmente registros. Cada patrón de Grok es una expresión regular con nombre. Puede identificar y volver a utilizar estos patrones de deserialización según sea necesario. De ese modo resulta más sencillo utilizar Grok en comparación con el uso de expresiones regulares. Grok proporciona un conjunto de [patrones predefinidos](#). También puede crear patrones personalizados.

Para especificar el SerDe de Grok al crear una tabla en Athena, utilice la cláusula `ROW FORMAT SERDE 'com.amazonaws.glue.serde.GrokSerDe'` seguida de la cláusula `WITH SERDEPROPERTIES` que especifica los patrones que concuerdan con los datos, donde:

- La expresión `input.format` define los patrones que concuerdan con los datos. Es obligatorio.
- La expresión `input.grokCustomPatterns` define un patrón personalizado con nombre, que posteriormente puede utilizar dentro de la expresión `input.format`. Es opcional. Para incluir varias entradas de patrones en la expresión `input.grokCustomPatterns`, sepárelas con el carácter de escape para nueva línea (`\n`), de esta forma:  
`'input.grokCustomPatterns'='INSIDE_QS ([^\"]*)\nINSIDE_BRACKETS ([^\n\]]*)'`.
- Las cláusulas `STORED AS INPUTFORMAT` y `OUTPUTFORMAT` son obligatorias.
- La cláusula `LOCATION` especifica un bucket de Amazon S3 que puede contener varios objetos de datos. Todos los objetos de datos del bucket se deserializan para crear la tabla.

## Ejemplos

Estos ejemplos se basan en la lista de patrones de Grok predefinidos. Consulte los [patrones predefinidos](#).

### Ejemplo 1

En este ejemplo se utilizan datos de origen de entradas de registro de correo Postfix guardadas en `s3://DOC-EXAMPLE-BUCKET/groksample/`.

```
Feb  9 07:15:00 m4eastmail postfix/smtpd[19305]: B88C4120838: connect from
unknown[192.168.55.4]
Feb  9 07:15:00 m4eastmail postfix/smtpd[20444]: B58C4330038:
client=unknown[192.168.55.4]
Feb  9 07:15:03 m4eastmail postfix/cleanup[22835]: BDC22A77854: message-
id=<31221401257553.5004389LCBF@m4eastmail.example.com>
```

La siguiente instrucción crea una tabla en Athena llamada `mygroktable` a partir de los datos de origen, usando un patrón personalizado y los patrones predefinidos que especifique:

```
CREATE EXTERNAL TABLE `mygroktable` (
  syslogbase string,
  queue_id string,
  syslog_message string
)
ROW FORMAT SERDE
  'com.amazonaws.glue.serde.GrokSerDe'
WITH SERDEPROPERTIES (
  'input.grokCustomPatterns' = 'POSTFIX_QUEUEID [0-9A-F]{7,12}',
  'input.format' = '%{SYSLOGBASE} %{POSTFIX_QUEUEID:queue_id}:
%{GREEDYDATA:syslog_message}'
)
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET/groksample/';
```

Comience con un patrón sencillo, como `%{NOTSPACE:column}`, para asignar primero las columnas y después especialice las columnas si es necesario.

## Ejemplo 2

En el siguiente ejemplo se crea una consulta para registros Log4j. Los registros de ejemplo tienen las entradas en este formato:

```
2017-09-12 12:10:34,972 INFO - processType=AZ, processId=ABCDEFG614B6F5E49,
  status=RUN,
threadId=123:amqListenerContainerPool23P:AJ|ABCDE9614B6F5E49||
2017-09-12T12:10:11.172-0700],
executionTime=7290, tenantId=12456, userId=123123f8535f8d76015374e7a1d87c3c,
  shard=testapp1,
jobId=12312345e5e7df0015e777fb2e03f3c, messageType=REAL_TIME_SYNC,
action=receive, hostname=1.abc.def.com
```

Para consultar estos datos de registro:

- Añada el patrón de Grok al `input.format` de cada columna. Por ejemplo, para `timestamp`, añade `%{TIMESTAMP_ISO8601:timestamp}`. Para `loglevel`, añade `%{LOGLEVEL:loglevel}`.
- Asegúrese de que el patrón de `input.format` coincida exactamente con el formato del log mapeando los guiones (-) y las comas que separan las entregas en el formato de log.

```
CREATE EXTERNAL TABLE bltest (
  timestamp STRING,
  loglevel STRING,
  processtype STRING,
  processid STRING,
  status STRING,
  threadid STRING,
  executiontime INT,
  tenantid INT,
  userid STRING,
  shard STRING,
  jobid STRING,
  messagetype STRING,
  action STRING,
  hostname STRING
)
ROW FORMAT SERDE 'com.amazonaws.glue.serde.GrokSerDe'
WITH SERDEPROPERTIES (
  'input.grokCustomPatterns' = 'C_ACTION receive|send',
```

```
"input.format" = "%{TIMESTAMP_ISO8601:timestamp} %{LOGLEVEL:loglevel} - procesType=
%{NOTSPACE:processtype}, processId=%{NOTSPACE:processid}, status=%{NOTSPACE:status},
threadId=%{NOTSPACE:threadid}, executionTime=%{POSINT:executiontime}, tenantId=
%{POSINT:tenantid}, userId=%{NOTSPACE:userid}, shard=%{NOTSPACE:shard}, jobId=
%{NOTSPACE:jobid}, messageType=%{NOTSPACE:messagetype}, action=%{C_ACTION:action},
hostname=%{HOST:hostname}"
) STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://DOC-EXAMPLE-BUCKET/samples/';
```

### Ejemplo 3

El ejemplo siguiente de consulta de registros de Amazon S3 muestra la expresión

'input.grokCustomPatterns' con dos entradas de patrón separadas por el carácter de escape de nueva línea (\n), como indica este fragmento de código de la consulta de ejemplo: 'input.grokCustomPatterns'='INSIDE\_QS ([^\"]\*)\nINSIDE\_BRACKETS ([^\n\]]\*)').

```
CREATE EXTERNAL TABLE `s3_access_auto_raw_02` (
  `bucket_owner` string COMMENT 'from deserializer',
  `bucket` string COMMENT 'from deserializer',
  `time` string COMMENT 'from deserializer',
  `remote_ip` string COMMENT 'from deserializer',
  `requester` string COMMENT 'from deserializer',
  `request_id` string COMMENT 'from deserializer',
  `operation` string COMMENT 'from deserializer',
  `key` string COMMENT 'from deserializer',
  `request_uri` string COMMENT 'from deserializer',
  `http_status` string COMMENT 'from deserializer',
  `error_code` string COMMENT 'from deserializer',
  `bytes_sent` string COMMENT 'from deserializer',
  `object_size` string COMMENT 'from deserializer',
  `total_time` string COMMENT 'from deserializer',
  `turnaround_time` string COMMENT 'from deserializer',
  `referrer` string COMMENT 'from deserializer',
  `user_agent` string COMMENT 'from deserializer',
  `version_id` string COMMENT 'from deserializer')
ROW FORMAT SERDE
  'com.amazonaws.glue.serde.GrokSerDe'
WITH SERDEPROPERTIES (
  'input.format'='%{NOTSPACE:bucket_owner} %{NOTSPACE:bucket} \
\[%{INSIDE_BRACKETS:time}\]\] %{NOTSPACE:remote_ip} %{NOTSPACE:requester}
```



```

%{NOTSPACE:request_id} %{NOTSPACE:operation} %{NOTSPACE:key} \"?
%{INSIDE_QS:request_uri}\"? %{NOTSPACE:http_status} %{NOTSPACE:error_code}
%{NOTSPACE:bytes_sent} %{NOTSPACE:object_size} %{NOTSPACE:total_time}
%{NOTSPACE:turnaround_time} \"?%{INSIDE_QS:referrer}\"? \"?%{INSIDE_QS:user_agent}\"?
%{NOTSPACE:version_id}',
  'input.grokCustomPatterns'='INSIDE_QS ([^\""]*)\nINSIDE_BRACKETS ([^\\\"]*)')
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET'

```

## Bibliotecas de SerDe JSON

En Athena, puede utilizar dos bibliotecas SerDe para deserializar datos JSON. La deserialización convierte los datos JSON para que se puedan serializar (escribir) en un formato diferente como Parquet u ORC.

- [El SerDe JSON de Hive](#) nativo
- Con la [El SerDe JSON de OpenX](#)
- Con la [SerDe de Amazon Ion Hive](#)

### Note

Las bibliotecas de Hive y OpenX esperan que los datos JSON estén en una sola línea (sin formato), con registros separados por un carácter de nueva línea. Sin embargo, Amazon Ion Hive SerDe no tiene ese requisito y puede utilizarse como alternativa, ya que el formato de datos Ion es un superconjunto de JSON.

Nombres de las bibliotecas

Utilice una de las siguientes:

[org.apache.hive.hcatalog.data.JsonSerDe](http://org.apache.hive.hcatalog.data.JsonSerDe)

[org.openx.data.jsonserde.JsonSerDe](http://org.openx.data.jsonserde.JsonSerDe)

[com.amazon.ionhiveserde.IonHiveSerDe](http://com.amazon.ionhiveserde.IonHiveSerDe)

## El SerDe JSON de Hive

SerDe JSON de Hive se usa normalmente para procesar datos JSON como eventos. Estos eventos se representan como cadenas de una sola línea de texto con codificación JSON separados por una línea nueva. SerDe JSON de Hive no permite claves duplicadas en map ni nombres de claves `struct`.

### Note

El SerDe espera que cada documento JSON esté en una sola línea de texto sin caracteres de terminación de línea que separen los campos del registro. Si el texto JSON está en formato de impresión, puede recibir un mensaje de error como `HIVE_CURSOR_ERROR: la fila no es un objeto JSON válido` o `HIVE_CURSOR_ERROR: JsonParseException: fin de entrada inesperado: marcador de cierre esperado para OBJECT cuando intenta consultar la tabla después de crearla`. Para obtener más información, consulte los [Archivos de datos JSON](#) en la documentación de OpenX SerDE en GitHub.

La instrucción DDL de ejemplo que se muestra a continuación, utiliza SerDe JSON de Hive para crear una tabla en función de los datos de publicidad en línea de ejemplo. En la cláusula `LOCATION`, reemplace *myregion* en `s3://DOC-EXAMPLE-BUCKET.elasticmapreduce/samples/hive-ads/tables/impressions` por el identificador de región donde se ejecuta Athena (por ejemplo, `s3://us-west-2.elasticmapreduce/samples/hive-ads/tables/impressions`).

```
CREATE EXTERNAL TABLE impressions (  
    requestbegintime string,  
    adid string,  
    impressionid string,  
    referrer string,  
    useragent string,  
    usercookie string,  
    ip string,  
    number string,  
    processid string,  
    browsercookie string,  
    requestendtime string,  
    timers struct  
    <  
        modellookup:string,  
        requesttime:string
```

```

        >,
        threadid string,
        hostname string,
        sessionid string
    )
PARTITIONED BY (dt string)
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
LOCATION 's3://DOC-EXAMPLE-BUCKET.elasticmapreduce/samples/hive-ads/tables/
impressions';

```

## Especificación de formatos de marca de tiempo con el SerDe JSON de Hive

Para analizar los valores de marca de tiempo de una cadena, se puede agregar el subcampo `WITH SERDEPROPERTIES` a la cláusula `ROW FORMAT SERDE` y utilizarla para especificar el parámetro `timestamp.formats`. En el parámetro, se debe especificar una lista separada por comas de uno o más patrones de marca de tiempo, como en el siguiente ejemplo:

```

...
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
WITH SERDEPROPERTIES ("timestamp.formats"="yyyy-MM-dd'T'HH:mm:ss.SSS'Z',yyyy-MM-
dd'T'HH:mm:ss")
...

```

Para obtener más información, consulte [Timestamps](#) en la documentación de Apache Hive.

## Carga de la tabla para consulta

Después de crear la tabla, ejecute [MSCK REPAIR TABLE](#) para cargar la tabla y hacerla consultable desde Athena:

```
MSCK REPAIR TABLE impressions
```

## Consulta de registros de CloudTrail

Puede usar los SerDe JSON de Hive para consultar los registros de CloudTrail. Para obtener más información y ejemplos de las instrucciones `CREATE TABLE`, consulte [Consulta de registros de AWS CloudTrail](#).

## El SerDe JSON de OpenX

Al igual que el SerDe JSON de Hive, puede utilizar el JSON de OpenX para procesar datos JSON. Los datos también se representan como cadenas de una sola línea de texto con codificación JSON

separadas por una línea nueva. Al igual que SerDe JSON de Hive, SerDe JSON de OpenX no permite claves duplicadas en map ni nombres de claves `struct`.

### Note

El SerDe espera que cada documento JSON esté en una sola línea de texto sin caracteres de terminación de línea que separen los campos del registro. Si el texto JSON está en formato de impresión, puede recibir un mensaje de error como `HIVE_CURSOR_ERROR: la fila no es un objeto JSON válido` o `HIVE_CURSOR_ERROR: JsonParseException: fin de entrada inesperado: marcador de cierre esperado para OBJECT cuando intenta consultar la tabla después de crearla`. Para obtener más información, consulte los [Archivos de datos JSON](#) en la documentación de OpenX SerDE en GitHub.

## Propiedades opcionales

A diferencia del SerDe JSON de Hive, el SerDe JSON de OpenX también tiene las siguientes propiedades SerDe opcionales que pueden resultar útiles para solucionar incoherencias en los datos.

### `ignore.malformed.json`

Opcional. Cuando se establece en `TRUE`, le permite omitir la sintaxis JSON con formato incorrecto. El valor predeterminado es `FALSE`.

### `dots.in.keys`

Opcional. El valor predeterminado es `FALSE`. Cuando se establece en `TRUE`, permite que el SerDe sustituya los puntos en nombres de claves con guiones bajos. Por ejemplo, si el conjunto de datos de JSON contiene una clave con el nombre `"a.b"`, puede utilizar esta propiedad para definir el nombre de la columna para que sea `"a_b"` en Athena. De forma predeterminada (sin este SerDe), Athena no permite puntos en los nombres de columnas.

### `case.insensitive`

Opcional. El valor predeterminado es `TRUE`. Cuando se establece en `TRUE`, el SerDe convierte todas las columnas en mayúscula a minúscula.

Para utilizar nombres de clave que distinguen mayúsculas de minúsculas en los datos, utilice `WITH SERDEPROPERTIES ("case.insensitive"= FALSE;)`. A continuación, para cada clave que no esté ya en minúsculas, proporcione un mapeo desde el nombre de la columna al nombre de la propiedad utilizando la siguiente sintaxis:

```
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
WITH SERDEPROPERTIES ("case.insensitive" = "FALSE", "mapping.userid" = "userId")
```

Si tiene dos claves como URL y Ur1 que son iguales cuando se escriben en minúsculas, puede producirse un error como el siguiente:

HIVE\_CURSOR\_ERROR: Row is not a valid JSON Object - JSONException: Duplicate key "url"

Para resolver esto, establezca la propiedad `case.insensitive` en `FALSE` y asigne las claves a nombres diferentes, como en el ejemplo siguiente:

```
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
WITH SERDEPROPERTIES ("case.insensitive" = "FALSE", "mapping.url1" = "URL",
"mapping.url2" = "Ur1")
```

## asignación

Opcional. Asigna nombres de columna a las claves JSON que no son idénticas a los nombres de las columnas. El parámetro `mapping` es útil cuando los datos JSON contienen claves que son [palabras clave](#). Por ejemplo, si tiene una clave JSON denominada `timestamp`, utilice la siguiente sintaxis para mapear la clave a una columna denominada `ts`:

```
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
WITH SERDEPROPERTIES ("mapping.ts" = "timestamp")
```

Asignación de nombres de campos anidados con dos puntos a nombres compatibles con Hive

Si tiene un nombre de campo con dos puntos dentro de una estructura `struct`, puede utilizar la propiedad `mapping` para asignar ese campo a un nombre compatible con Hive. Por ejemplo, si las definiciones de tipos de columna contienen `my:struct:field:string`, puede asignar la definición a `my_struct_field:string` incluyendo la siguiente entrada en `WITH SERDEPROPERTIES`:

```
("mapping.my_struct_field" = "my:struct:field")
```

En el siguiente ejemplo se muestra la instrucción `CREATE TABLE` correspondiente.

```
CREATE EXTERNAL TABLE colon_nested_field (
item struct<my_struct_field:string>)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
```

```
WITH SERDEPROPERTIES ("mapping.my_struct_field" = "my:struct:field")
```

### Ejemplo: datos de publicidad

La instrucción DDL de ejemplo que se muestra a continuación, utiliza SerDe JSON de OpenX para crear una tabla en función de los mismos datos de publicidad en línea de ejemplo utilizados en el ejemplo para SerDe JSON de Hive. En la cláusula LOCATION, reemplace *myregion* por el identificador de región donde se ejecuta Athena.

```
CREATE EXTERNAL TABLE impressions (  
    requestbegintime string,  
    adid string,  
    impressionId string,  
    referrer string,  
    useragent string,  
    usercookie string,  
    ip string,  
    number string,  
    processid string,  
    browsercookie string,  
    requestendtime string,  
    timers struct<  
        modellookup:string,  
        requesttime:string>,  
    threadid string,  
    hostname string,  
    sessionid string  
) PARTITIONED BY (dt string)  
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'  
LOCATION 's3://DOC-EXAMPLE-BUCKET.elasticmapreduce/samples/hive-ads/tables/  
impressions';
```

### Ejemplo: deserialización de JSON anidado

Puede usar SerDes JSON para analizar datos codificados en JSON más complejos. Esto requiere el uso de instrucciones CREATE TABLE que utilizan elementos struct y array para representar estructuras anidadas.

En el ejemplo siguiente, se crea una tabla de Athena a partir de datos JSON que tienen estructuras anidadas. Para analizar los datos con codificación JSON en Athena, asegúrese de que cada documento JSON esté en su propia línea, separado por una nueva línea.

En este ejemplo se supone que los datos codificados en JSON tienen la siguiente estructura:

```
{
  "DocId": "AWS",
  "User": {
    "Id": 1234,
    "Username": "bob1234",
    "Name": "Bob",
  "ShippingAddress": {
    "Address1": "123 Main St.",
    "Address2": null,
    "City": "Seattle",
    "State": "WA"
  },
  "Orders": [
    {
      "ItemId": 6789,
      "OrderDate": "11/11/2017"
    },
    {
      "ItemId": 4352,
      "OrderDate": "12/12/2017"
    }
  ]
}
```

La instrucción CREATE TABLE siguiente utiliza [OpenX-JsonSerDe](#) con los tipos de datos de colección struct y array para establecer grupos de objetos. Cada documento JSON se encuentra en su propia línea, separado de los demás por un carácter de nueva línea. Para evitar errores, los datos que se van a consultar no incluyen claves duplicadas en struct ni nombres de clave de mapeo.

```
CREATE external TABLE complex_json (
  docid string,
  `user` struct<
    id:INT,
    username:string,
    name:string,
    shippingaddress:struct<
      address1:string,
      address2:string,
```

```

        city:string,
        state:string
    >,
    orders:array<
        struct<
            itemid:INT,
            orderdate:string
        >
    >
)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION 's3://DOC-EXAMPLE-BUCKET/myjsondata/';

```

## Recursos adicionales de

Para obtener más información sobre cómo trabajar con JSON y JSON anidado en Athena, consulte los siguientes recursos:

- [Create tables in Amazon Athena from nested JSON and mappings using JSONSerDe](#) (Crear tablas en Amazon Athena a partir de JSON anidado y asignaciones por medio de JSONSerDe) (Blog de macrodatos en AWS)
- [Recibo errores cuando intento leer datos JSON en Amazon Athena](#) (artículo del centro de conocimiento de AWS)
- [hive-json-schema](#) (GitHub): herramienta escrita en Java que genera instrucciones CREATE TABLE de documentos JSON de ejemplo. Las instrucciones CREATE TABLE que se generan utilizan SerDe JSON de OpenX.

## LazySimpleSerDe para CSV, TSV y archivos con delimitación personalizada

La especificación de este SerDe es opcional. Este es el SerDe para datos con formato CSV, TSV o con delimitación personalizada que Athena utiliza de forma predeterminada. Este SerDe se utiliza si no especifica ningún SerDe y solo especifica ROW FORMAT DELIMITED. Use este SerDe si los datos no tienen valores entre comillas.

Para obtener documentación de referencia sobre LazySimpleSerDe, consulte la sección [SerDe de Hive](#) de la Guía para desarrolladores de Apache Hive.



## Nombre de la biblioteca

El nombre de la biblioteca de clases de LazySimpleSerDe es `org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe`. Para obtener información sobre la clase LazySimpleSerDe, consulte [LazySimpleSerDe](#) en GitHub.com.

## Ignorar encabezados

Para ignorar los encabezados en los datos cuando defina una tabla, puede utilizar la propiedad de tabla `skip.header.line.count`, como en el siguiente ejemplo.

```
TBLPROPERTIES ("skip.header.line.count"="1")
```

Para ver ejemplos, consulte las instrucciones CREATE TABLE en [Consulta de los registros de flujo de Amazon VPC](#) y [Consultas de registros de Amazon CloudFront](#).

## Ejemplo de CSV

En los siguientes ejemplos se muestra cómo utilizar LazySimpleSerDe para crear una tabla en Athena a partir de datos CSV. Para deserializar archivos delimitados personalizados mediante este SerDe, siga el patrón de los ejemplos, pero utilice la cláusula `FIELDS TERMINATED BY` para especificar un delimitador de un solo carácter diferente. LazySimpleSerDe no admite delimitadores de múltiples caracteres.

### Note

Reemplace *myregion* en `s3://athena-examples-myregion/path/to/data/` por el identificador de región donde se ejecuta Athena, por ejemplo, `s3://athena-examples-us-west-1/path/to/data/`.

Utilice la instrucción CREATE TABLE para crear una tabla de Athena a partir de datos subyacentes en formato CSV almacenados en Amazon S3.

```
CREATE EXTERNAL TABLE flight_delays_csv (  
  yr INT,  
  quarter INT,  
  month INT,  
  dayofmonth INT,  
  dayofweek INT,  
  flightdate STRING,
```

```
uniquecarrier STRING,  
airlineid INT,  
carrier STRING,  
tailnum STRING,  
flightnum STRING,  
originairportid INT,  
originairportseqid INT,  
origincitymarketid INT,  
origin STRING,  
origincityname STRING,  
originstate STRING,  
originstatefips STRING,  
originstatename STRING,  
originwac INT,  
destairportid INT,  
destairportseqid INT,  
destcitymarketid INT,  
dest STRING,  
destcityname STRING,  
deststate STRING,  
deststatefips STRING,  
deststatename STRING,  
destwac INT,  
crsdeptime STRING,  
deptime STRING,  
depdelay INT,  
depdelayminutes INT,  
depdel15 INT,  
departuredelaygroups INT,  
deptimeblk STRING,  
taxiout INT,  
wheelsoff STRING,  
wheelson STRING,  
taxiin INT,  
crsarrrtime INT,  
arrrtime STRING,  
arrrdelay INT,  
arrrdelayminutes INT,  
arrrdel15 INT,  
arrivaldelaygroups INT,  
arrrtimeblk STRING,  
cancelled INT,  
cancellationcode STRING,  
diverted INT,
```

```
crselapsedtime INT,  
actualelapsedtime INT,  
airtime INT,  
flights INT,  
distance INT,  
distancegroup INT,  
carrierdelay INT,  
weatherdelay INT,  
nasdelay INT,  
securitydelay INT,  
lateaircraftdelay INT,  
firstdeptime STRING,  
totaladdgtime INT,  
longestaddgtime INT,  
divairportlandings INT,  
divreacheddest INT,  
divactualelapsedtime INT,  
divarrdelay INT,  
divdistance INT,  
div1airport STRING,  
div1airportid INT,  
div1airportseqid INT,  
div1wheelson STRING,  
div1totalgtime INT,  
div1longestgtime INT,  
div1wheelsoff STRING,  
div1tailnum STRING,  
div2airport STRING,  
div2airportid INT,  
div2airportseqid INT,  
div2wheelson STRING,  
div2totalgtime INT,  
div2longestgtime INT,  
div2wheelsoff STRING,  
div2tailnum STRING,  
div3airport STRING,  
div3airportid INT,  
div3airportseqid INT,  
div3wheelson STRING,  
div3totalgtime INT,  
div3longestgtime INT,  
div3wheelsoff STRING,  
div3tailnum STRING,  
div4airport STRING,
```

```
div4airportid INT,  
div4airportseqid INT,  
div4wheelson STRING,  
div4totalgtime INT,  
div4longestgtime INT,  
div4wheelsoff STRING,  
div4tailnum STRING,  
div5airport STRING,  
div5airportid INT,  
div5airportseqid INT,  
div5wheelson STRING,  
div5totalgtime INT,  
div5longestgtime INT,  
div5wheelsoff STRING,  
div5tailnum STRING  
)  
PARTITIONED BY (year STRING)  
ROW FORMAT DELIMITED  
  FIELDS TERMINATED BY ','  
  ESCAPED BY '\\'  
  LINES TERMINATED BY '\\n'  
LOCATION 's3://athena-examples-myregion/flight/csv/';
```

Ejecute `MSCK REPAIR TABLE` para actualizar los metadatos de la partición cada vez que se añada una partición nueva a esta tabla:

```
MSCK REPAIR TABLE flight_delays_csv;
```

Consulte las 10 principales rutas que tengan un retraso de más de 1 hora:

```
SELECT origin, dest, count(*) as delays  
FROM flight_delays_csv  
WHERE depdelayminutes > 60  
GROUP BY origin, dest  
ORDER BY 3 DESC  
LIMIT 10;
```

**Note**

Los datos de la tabla de vuelos provienen de [Vuelos](#) proporcionados por la [Oficina de Estadísticas del Transporte](#) del Departamento de Transporte de los Estados Unidos. Desaturado del original.

## Ejemplo de TSV

Para crear una tabla de Athena a partir de los datos de TSV almacenados en Amazon S3, utilice `ROW FORMAT DELIMITED` y especifique `\t` como delimitador del campo de pestañas, `\n` como separador de líneas y `\` como carácter de escape. En el siguiente fragmento se muestra esta sintaxis. No hay ningún ejemplo de datos de vuelo de TSV disponibles en la ubicación `athena-examples`, pero al igual que con la tabla CSV, debería ejecutar `MSCK REPAIR TABLE` para actualizar los metadatos de partición cada vez que se agregue una partición nueva.

```
...  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t'  
ESCAPED BY '\\'  
LINES TERMINATED BY '\n'  
...
```

## OpenCSVSerDe para el procesamiento de CSV

Cuando crea una tabla de datos CSV en Athena, determine qué SerDe utilizar en función de los tipos de valores contenidos en sus datos:

- Si los datos contienen valores entre comillas dobles ("), puede utilizar el [OpenCSV SerDe](#) para deserializar los valores en Athena. Si los datos no contienen valores entre comillas dobles ("), puede omitir la especificación del SerDe. En este caso, Athena utiliza la opción predeterminada, `LazySimpleSerDe`. Para obtener más información, consulte [LazySimpleSerDe para CSV, TSV y archivos con delimitación personalizada](#).
- Si los datos tienen valores `TIMESTAMP UNIX` numéricos (por ejemplo, `1579059880000`), utilice el `OpenCSVSerDe`. Si los datos utilizan el formato `java.sql.Timestamp`, utilice el formato `LazySimpleSerDe`.

## CSV SerDe (OpenCSVSerDe)

El [SerDe de OpenCSV](#) tiene las siguientes características para los datos de cadena:

- Utiliza comillas (") como carácter de comillas predeterminado y permite especificar caracteres separadores, de comillas y de escape como:

```
WITH SERDEPROPERTIES ("separatorChar" = ",", "quoteChar" = "`", "escapeChar" = "\\")
```

- No se pueden aplicar directamente los caracteres de escape `\t` y `\n`. Para aplicarlos, utilice `"escapeChar" = "\\`". Consulte el ejemplo de este tema.
- No admite saltos de línea incrustados en los archivos CSV.

Para tipos de datos distintos de `STRING`, el `OpenCSVSerDe` se comporta de la siguiente manera:

- Reconoce tipos de datos `BOOLEAN`, `BIGINT`, `INT` y `DOUBLE`.
- No reconoce valores vacíos o nulos en columnas definidas como un tipo de datos numérico, y los deja como `string`. Una solución alternativa es crear la columna con los valores nulos como `string` y, a continuación, utilizar `CAST` para convertir el campo de una consulta a un tipo de datos numérico, y proporcionar un valor predeterminado de `0` para valores nulos. Para obtener más información, consulte [Cuando consulto datos CSV en Athena, obtengo el error HIVE\\_BAD\\_DATA: Error parsing field value](#) en el Centro de conocimientos de AWS.
- Para las columnas especificadas con el tipo de dato `timestamp` en la instrucción `CREATE TABLE`, reconoce `TIMESTAMP` si se especifica en formato numérico de UNIX en milisegundos, como `1579059880000`.
  - El `OpenCSVSerDe` no admite `TIMESTAMP` en formato `java.sql.Timestamp` compatible con JDBC, como `"YYYY-MM-DD HH:MM:SS.ffffffffff"` (precisión de posición de 9 decimales).
- Para las columnas especificadas con el tipo de dato `DATE` en la instrucción `CREATE TABLE`, reconoce los valores como fechas si estos representan el número de días transcurridos desde el 1 de enero de 1970. Por ejemplo, el valor `18276` en una columna con el tipo de datos `date` se representa como `2020-01-15` cuando se consulta. En este formato UNIX, cada día se considera que tiene 86 400 segundos.
  - El `OpenCSVSerDe` no es compatible con `DATE` en cualquier otro formato directamente. Para procesar datos de marca de tiempo en otros formatos, puede definir la columna como `string` y, a continuación, utilizar las funciones de conversión de tiempo para devolver los resultados deseados en la consulta `SELECT`. Para obtener más información, consulte el artículo [Cuando](#)

[consulta una tabla en Amazon Athena, el resultado TIMESTAMP está vacío](#) en el [Centro de conocimientos de AWS](#).

- Si desea seguir convirtiendo las columnas al tipo deseado en una tabla, puede [crear una vista](#) de la tabla y utilizar CAST para la conversión.

Example Ejemplo: uso del tipo TIMESTAMP y el tipo DATE especificados en el formato numérico de UNIX.

Tenga en cuenta las siguientes tres columnas de datos separados por comas. Los valores de cada columna se encierran entre comillas dobles.

```
"unixvalue creationdate 18276 creationdatetime 1579059880000","18276","1579059880000"
```

La siguiente instrucción crea una tabla en Athena desde la ubicación del bucket de Amazon S3 especificada.

```
CREATE EXTERNAL TABLE IF NOT EXISTS testtimestamp1(
  `profile_id` string,
  `creationdate` date,
  `creationdatetime` timestamp
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
LOCATION 's3://DOC-EXAMPLE-BUCKET'
```

A continuación, ejecute la consulta siguiente:

```
SELECT * FROM testtimestamp1
```

La consulta devuelve el siguiente resultado, mostrando los datos de fecha y hora:

profile_id	creationdate
creationdatetime unixvalue creationdate 18276 creationdatetime 1579146280000 2020-01-15 03:44:40.000	2020-01-15

Example Ejemplo: secuencia de escape `\t` o `\n`

Considere los datos de prueba siguientes:

```
" \t\t\t\n 123 \t\t\t\n ",abc
" 456 ",xyz
```

La instrucción siguiente crea una tabla en Athena y especifica que "escapeChar" = "\\\".

```
CREATE EXTERNAL TABLE test1 (
  f1 string,
  s2 string)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES ("separatorChar" = ",", "escapeChar" = "\\")
LOCATION 's3://DOC-EXAMPLE-BUCKET/dataset/test1/'
```

A continuación, ejecute la consulta siguiente:

```
SELECT * FROM test1;
```

Devuelve este resultado, utilizando correctamente la secuencia de escape \t o \n:

f1	s2	
\t\t\t\n 123 \t\t\t\n		abc
456		xyz

Nombre del SerDe

### [El SerDe de CSV](#)

Nombre de la biblioteca

Para utilizar este SerDe, especifique el nombre completo de la clase ROW FORMAT SERDE. Especifique también los delimitadores dentro de SERDEPROPERTIES, tal y como se indica a continuación:

```
...
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
  "separatorChar" = ",",
  "quoteChar"     = "`",
  "escapeChar"   = "\\")
```



## Ignorar encabezados

Para ignorar los encabezados en los datos cuando defina una tabla, puede utilizar la propiedad de tabla `skip.header.line.count`, como en el siguiente ejemplo.

```
TBLPROPERTIES ("skip.header.line.count"="1")
```

Para ver ejemplos, consulte las instrucciones `CREATE TABLE` en [Consulta de los registros de flujo de Amazon VPC](#) y [Consultas de registros de Amazon CloudFront](#).

## Ejemplo

En este ejemplo se presupone que hay datos CSV guardados en `s3://DOC-EXAMPLE-BUCKET/mycsv/` con el siguiente contenido:

```
"a1","a2","a3","a4"
"1","2","abc","def"
"a","a1","abc3","ab4"
```

Utilice una instrucción `CREATE TABLE` para crear una tabla Athena basada en los datos. Haga referencia a la clase `OpenCSVSerDe` después de `ROW FORMAT SERDE` y especifique el separador de caracteres, el carácter de comillas y el carácter de escape en `WITH SERDEPROPERTIES`, como en el siguiente ejemplo.

```
CREATE EXTERNAL TABLE myopencsvtable (
  col1 string,
  col2 string,
  col3 string,
  col4 string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
  'separatorChar' = ',',
  'quoteChar' = '"',
  'escapeChar' = '\\'
)
STORED AS TEXTFILE
LOCATION 's3://DOC-EXAMPLE-BUCKET/mycsv/';
```

Consulte todos los valores de la tabla:

```
SELECT * FROM myopencsvtable;
```

La consulta devuelve los siguientes valores:

col1	col2	col3	col4
a1	a2	a3	a4
1	2	abc	def
a	a1	abc3	ab4

## El SerDe de ORC

Nombre del SerDe

OrcSerDe

Nombre de la biblioteca

Esta biblioteca utiliza la clase Apache Hive [OrcSerde.java](#) para los datos en formato ORC. Pasa el objeto de ORC al lector y de ORC al escritor.

### Ejemplos

#### Note

Reemplace *myregion* en `s3://athena-examples-myregion/path/to/data/` por el identificador de región donde se ejecuta Athena, por ejemplo, `s3://athena-examples-us-west-1/path/to/data/`.

En el siguiente ejemplo se crea una tabla de los datos de retrasos de vuelo en ORC. La tabla contiene particiones:

```
DROP TABLE flight_delays_orc;
CREATE EXTERNAL TABLE flight_delays_orc (
  yr INT,
  quarter INT,
  month INT,
  dayofmonth INT,
  dayofweek INT,
  flightdate STRING,
```

```
uniquecarrier STRING,  
airlineid INT,  
carrier STRING,  
tailnum STRING,  
flightnum STRING,  
originairportid INT,  
originairportseqid INT,  
origincitymarketid INT,  
origin STRING,  
origincityname STRING,  
originstate STRING,  
originstatefips STRING,  
originstatename STRING,  
originwac INT,  
destairportid INT,  
destairportseqid INT,  
destcitymarketid INT,  
dest STRING,  
destcityname STRING,  
deststate STRING,  
deststatefips STRING,  
deststatename STRING,  
destwac INT,  
crsdeptime STRING,  
deptime STRING,  
depdelay INT,  
depdelayminutes INT,  
depdel15 INT,  
departuredelaygroups INT,  
deptimeblk STRING,  
taxiout INT,  
wheelsoff STRING,  
wheelson STRING,  
taxiin INT,  
crsarrrtime INT,  
arrrtime STRING,  
arrrdelay INT,  
arrrdelayminutes INT,  
arrrdel15 INT,  
arrivaldelaygroups INT,  
arrrtimeblk STRING,  
cancelled INT,  
cancellationcode STRING,  
diverted INT,
```

```
crselapsedtime INT,  
actualelapsedtime INT,  
airtime INT,  
flights INT,  
distance INT,  
distancegroup INT,  
carrierdelay INT,  
weatherdelay INT,  
nasdelay INT,  
securitydelay INT,  
lateaircraftdelay INT,  
firstdeptime STRING,  
totaladdgtime INT,  
longestaddgtime INT,  
divairportlandings INT,  
divreacheddest INT,  
divactualelapsedtime INT,  
divarrdelay INT,  
divdistance INT,  
div1airport STRING,  
div1airportid INT,  
div1airportseqid INT,  
div1wheelson STRING,  
div1totalgtime INT,  
div1longestgtime INT,  
div1wheelsoff STRING,  
div1tailnum STRING,  
div2airport STRING,  
div2airportid INT,  
div2airportseqid INT,  
div2wheelson STRING,  
div2totalgtime INT,  
div2longestgtime INT,  
div2wheelsoff STRING,  
div2tailnum STRING,  
div3airport STRING,  
div3airportid INT,  
div3airportseqid INT,  
div3wheelson STRING,  
div3totalgtime INT,  
div3longestgtime INT,  
div3wheelsoff STRING,  
div3tailnum STRING,  
div4airport STRING,
```

```
div4airportid INT,  
div4airportseqid INT,  
div4wheelson STRING,  
div4totalgtime INT,  
div4longestgtime INT,  
div4wheelsoff STRING,  
div4tailnum STRING,  
div5airport STRING,  
div5airportid INT,  
div5airportseqid INT,  
div5wheelson STRING,  
div5totalgtime INT,  
div5longestgtime INT,  
div5wheelsoff STRING,  
div5tailnum STRING  
)  
PARTITIONED BY (year String)  
STORED AS ORC  
LOCATION 's3://athena-examples-myregion/flight/orc/'  
tblproperties ("orc.compress"="ZLIB");
```

Ejecute la instrucción `MSCK REPAIR TABLE` en la tabla para actualizar los metadatos de partición:

```
MSCK REPAIR TABLE flight_delays_orc;
```

Utilice esta consulta para obtener las 10 principales rutas con un retraso de más de 1 hora:

```
SELECT origin, dest, count(*) as delays  
FROM flight_delays_orc  
WHERE depdelayminutes > 60  
GROUP BY origin, dest  
ORDER BY 3 DESC  
LIMIT 10;
```

## El SerDe de Parquet

Nombre del SerDe

ParquetHiveSerDe se utiliza para los datos almacenados en [formato Parquet](#).

**Note**

Para convertir datos en formato Parquet, puede utilizar las consultas [CREATE TABLE AS SELECT \(CTAS\)](#). Para obtener más información, consulte [Creación de una tabla a partir de los resultados de una consulta \(CTAS\)](#), [Ejemplos de consultas CTAS](#) y [Uso de CTAS e INSERT INTO en ETL y análisis de datos](#).

**Nombre de la biblioteca**

Cuando Athena necesita deserializar datos almacenados en Parquet, utiliza la siguiente clase:  
`org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe`

Ejemplo: consulta de un archivo almacenado en Parquet

**Note**

Reemplace *myregion* en `s3://athena-examples-myregion/path/to/data/` por el identificador de región donde se ejecuta Athena, por ejemplo, `s3://athena-examples-us-west-1/path/to/data/`.

Utilice la instrucción CREATE TABLE siguiente para crear una tabla de Athena a partir de datos subyacentes almacenados en formato Parquet en Amazon S3:

```
CREATE EXTERNAL TABLE flight_delays_pq (  
  yr INT,  
  quarter INT,  
  month INT,  
  dayofmonth INT,  
  dayofweek INT,  
  flightdate STRING,  
  uniquecarrier STRING,  
  airlineid INT,  
  carrier STRING,  
  tailnum STRING,  
  flightnum STRING,  
  originairportid INT,  
  originairportseqid INT,  
  origincitymarketid INT,  
  origin STRING,
```

```
origincityname STRING,  
originstate STRING,  
originstatefips STRING,  
originstatename STRING,  
originwac INT,  
destairportid INT,  
destairportseqid INT,  
destcitymarketid INT,  
dest STRING,  
destcityname STRING,  
deststate STRING,  
deststatefips STRING,  
deststatename STRING,  
destwac INT,  
crsdeptime STRING,  
deptime STRING,  
depdelay INT,  
depdelayminutes INT,  
depdel15 INT,  
departuredelaygroups INT,  
deptimeblk STRING,  
taxiout INT,  
wheelsoff STRING,  
wheelson STRING,  
taxiin INT,  
crsarrrtime INT,  
arrtime STRING,  
arrdelay INT,  
arrdelayminutes INT,  
arrdel15 INT,  
arrivaldelaygroups INT,  
arrtimeblk STRING,  
cancelled INT,  
cancellationcode STRING,  
diverted INT,  
crselapsedtime INT,  
actualelapsedtime INT,  
airtime INT,  
flights INT,  
distance INT,  
distancegroup INT,  
carrierdelay INT,  
weatherdelay INT,  
nasdelay INT,
```

```
securitydelay INT,  
lateaircraftdelay INT,  
firstdeptime STRING,  
totaladdgtime INT,  
longestaddgtime INT,  
divairportlandings INT,  
divreacheddest INT,  
divactualelapsedtime INT,  
divarrdelay INT,  
divdistance INT,  
div1airport STRING,  
div1airportid INT,  
div1airportseqid INT,  
div1wheelson STRING,  
div1totalgtime INT,  
div1longestgtime INT,  
div1wheelsoff STRING,  
div1tailnum STRING,  
div2airport STRING,  
div2airportid INT,  
div2airportseqid INT,  
div2wheelson STRING,  
div2totalgtime INT,  
div2longestgtime INT,  
div2wheelsoff STRING,  
div2tailnum STRING,  
div3airport STRING,  
div3airportid INT,  
div3airportseqid INT,  
div3wheelson STRING,  
div3totalgtime INT,  
div3longestgtime INT,  
div3wheelsoff STRING,  
div3tailnum STRING,  
div4airport STRING,  
div4airportid INT,  
div4airportseqid INT,  
div4wheelson STRING,  
div4totalgtime INT,  
div4longestgtime INT,  
div4wheelsoff STRING,  
div4tailnum STRING,  
div5airport STRING,  
div5airportid INT,
```



```
div5airportseqid INT,  
div5wheelson STRING,  
div5totalgtime INT,  
div5longestgtime INT,  
div5wheelsoff STRING,  
div5tailnum STRING  
)  
PARTITIONED BY (year STRING)  
STORED AS PARQUET  
LOCATION 's3://athena-examples-myregion/flight/parquet/'  
tblproperties ("parquet.compression"="SNAPPY");
```

Ejecute la instrucción `MSCK REPAIR TABLE` en la tabla para actualizar los metadatos de partición:

```
MSCK REPAIR TABLE flight_delays_pq;
```

Consulte las 10 principales rutas que tengan un retraso de más de 1 hora:

```
SELECT origin, dest, count(*) as delays  
FROM flight_delays_pq  
WHERE depdelayminutes > 60  
GROUP BY origin, dest  
ORDER BY 3 DESC  
LIMIT 10;
```

#### Note

Los datos de la tabla de vuelos provienen de [Vuelos](#) proporcionados por la [Oficina de Estadísticas del Transporte](#) del Departamento de Transporte de los Estados Unidos. Desaturado del original.

### Ignorar las estadísticas de Parquet

Al leer los datos de Parquet, es posible que reciba mensajes de error como los siguientes:

```
HIVE_CANNOT_OPEN_SPLIT: Index x out of bounds for length y  
HIVE_CURSOR_ERROR: Failed to read x bytes  
HIVE_CURSOR_ERROR: FailureException at Malformed input: offset=x  
HIVE_CURSOR_ERROR: FailureException at java.io.IOException:
```

```
can not read class org.apache.parquet.format.PageHeader: Socket is closed by peer.
```

Para solucionar este problema, utilice la instrucción [ALTER TABLE SET TBLPROPERTIES](#) o [CREATE TABLE](#) para establecer la propiedad `parquet.ignore.statistics` de Parquet SerDe en `true`, como en los siguientes ejemplos.

### Ejemplo de CREATE TABLE

```
...  
ROW FORMAT SERDE  
'org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe'  
WITH SERDEPROPERTIES (  
'parquet.ignore.statistics'='true')  
STORED AS PARQUET  
...
```

### Ejemplo de ALTER TABLE

```
ALTER TABLE ... SET TBLPROPERTIES ('parquet.ignore.statistics'='true')
```

## SerDe de Regex

SerDe de Regex utiliza una expresión regular (regex) para deserializar datos extrayendo grupos de expresiones regulares en columnas de tabla.

Si una fila de los datos no coincide con la expresión regular, todas las columnas de la fila se devuelven como NULL. Si una fila coincide con la expresión regular pero tiene menos grupos de los esperados, los grupos que faltan son NULL. Si una fila de los datos coincide con la expresión regular pero tiene más columnas que grupos en la expresión regular, se ignoran las columnas adicionales.

Para obtener más información, consulte [Class RegexSerDe](#) en la documentación de Apache Hive.

Nombre del SerDe

RegexSerDe

Nombre de la biblioteca

RegexSerDe



de Amazon S3. Cuando ejecute una consulta, Athena guardará los resultados de una consulta en una ubicación de resultados de consulta que especifique. Esto le permitirá ver el historial de consultas, además de descargar y ver conjuntos de resultados de consultas.

En esta sección se proporciona orientación para ejecutar consultas de Athena sobre orígenes de datos y tipos de datos comunes mediante una variedad de instrucciones SQL. Se proporciona orientación general para trabajar con estructuras y operadores comunes, por ejemplo, para trabajar con matrices, concatenar, filtrar, aplanar y ordenar. Entre otros ejemplos se incluyen consultas de datos en tablas con estructuras y mapas anidados, tablas basadas en conjuntos de datos codificados en JSON y conjuntos de datos asociados con Servicios de AWS como registros de AWS CloudTrail y registros de Amazon EMR. La cobertura completa del uso de SQL estándar está fuera del alcance de esta documentación. Para obtener más información sobre SQL, consulte las referencias de los lenguajes [Trino](#) y [Presto](#).

## Temas

- [Ver planes de ejecución para consultas SQL](#)
- [Trabajo con resultados de las consultas, consultas recientes y archivos de salida](#)
- [Reutilización de los resultados de las consultas](#)
- [Visualización de estadísticas y detalles de ejecución de consultas completadas](#)
- [Uso de vistas](#)
- [Uso de consultas guardadas](#)
- [Uso de consultas parametrizadas](#)
- [Uso del optimizador basado en costes](#)
- [Consulta de datos de S3 Express One Zone](#)
- [Consulta de objetos de Amazon S3 Glacier restaurados](#)
- [Gestión de las actualizaciones de los esquemas](#)
- [Matrices de consulta](#)
- [Consulta de datos geoespaciales](#)
- [Consulta de JSON](#)
- [Uso de Machine Learning \(ML\) con Amazon Athena](#)
- [Consulta con funciones definidas por el usuario](#)
- [Consultas entre regiones](#)
- [Consulta de AWS Glue Data Catalog](#)

- [Consulta de registros de Servicio de AWS](#)
- [Consulta de registros de servidor web almacenados en Simple Storage Service \(Amazon S3\)](#)

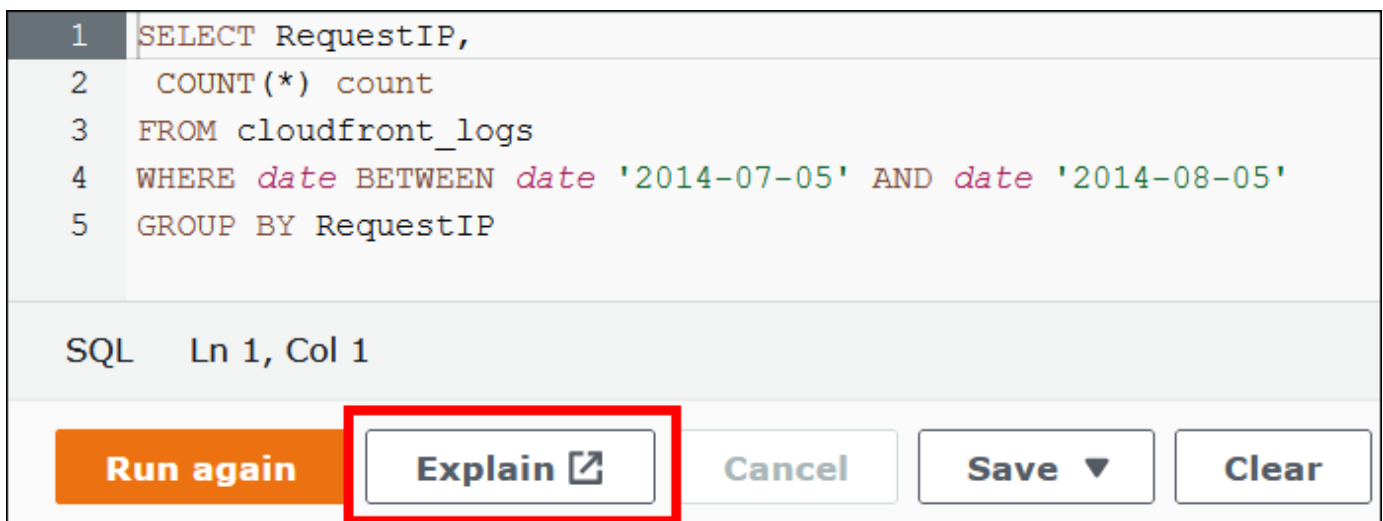
Para ver las consideraciones y limitaciones, consulte [Consideraciones y limitaciones de las consultas SQL en Amazon Athena](#).

## Ver planes de ejecución para consultas SQL

Puede usar el editor de consultas de Athena para ver representaciones gráficas que muestren cómo se ejecutará la consulta. Al ingresar una consulta en el editor y seleccionar la opción Explain (Explicar), Athena usa una instrucción SQL [EXPLAIN](#) en la consulta para crear dos gráficos correspondientes: un plan de ejecución distribuido y un plan de ejecución lógico. Puede utilizar estos gráficos para analizar, solucionar problemas y mejorar la eficiencia de sus consultas.



Para ver los planes de ejecución de una consulta

1. Ingrese la consulta en el editor de consultas de Athena y, a continuación, elija Explain (Explicar).



```
1 SELECT RequestIP,
2   COUNT(*) count
3 FROM cloudfront_logs
4 WHERE date BETWEEN date '2014-07-05' AND date '2014-08-05'
5 GROUP BY RequestIP
```

SQL Ln 1, Col 1

Run again Explain  Cancel Save  Clear

La pestaña Distributed plan (Plan distribuido) muestra el plan de ejecución de la consulta en un entorno distribuido. Un plan distribuido tiene fragmentos de procesamiento o etapas. Cada etapa tiene un número de índice basado en cero y es procesada por uno o más nodos. Los datos se pueden intercambiar entre nodos.

Amazon Athena > Query editor > Explain

# Explain

**Distributed plan** | Logical plan

---

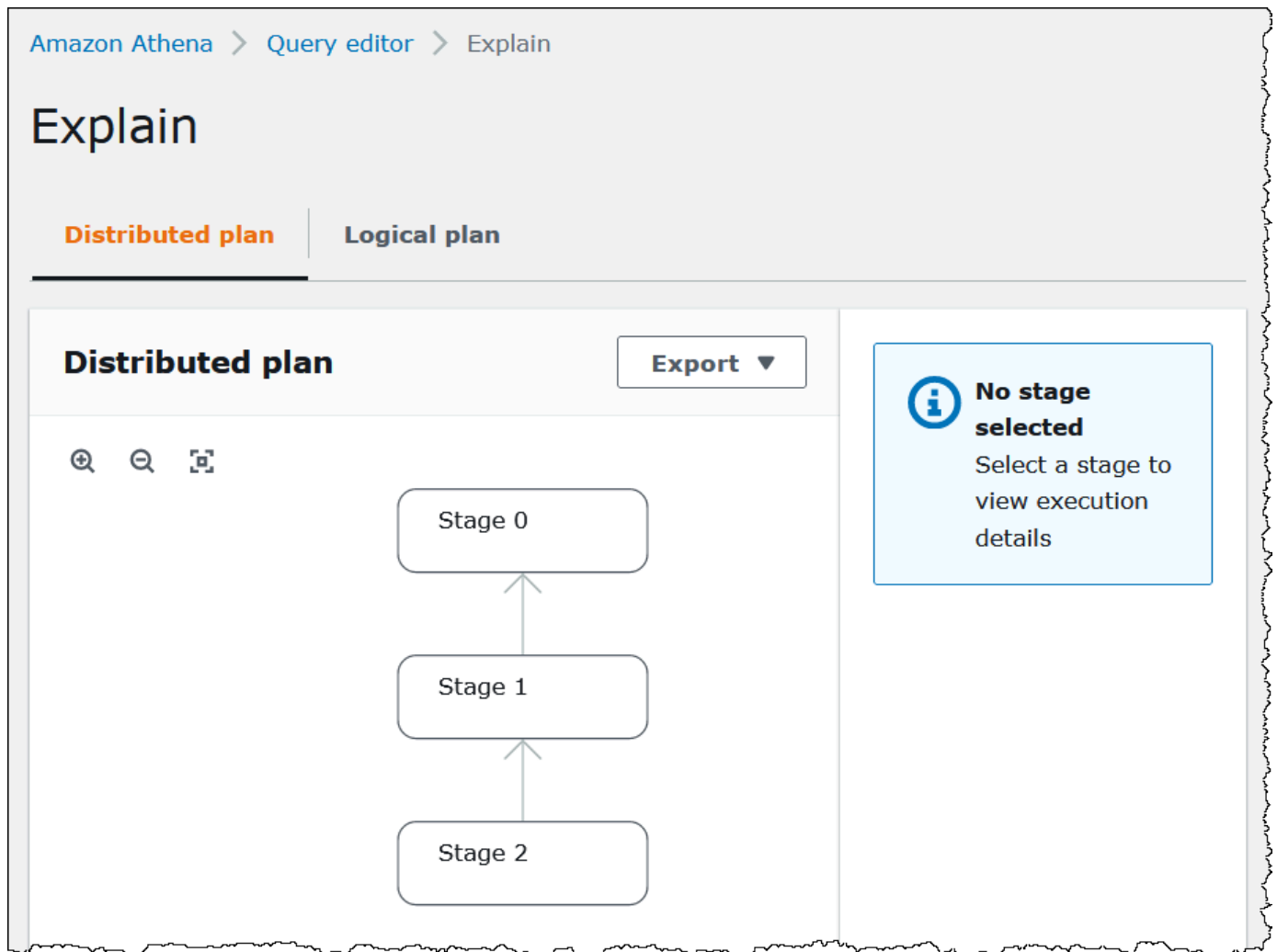
## Distributed plan

Export ▼

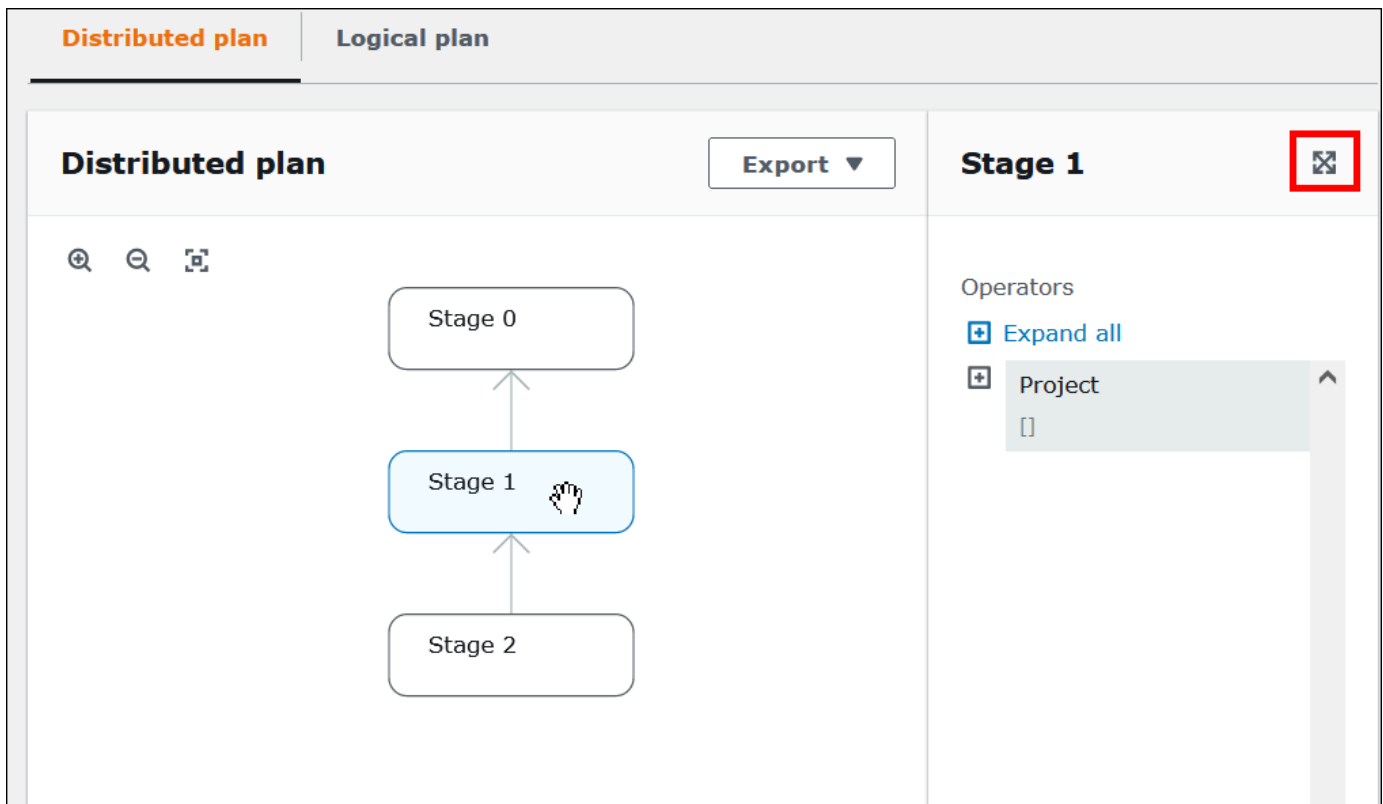
🔍 🔍 🖼️

```
graph BT; S2[Stage 2] --> S1[Stage 1]; S1 --> S0[Stage 0];
```

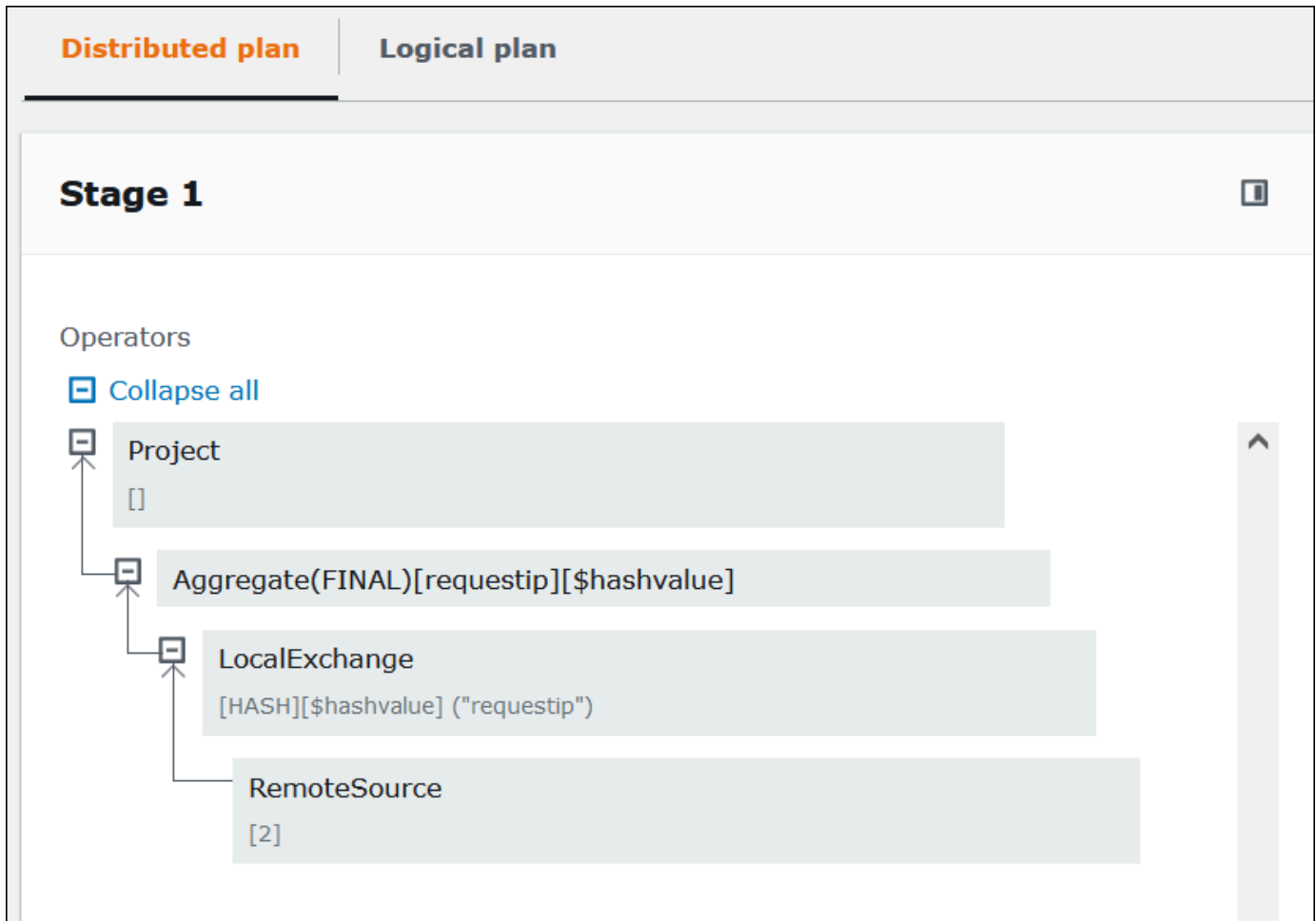
**No stage selected**  
Select a stage to view execution details



2. Para navegar por el gráfico, utilice las siguientes opciones:
  - Para ampliar o reducir la imagen, haga girar el ratón o utilice los iconos de la lupa.
  - Para ajustar el gráfico y que se ajuste a la pantalla, utilice el icono Aumentar para ajustar.
  - Para mover el gráfico, arrastre el puntero del ratón.
3. Para ver los detalles de una etapa, elija la etapa.



4. Para ver los detalles de la etapa en todo su ancho, elija el icono de ampliar en la parte superior derecha del panel de detalles.
5. Para ver más detalles, expanda uno o más elementos del árbol de operadores. Para obtener información acerca de los fragmentos de planes distribuidos, consulte [Tipos de salida de instrucciones EXPLAIN](#).

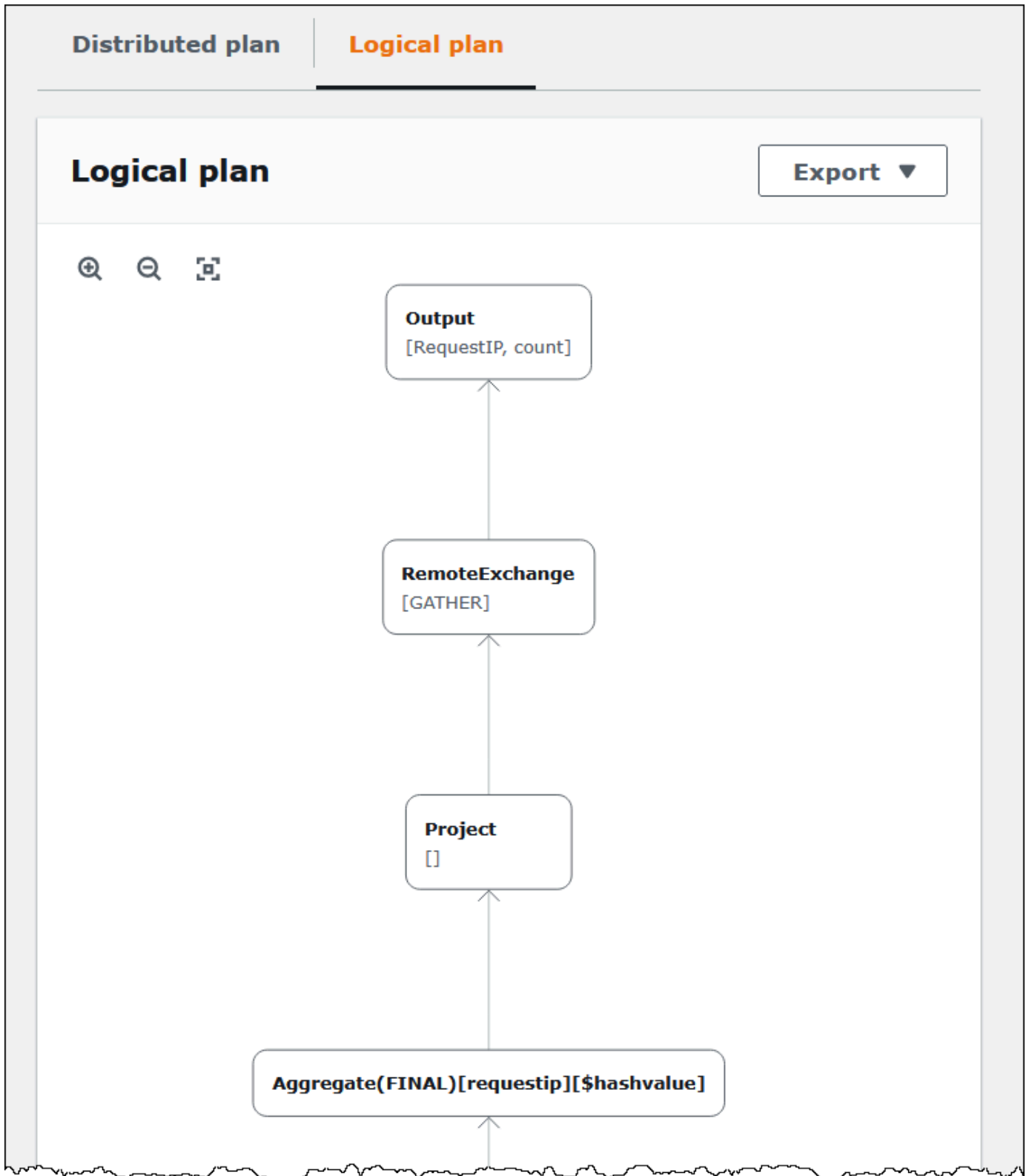


**⚠ Important**

Actualmente, es posible que algunos filtros de partición no estén visibles en el gráfico de árbol de operadores anidado, aunque Athena los aplique a la consulta. Para comprobar el efecto de dichos filtros, ejecute [EXPLAIN](#) o [EXPLAIN ANALYZE](#) en la consulta y vea los resultados.

6. Elija la pestaña Logical plan (Plan lógico). El gráfico muestra el plan lógico para ejecutar la consulta. Para obtener más información acerca de los términos de operación, consulte [Descripción de los resultados de la instrucción EXPLAIN de Athena](#).





7. Para exportar un plan como imagen SVG o PNG, o como texto JSON, elija Export (Exportar).

## Recursos adicionales de

Para obtener más información, consulte los siguientes recursos.

[Uso de EXPLAIN y EXPLAIN ANALYZE en Athena](#)

[Descripción de los resultados de la instrucción EXPLAIN de Athena](#)

[Visualización de estadísticas y detalles de ejecución de consultas completadas](#)

## Trabajo con resultados de las consultas, consultas recientes y archivos de salida

Amazon Athena almacena de forma automática los resultados y la información de metadatos de cada consulta que se ejecuta en una ubicación de resultados de consulta que puede especificar en Amazon S3. Si es necesario, puede obtener acceso a los archivos de esta ubicación para trabajar con ellos. También puede descargar archivos de resultados de consultas directamente desde la consola de Athena.

Para configurar una ubicación de resultados de consulta de Amazon S3 por primera vez, consulte [Especificación de una ubicación de resultados de consulta mediante la consola de Athena](#).

Los archivos de salida se guardan automáticamente por cada consulta que se hace. Para acceder y ver los archivos de salida de la consulta con la consola de Athena, las entidades principales de IAM (usuarios y roles) necesitan permiso para la acción [GetObject](#) de Amazon S3 para la ubicación del resultado de la consulta, así como permiso para la acción [GetQueryResults](#) de Athena. La ubicación del resultado de la consulta puede estar cifrada. Si la ubicación está cifrada, los usuarios deben tener los permisos de clave adecuados para cifrar y descifrar la ubicación del resultado de la consulta.

### Important

Las entidades principales de IAM con permiso para la acción `GetObject` de Amazon S3 para la ubicación del resultado de la consulta pueden recuperar los resultados de la consulta de Amazon S3, incluso si se deniega el permiso para la acción `GetQueryResults` de Athena.

## Especificación de una ubicación de resultados de consulta

La ubicación de los resultados de la consulta que Athena utiliza viene determinada por una combinación de configuración del grupo de trabajo y la configuración del lado del cliente. La configuración del lado del cliente se basa en cómo se ejecuta la consulta.

- Si ejecuta la consulta con la consola de Athena, la ubicación del resultado de la consulta, ingresada en Settings (Configuración) en la barra de navegación, determina la configuración del lado del cliente.
- Si ejecuta la consulta con la API de Athena, el parámetro `OutputLocation` de la acción [StartQueryExecution](#) determina la configuración del lado del cliente.
- Si utiliza los controladores ODBC o JDBC para ejecutar consultas, la propiedad `S3OutputLocation` especificada en la URL de conexión determina la configuración del lado del cliente.

### Important

Cuando se ejecuta una consulta con la API o con el controlador ODBC o JDBC, la configuración de la consola no se aplica.

Cada configuración de grupo de trabajo tiene una opción [Override client-side settings \(Anular configuración del lado del cliente\)](#) que se puede habilitar. Cuando esta opción está habilitada, la configuración del grupo de trabajo prevalece sobre la configuración del lado del cliente aplicable cuando una entidad principal de IAM asociada a ese grupo de trabajo ejecuta la consulta.


### Especificación de una ubicación de resultados de consulta mediante la consola de Athena

Antes de poder ejecutar una consulta, se debe especificar una ubicación de bucket de resultado de consulta en Amazon S3 o debe utilizar un grupo de trabajo que tenga especificado un bucket y cuya configuración sobrescriba la configuración del cliente.

Para especificar una ubicación de resultados de consulta de configuración del lado del cliente mediante la consola de Athena


1. [Cambie](#) al grupo de trabajo para el que desea especificar una ubicación de resultados de consulta. El nombre del grupo de trabajo predeterminado es Primary (Principal).
2. En la barra de navegación, elija Settings (Configuración).

3. En la barra de navegación, elija Manage (Administrar).
4. En Manage settings (Administrar configuración), realice una de las siguientes operaciones:
  - En el cuadro Location of query result (Ubicación del resultado de la consulta), ingrese la ruta de acceso al bucket que creó en Amazon S3 para los resultados de la consulta. Prefije la ruta con `s3://`.
  - Elija Browse S3 (Navegar S3), elija el bucket de Amazon S3 que creó para su región actual y, a continuación, elija Choose (Elegir).

 Note

Si utiliza un grupo de trabajo que especifica una ubicación del resultado de la consulta para todos los usuarios del grupo de trabajo, la opción para cambiar la ubicación del resultado de la consulta no estará disponible.

5. (Opcional) Seleccione View lifecycle configuration (Ver configuración del ciclo de vida) para ver y configurar las [reglas del ciclo de vida de Amazon S3](#) en el bucket de resultados de consultas. Las reglas del ciclo de vida de Amazon S3 que cree pueden ser reglas de caducidad o reglas de transición. Las reglas de caducidad eliminan automáticamente los resultados de consultas transcurrido cierto tiempo. Las reglas de transición los mueven a otro nivel de almacenamiento de Amazon S3. Para obtener más información, consulte [Configurar el ciclo de vida de un bucket](#) en la Guía del usuario de Amazon Simple Storage Service.
6. (Opcional) En Expected bucket owner (Propietario esperado del bucket), ingrese el ID de la Cuenta de AWS que espera que sea el propietario del bucket de ubicación de salida. Se trata de una medida de seguridad añadida. Si el ID de cuenta del propietario del bucket no coincide con el ID que se especifique aquí, los intentos de generar el bucket fallarán. Para obtener información detallada, consulte [Verificación de la propiedad del bucket con la condición de propietario del bucket](#) en la Guía del usuario de Amazon S3.

 Note

La configuración del propietario esperado del bucket se aplica solo a la ubicación de salida de Amazon S3 especificada para los resultados de las consultas de Athena. No se aplica a otras ubicaciones de Amazon S3, como las ubicaciones de origen de datos en buckets externos de Amazon S3, ubicaciones de tablas de destino de CTAS y INSERT INTO, ubicaciones de salida de instrucciones UNLOAD, operaciones para desbordar

buckets para consultas federadas, o consultas SELECT ejecutadas en una tabla de otra cuenta.

7. (Opcional) Elija Encrypt query results (Cifrar los resultados de las consultas) si desea cifrar los resultados de las consultas almacenados en Simple Storage Service (Amazon S3). Para obtener más información sobre el cifrado en Athena, consulte [Cifrado en reposo](#).
8. (Opcional) Elija Assign bucket owner full control over query results (Asignar control total sobre los resultados de las consultas al propietario del bucket) para conceder acceso de control total sobre los resultados de las consultas al propietario del bucket cuando [estén habilitadas las ACL](#) para el bucket de resultados de las consultas. Por ejemplo, si la ubicación del resultado de las consultas es propiedad de otra cuenta, puede conceder la propiedad y el control total sobre los resultados de las consultas a la otra cuenta. Para obtener más información, consulte [Control de la propiedad de los objetos y desactivación de las ACL del bucket](#) en la Guía del usuario de Simple Storage Service (Amazon S3).
9. Seleccione Guardar.

## Ubicaciones predeterminadas creadas anteriormente

Anteriormente en Athena, si ejecutaba una consulta sin especificar un valor para Query result location (Ubicación de resultado de consulta) y un grupo de trabajo no anulaba la configuración de la ubicación del resultado de la consulta, Athena creaba una ubicación predeterminada automáticamente. La ubicación predeterminada era `aws-athena-query-results-MyAcctID-MyRegion`, donde *MyAcctID* era el ID de cuenta de Amazon Web Services de la entidad principal de IAM que ejecutaba la consulta y *MyRegion* era la región donde se ejecutaba la consulta (por ejemplo, `us-west-1`).

Ahora, antes de poder ejecutar una consulta de Athena en una región en la que su cuenta no haya utilizado Athena anteriormente, debe especificar una ubicación de resultado de la consulta o utilizar un grupo de trabajo que anule la configuración de ubicación del resultado de la consulta. Aunque Athena ya no crea una ubicación predeterminada de resultados de consulta, las ubicaciones `aws-athena-query-results-MyAcctID-MyRegion` predeterminadas creadas anteriormente siguen siendo válidas y puede seguir utilizándolas.

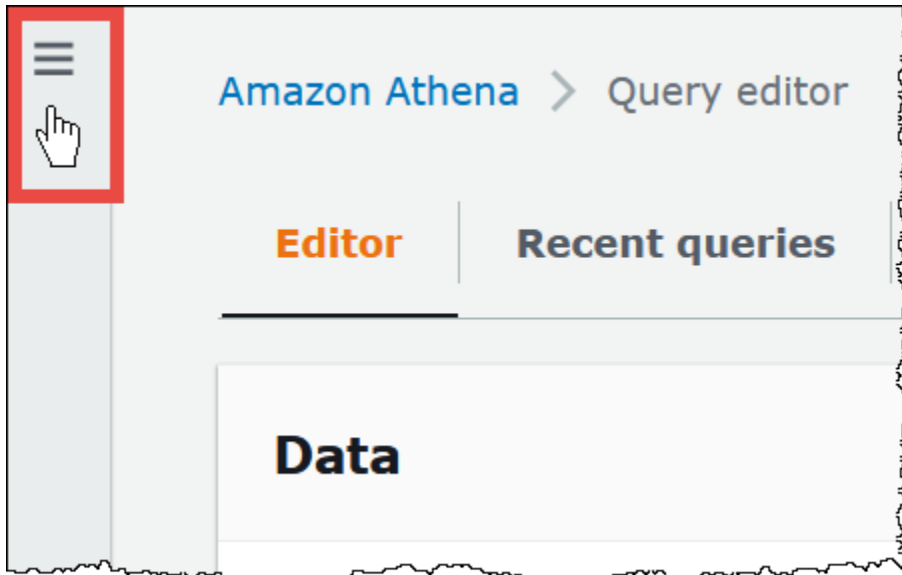
## Especificación de una ubicación de resultados de consulta mediante un grupo de trabajo

Puede especificar la ubicación del resultado de la consulta en una configuración de grupo de trabajo mediante la AWS Management Console, la AWS CLI o la API de Athena.

Cuando utilice la AWS CLI, especifique la ubicación del resultado de la consulta mediante el parámetro `OutputLocation` de la opción `--configuration` cuando ejecute el comando [aws athena create-work-group](#) o [aws athena update-work-group](#).

Para especificar la ubicación del resultado de la consulta para un grupo de trabajo mediante la consola de Athena

1. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.



2. En el panel de navegación, elija Workgroups (Redes globales).
3. En la lista de grupos de trabajo, elija el enlace del grupo de trabajo que quiere editar.
4. Elija Editar.
5. Para Ubicación de resultados de consulta y cifrado, realice una de las siguientes operaciones:
  - En el cuadro Location of query result (Ubicación del resultado de la consulta), ingrese la ruta de acceso al bucket de Amazon S3 para los resultados de la consulta. Prefije la ruta con `s3://`.
  - Elija Browse S3 (Navegar S3), elija el bucket de Amazon S3 para la región actual que quiere utilizar y, a continuación, elija Choose (Elegir).
6. (Opcional) En Expected bucket owner (Propietario esperado del bucket), ingrese el ID de la Cuenta de AWS que espera que sea la propietaria del bucket de ubicación de salida. Se trata de una medida de seguridad añadida. Si el ID de cuenta del propietario del bucket no coincide con el ID que se especifique aquí, los intentos de generar el bucket fallarán. Para obtener

información detallada, consulte [Verificación de la propiedad del bucket con la condición de propietario del bucket](#) en la Guía del usuario de Amazon S3.

 Note

La configuración del propietario esperado del bucket se aplica solo a la ubicación de salida de Amazon S3 especificada para los resultados de las consultas de Athena. No se aplica a otras ubicaciones de Amazon S3, como las ubicaciones de origen de datos en buckets externos de Amazon S3, ubicaciones de tablas de destino de CTAS y INSERT INTO, ubicaciones de salida de instrucciones UNLOAD, operaciones para desbordar buckets para consultas federadas, o consultas SELECT ejecutadas en una tabla de otra cuenta.

7. (Opcional) Elija Encrypt query results (Cifrar los resultados de las consultas) si desea cifrar los resultados de las consultas almacenados en Simple Storage Service (Amazon S3). Para obtener más información sobre el cifrado en Athena, consulte [Cifrado en reposo](#).
8. (Opcional) Elija Assign bucket owner full control over query results (Asignar control total sobre los resultados de las consultas al propietario del bucket) para conceder acceso de control total sobre los resultados de las consultas al propietario del bucket cuando [estén habilitadas las ACL](#) para el bucket de resultados de las consultas. Por ejemplo, si la ubicación del resultado de las consultas es propiedad de otra cuenta, puede conceder la propiedad y el control total sobre los resultados de las consultas a la otra cuenta.

Si la configuración de propiedad de objetos de S3 del bucket es Bucket owner preferred (Propietario del bucket preferido), el propietario del bucket también es propietario de todos los objetos de resultados de las consultas escritos desde este grupo de trabajo. Por ejemplo, si el grupo de trabajo de una cuenta externa habilita esta opción y establece la ubicación del resultado de la consulta en el bucket de Simple Storage Service (Amazon S3) de la cuenta que tiene Bucket owner preferred (Propietario del bucket preferido) como configuración de propiedad de objetos de S3, usted es propietario de los resultados de las consultas del grupo de trabajo externo y tiene control total sobre ellos.

Seleccionar esta opción cuando la configuración de la propiedad de objetos S3 del bucket de resultados de las consultas es Bucket owner enforced (Propietario del bucket obligatorio) no surte ningún efecto. Para obtener más información, consulte [Control de la propiedad de los objetos y desactivación de las ACL del bucket](#) en la Guía del usuario de Simple Storage Service (Amazon S3).

9. Si quiere solicitar a todos los usuarios del grupo de trabajo que utilicen la ubicación de los resultados de la consulta especificada, desplácese hacia abajo hasta la sección Settings (Configuración) y seleccione Override client-side settings (Invalidar la configuración del lado del cliente).
10. Elija Guardar cambios.

## Descarga de archivos de resultados de consultas mediante la consola de Athena

Puede descargar el archivo CSV de resultados de la consulta desde el panel de consultas, inmediatamente después de ejecutar una consulta. También puede descargar los resultados de las consultas recientes de la pestaña Recent queries (Consultas recientes).

### Note

Los archivos de resultados de la consulta de Athena son archivos de datos que contienen información que los usuarios pueden configurar. Algunos programas que se utilizan para leer y analizar esta información podrían interpretar estos datos como comandos (inyección CSV). Por este motivo, al importar datos de CSV de resultados de consulta a un programa de hoja de cálculo, este programa podría advertirle acerca de problemas de seguridad. Para mantener el sistema seguro, siempre debe optar por desactivar enlaces o macros de los resultados de consulta descargados.

Para ejecutar una consulta y descargar los resultados de la consulta

1. Ingrese la consulta en el editor de consultas y, a continuación, elija Run (Ejecutar).

Cuando la consulta termina de ejecutarse, el panel Results (Resultados) muestra los resultados de la consulta.

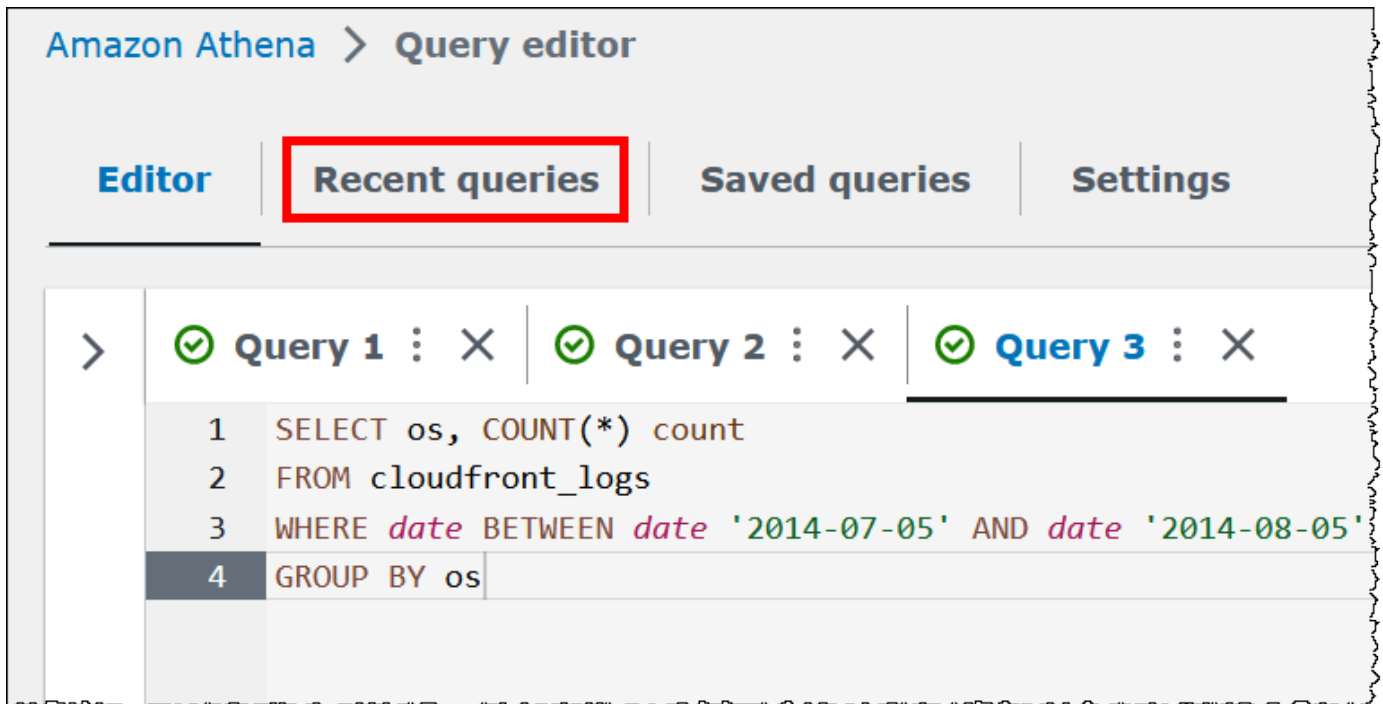
2. Para descargar un archivo CSV de los resultados de la consulta, elija Download results (Descargar resultados) encima del panel de resultados de la consulta. En función del navegador y su configuración, es posible que tenga que confirmar la descarga.





Para descargar un archivo de resultados de consulta para una consulta anterior

1. Elija Recent queries (Consultas recientes).



2. Utilice el cuadro de búsqueda para encontrar la consulta, seleccione la consulta y, a continuación, elija Download results (Descargar resultados).

#### Note

No se puede utilizar la opción Download results (Descargar resultados) para recuperar resultados de consultas que se hayan eliminado manualmente, ni recuperar resultados

de consultas que se hayan eliminado o movido a otra ubicación debido a [reglas del ciclo de vida](#) de Amazon S3.

Amazon Athena > Query editor

Workgroup primary

Editor Recent queries Saved queries Settings

Recent queries (1/42) [Refresh] [Cancel] [Download results]

[Search recent queries]

< 1 2 3 > [Settings]

2	Execution ID	Query
[Select]	3679f78b-5228-4810-afd3-09d97a85075f	SELECT os, COUNT(*) count
[Select]	ae8c4fa1-8a65-4bfc-aa77-471cde2ca1af	SELECT os, COUNT(*) count

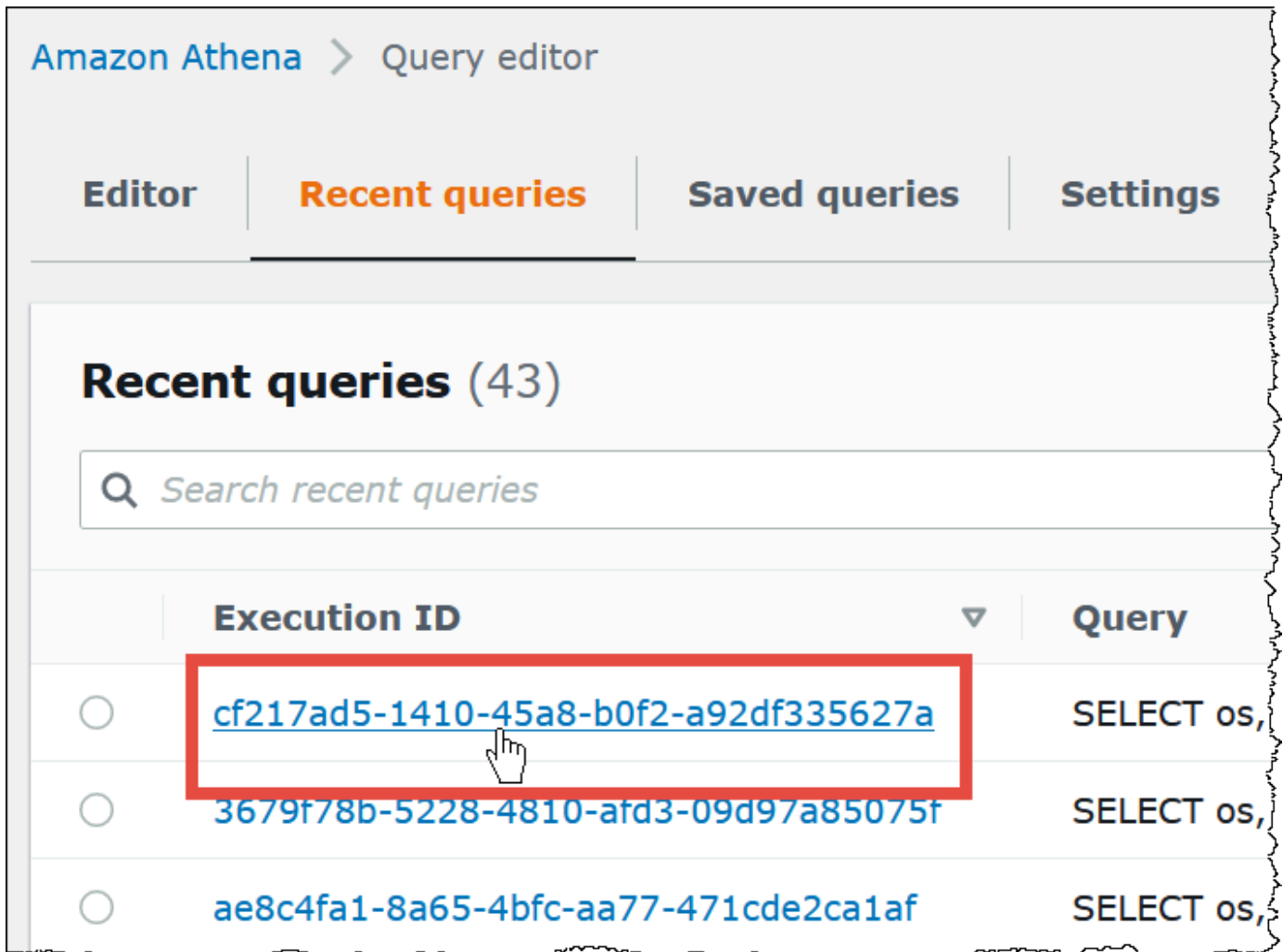
## Vista de consultas recientes

Puede utilizar la consola de Athena para ver qué consultas se hicieron correctamente o no pudieron hacerse y ver los detalles de error de las consultas que no pudieron hacerse. Athena conserva el historial de consultas durante 45 días.

Para ver las consultas recientes en la consola de Athena

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Elija Recent queries (Consultas recientes). En la pestaña Recent queries (Consultas recientes) se muestra información sobre cada consulta ejecutada.

- Para abrir una instrucción de consulta en el editor de consultas, elija el ID de ejecución de la consulta.



The screenshot shows the Amazon Athena Query Editor interface. At the top, there are four tabs: "Editor", "Recent queries" (which is selected and highlighted in orange), "Saved queries", and "Settings". Below the tabs, the heading "Recent queries (43)" is displayed. A search bar with the placeholder text "Search recent queries" is located below the heading. A table lists recent queries with the following columns: "Execution ID" and "Query". The first row in the table has an "Execution ID" of [cf217ad5-1410-45a8-b0f2-a92df335627a](#) and a "Query" starting with "SELECT os,". This row is highlighted with a red rectangular box, and a mouse cursor is pointing at the execution ID. The second row has an "Execution ID" of [3679f78b-5228-4810-afd3-09d97a85075f](#) and a "Query" starting with "SELECT os,". The third row has an "Execution ID" of [ae8c4fa1-8a65-4bfc-aa77-471cde2ca1af](#) and a "Query" starting with "SELECT os,".

	Execution ID	Query
<input type="radio"/>	<a href="#">cf217ad5-1410-45a8-b0f2-a92df335627a</a>	SELECT os,
<input type="radio"/>	<a href="#">3679f78b-5228-4810-afd3-09d97a85075f</a>	SELECT os,
<input type="radio"/>	<a href="#">ae8c4fa1-8a65-4bfc-aa77-471cde2ca1af</a>	SELECT os,

- Para ver los detalles de una consulta que ha producido un error, elija el enlace Failed (Error) de la consulta.

The screenshot shows the Amazon Athena console interface. At the top, there are buttons for 'Cancel' and 'Download results'. Below the buttons is a pagination control showing page 1 of 3. The main area displays a table of query results with columns for 'Start time', 'Status', and 'Run time'. An error modal is open over the first two rows of the table, which are marked as 'Failed'. The modal contains the following text:

**Error** [Close]

Query ID  
6a242b5c-226b-4a51-aec6-e9667c5bcd66 [Copy]

Error details  
SYNTAX\_ERROR: line 1:18: Table  
awsdatacatalog.mydatabase.mytable does not exist

This query ran against the "mydatabase" database, unless qualified by the query. Please post the error message on our [forum](#) or contact [customer support](#) with query id.

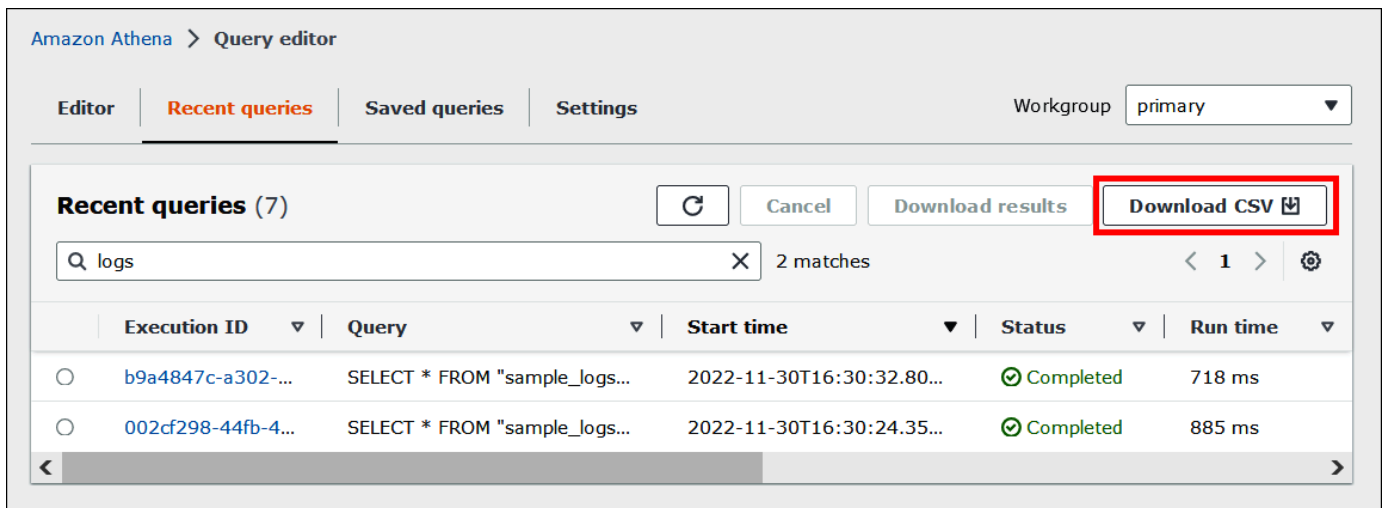
Start time	Status	Run time
	Failed	0.229 sec
	Failed	0.203 sec
	Completed	3.484 sec
	Completed	3.143 sec
	Completed	3.517 sec
	Completed	3.398 sec
	Completed	3.412 sec

## Descarga de varias consultas recientes a un archivo CSV

Puede utilizar la pestaña Recent queries (Consultas recientes) de la consola de Athena para exportar una o más consultas recientes a un archivo CSV para verlas en formato tabular. El archivo descargado no contiene los resultados de la consulta, sino la cadena de consulta SQL propiamente dicha y otra información sobre la consulta. Los campos exportados incluyen el ID de ejecución, el contenido de la cadena de consulta, la hora de inicio de la consulta, el estado, el tiempo de ejecución, la cantidad de datos escaneados, la versión del motor de consultas utilizada y el método de cifrado. Puede exportar un máximo de 500 consultas recientes o un máximo filtrado de 500 consultas mediante los criterios que introduzca en el cuadro de búsqueda.

Para exportar una o más consultas recientes a un archivo CSV

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Elija Recent queries (Consultas recientes).
3. (Opcional) Utilice el cuadro de búsqueda para filtrar las consultas recientes que desea descargar.
4. Elija Download CSV (Descargar CSV).



The screenshot shows the Amazon Athena Query editor interface. At the top, there are tabs for 'Editor', 'Recent queries', 'Saved queries', and 'Settings'. The 'Recent queries' tab is active. Below the tabs, there is a search bar with the text 'logs' and '2 matches'. To the right of the search bar, there are buttons for 'Cancel', 'Download results', and 'Download CSV' (highlighted with a red box). Below the search bar, there is a table with columns: Execution ID, Query, Start time, Status, and Run time. The table contains two rows of query results, both with a status of 'Completed'.

Execution ID	Query	Start time	Status	Run time
b9a4847c-a302-...	SELECT * FROM "sample_logs..."	2022-11-30T16:30:32.80...	Completed	718 ms
002cf298-44fb-4...	SELECT * FROM "sample_logs..."	2022-11-30T16:30:24.35...	Completed	885 ms

5. Cuando aparezca el mensaje para guardar el archivo, seleccione Save (Guardar). El nombre de archivo predeterminado es Recent Queries seguido de una marca de tiempo (por ejemplo, Recent Queries 2022-12-05T16 04 27.352-08 00.csv).

### Configuración de las opciones de visualización de consultas recientes

Puede configurar las opciones de la pestaña Recent queries (Consultas recientes) como las columnas que mostrar y el ajuste de texto.

Para configurar las opciones de la pestaña Recent queries (Consultas recientes)

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Elija Recent queries (Consultas recientes).
3. Seleccione el botón de opciones (icono con forma de engranaje).

**Editor** | **Recent queries** | **Saved queries** | **Settings**

**Recent queries (1/45)** [Refresh] [Cancel] [Download results]

Q Search recent queries

< **1** 2 3 > [Settings]

Execution ID	Query
6a242b5c-226b-4a51-aec6-e9667c5bcd6	Select abcd from mytable

4. En el cuadro de diálogo Preferencias (Preferencias), elija el número de filas por página, el comportamiento del ajuste de líneas y las columnas que mostrar.

# Preferences



## Select rows per page

10 queries

20 queries

Wrap lines

Wraps long lines to show all the text

## Select visible content

### Properties

Execution ID



Query



Start time



Run time



Status



Data scanned



Query engine version used



Encryption



## 5. Elija Confirmar.

Mantener el historial de consultas durante más de 45 días

Si desea mantener el historial de consultas durante más de 45 días, puede recuperarlo y guardarlo en un almacén de datos como Amazon S3. Para automatizar este proceso, puede usar acciones de la API y comandos de la CLI de Athena y Amazon S3. En el siguiente procedimiento se resumen estos pasos.

Para recuperar y guardar el historial de consultas mediante programación

1. Utilice la acción de la API de Athena [ListQueryExecutions](#) o el comando de la CLI [list-query-executions](#) para recuperar los ID de consulta.
2. Utilice la acción de la API de Athena [GetQueryExecution](#) o el comando de la CLI [get-query-execution](#) para recuperar información sobre cada consulta en función de su ID.
3. Utilice la acción de la API de Amazon S3 [PutObject](#) o el comando de la CLI [put-object](#) para guardar la información en Amazon S3.

## Búsqueda de archivos de salida de consulta en Amazon S3

Los archivos de salida de la consulta se almacenan en subcarpetas en Amazon S3 con el siguiente patrón de ruta, a menos que la consulta se produzca en un grupo de trabajo cuya configuración invalida la configuración del lado del cliente. Cuando la configuración del grupo de trabajo invalida la configuración del lado del cliente, la consulta utiliza la ruta de resultados especificada por el grupo de trabajo.

```
QueryResultsLocationInS3/[QueryName | Unsaved/yyyy/mm/dd/]
```

- *QueryResultsLocationInS3* es la ubicación del resultado de la consulta especificada por la configuración del grupo de trabajo o la configuración del lado del cliente. Para obtener más información, consulte [the section called “Especificación de una ubicación de resultados de consulta”](#) más adelante en este documento.
- Las siguientes subcarpetas se crean solo para las consultas ejecutadas desde la consola cuya ruta de resultados no ha sido invalidada por la configuración del grupo de trabajo. Las consultas que se ejecutan desde la AWS CLI o mediante la API de Athena se guardan directamente en *QueryResultsLocationInS3*.



- *QueryName* es el nombre de la consulta cuyos resultados se guardan. Si la consulta se ejecutó pero no se guardó, se utiliza Unsaved.
- *aaaa/mm/dd* es la fecha de ejecución de la consulta.

Los archivos asociados con una consulta CREATE TABLE AS SELECT se almacenan en una subcarpeta tables del patrón anterior.

### Identificación de archivos de salida de consultas

Los archivos se guardan en la ubicación de resultados de las consultas de Amazon S3 en función del nombre, el ID y la fecha de ejecución de la consulta. A los archivos de cada consulta se les asigna nombre utilizando el *QueryID*, que es un identificador único que Athena asigna a cada consulta cuando se ejecuta.

Se guardan los siguientes tipos de archivos:

Tipo de archivo	Patrones de nomenclatura de archivos	Descripción
Archivos de resultados de consulta	<p><i>QueryID</i>.csv</p> <p><i>QueryID</i>.txt</p>	<p>Los archivos de resultados de consulta DML se guardan en un formato de valores separados por comas (CSV).</p> <p>Los resultados de la consulta DDL se guardan como archivos de texto sin formato.</p> <p>Puede descargar los archivos de resultados desde el panel Results (Resultados) de la consola cuando utilice la consola o desde el historial de consultas. Para obtener más información, consulte <a href="#">Descarga de archivos de resultados de consultas</a></p>

Tipo de archivo	Patrones de nomenclatura de archivos	Descripción
		<a href="#">mediante la consola de Athena.</a>
Archivos de metadatos de consulta	<i>QueryID.csv.metadata</i> <i>QueryID.txt.metadata</i>	Los archivos de metadatos de consulta DML y DDL se guardan en formato binario y no son legibles por el usuario. La extensión del archivo corresponde al archivo de resultados de consulta relacionado. Athena utiliza los metadatos al leer los resultados de la consulta mediante la acción <code>GetQueryResults</code> . Aunque estos archivos se pueden eliminar, no lo recomendamos porque se pierde información importante sobre la consulta.

Tipo de archivo	Patrones de nomenclatura de archivos	Descripción
Archivos de manifiesto de datos	<i>QueryID</i> -manifest.csv	Los archivos de manifiesto de datos se generan para realizar un seguimiento de los archivos que Athena crea en las ubicaciones de origen de datos de Amazon S3 cuando se ejecuta una consulta <a href="#">INSERT INTO</a> . Si se produce un error en una consulta, el manifiesto también realiza un seguimiento de los archivos que la consulta pretendía escribir. El manifiesto resulta útil para identificar los archivos que se quedaron huérfanos tras una consulta fallida.

Identificación de la ubicación de salida y los archivos de resultados de la consulta con la AWS CLI

Para utilizar la AWS CLI para identificar la ubicación de salida y los archivos de resultados de la consulta, ejecute el comando `aws athena get-query-execution`, como se muestra en el siguiente ejemplo. Sustituya *abc1234d-5efg-67hi-jklm-89n0op12qr34* por el ID de la consulta.

```
aws athena get-query-execution --query-execution-id abc1234d-5efg-67hi-
jklm-89n0op12qr34
```

El comando devuelve un resultado similar al siguiente. Para ver una descripción de cada parámetro de salida, consulte [get-query-execution](#) en la Referencia de comandos de la AWS CLI.

```
{
  "QueryExecution": {
    "Status": {
      "SubmissionDateTime": 1565649050.175,
```

```

        "State": "SUCCEEDED",
        "CompletionDateTime": 1565649056.6229999
    },
    "Statistics": {
        "DataScannedInBytes": 5944497,
        "DataManifestLocation": "s3://DOC-EXAMPLE-BUCKET/athena-query-
results-123456789012-us-west-1/MyInsertQuery/2019/08/12/abc1234d-5efg-67hi-
jklm-89n0op12qr34-manifest.csv",
        "EngineExecutionTimeInMillis": 5209
    },
    "ResultConfiguration": {
        "EncryptionConfiguration": {
            "EncryptionOption": "SSE_S3"
        },
        "OutputLocation": "s3://DOC-EXAMPLE-BUCKET/athena-query-
results-123456789012-us-west-1/MyInsertQuery/2019/08/12/abc1234d-5efg-67hi-
jklm-89n0op12qr34"
    },
    "QueryExecutionId": "abc1234d-5efg-67hi-jklm-89n0op12qr34",
    "QueryExecutionContext": {},
    "Query": "INSERT INTO mydb.elb_log_backup SELECT * FROM mydb.elb_logs LIMIT
100",
    "StatementType": "DML",
    "WorkGroup": "primary"
}
}

```

## Reutilización de los resultados de las consultas

Al volver a ejecutar una consulta en Athena, puede optar por reutilizar el último resultado almacenado de la consulta. Esta opción puede aumentar el rendimiento y reducir los costos en términos del número de bytes escaneados. Reutilizar los resultados de las consultas es útil si, por ejemplo, sabe que los resultados no cambiarán dentro de un periodo de tiempo determinado. Puede especificar una antigüedad máxima para reutilizar resultados de las consultas. Athena usa el resultado almacenado siempre que no sea anterior a la antigüedad que usted especifique. Para obtener más información, consulte [Reduce cost and improve query performance with Amazon Athena](#) (Reducir los costos y mejorar el rendimiento de las consultas con Amazon Athena) en el blog sobre macrodatos de AWS.

**Note**

La función de reutilización de resultados de consultas requiere la versión 3 del motor Athena. Para obtener más información acerca del cambio de las versiones del motor Athena, consulte [Cambio de las versiones del motor Athena](#).

## Características principales

- La reutilización de los resultados de las consultas es una característica opcional por consulta. Puede habilitar la reutilización de los resultados de las consultas por consulta.
- La antigüedad máxima para reutilizar resultados de las consultas puede especificarse en minutos, horas o días. La antigüedad máxima especificable equivale a 7 días, independientemente de la unidad de tiempo utilizada. El valor predeterminado es de 60 minutos.
- Al habilitar la reutilización de resultados para una consulta, Athena busca una ejecución anterior de la consulta dentro del mismo grupo de trabajo. Si Athena encuentra los resultados de la consulta almacenados correspondientes, no vuelve a ejecutar la consulta, sino que apunta a la ubicación del resultado anterior o extrae datos de ella.
- Para cualquier consulta que habilite la opción de reutilización de resultados, Athena reutiliza el último resultado de la consulta guardado en la carpeta del grupo de trabajo solo cuando se cumplen todas las condiciones siguientes:
  - La cadena de consulta coincide de forma exacta.
  - El nombre del catálogo y la base de datos coinciden.
  - El resultado anterior no supera la antigüedad máxima especificada o no supera los 60 minutos si no se ha especificado una antigüedad máxima.
  - Athena solo reutiliza una ejecución que tenga exactamente la misma [configuración de resultados](#) que la ejecución actual.
  - Tiene acceso a todas las tablas a las que se hace referencia en la consulta.
  - Tiene acceso a la ubicación del archivo de S3 donde se almacena el resultado anterior.

Si no se cumple alguna de estas condiciones, Athena ejecuta la consulta sin utilizar los resultados almacenados en caché.

## Consideraciones y limitaciones

Cuando utilice la función de reutilización de resultados de las consultas, tenga en cuenta los siguientes puntos:

- Athena reutiliza los resultados de las consultas solo dentro del mismo grupo de trabajo.
- La característica de reutilización de los resultados de las consultas respeta las configuraciones de los grupos de trabajo. Si anula la configuración de resultados de una consulta, la característica se deshabilita.
- Se admiten las tablas de Apache Hive, Apache Hudi, Apache Iceberg y Linux Foundation Delta Lake registradas con AWS Glue. No se admiten los almacenes de metadatos externos de Hive.
- No se admiten las consultas que hagan referencia a catálogos federados o a un metaalmacén de Hive externo.
- No se admite la reutilización de los resultados de las consultas en las tablas controladas de Lake Formation.
- No se admite la reutilización de los resultados de las consultas cuando la ubicación de Amazon S3 del origen de la tabla se ha registrado como una ubicación de datos en Lake Formation.
- No se admiten tablas con permisos de fila y columna.
- No se admiten las tablas que tienen un control de acceso detallado (por ejemplo, filtrado de columnas o filas).
- Las consultas que hagan referencia a una tabla no admitida no son aptas para la reutilización de los resultados de las consultas.
- Athena requiere que disponga de permisos de lectura de Amazon S3 para poder volver a utilizar el archivo de salida generado anteriormente.
- La característica de reutilización de los resultados de las consultas asume que el contenido del resultado anterior no se ha modificado. Athena no comprueba la integridad de un resultado anterior antes de usarlo.
- Si los resultados de las consultas de la ejecución anterior se eliminaron o se movieron a una ubicación diferente en Amazon S3, la ejecución posterior de las mismas consultas no reutilizará los resultados de las consultas.
- Se pueden devolver resultados potencialmente obsoletos. Athena no comprueba si hay cambios en los datos de origen hasta que se haya alcanzado la antigüedad máxima de reutilización que usted especifique.
- Si hay varios resultados disponibles para su reutilización, Athena utiliza el resultado más reciente.

- Las consultas que utilizan operadores no deterministas o funciones como `rand()` o `shuffle()` no utilizan resultados almacenados en caché. Por ejemplo, `LIMIT` sin `ORDER BY` no es determinista y no se almacena en caché, pero `LIMIT` con `ORDER BY` es determinista y se almacena en caché.
- La reutilización de los resultados de las consultas se admite en la consola de Athena, en la API de Athena y en el controlador JDBC. Actualmente, la compatibilidad con el controlador ODBC para la reutilización de los resultados de las consultas solo está disponible para Windows.
- Para utilizar la función de reutilización de resultados de consultas con JDBC, la versión del controlador mínima requerida es la 2.0.34.1000. Para ODBC, la versión del controlador mínima requerida es la 1.1.19.1002. Para obtener información acerca de la descarga de los controladores, consulte [Conexión a Amazon Athena con controladores ODBC y JDBC](#).
- No se admite la reutilización de los resultados de las consultas en las consultas que utilicen más de un catálogo de datos.
- No se admite la reutilización de los resultados de las consultas en las consultas que incluyan más de 20 tablas.

## Reutilización de los resultados de las consultas en la consola de Athena

Para utilizar la característica, active la opción Reutilizar los resultados de las consultas en el editor de consultas de Athena.

Query 1

```
1 SELECT * FROM mytable
```

SQL Ln 1, Col 22

Run Explain Cancel Save Clear Create

Reuse query results  
up to 60 minutes ago

Query results | Query stats

Results (0) Copy Download results

Search rows

No results  
Run a query to view results

## Configurar la característica de reutilización de los resultados de las consultas

1. En el editor de consultas de Athena, en la opción Reutilizar los resultados de las consultas, seleccione el icono de edición situado junto a hace menos de 60 minutos.
2. En el cuadro de diálogo Editar tiempo de reutilización, en el cuadro de la derecha, elija una unidad de tiempo (minutos, horas o días).
3. En el cuadro de la izquierda, introduzca o elija el número de unidades de tiempo que desee especificar. El tiempo máximo que puede introducir equivale a siete días, independientemente de la unidad de tiempo elegida.



## Edit reuse time ✕

**Maximum age of reused query results**  
Athena will return the most recent results available within this time frame.

↕  ▼

Minimum: 1 minute, Maximum: 10080 minutes.

**Cancel** **Confirm**

El siguiente ejemplo especifica un tiempo de reutilización máximo de dos días.

## Edit reuse time ✕

**Maximum age of reused query results**  
Athena will return the most recent results available within this time frame.

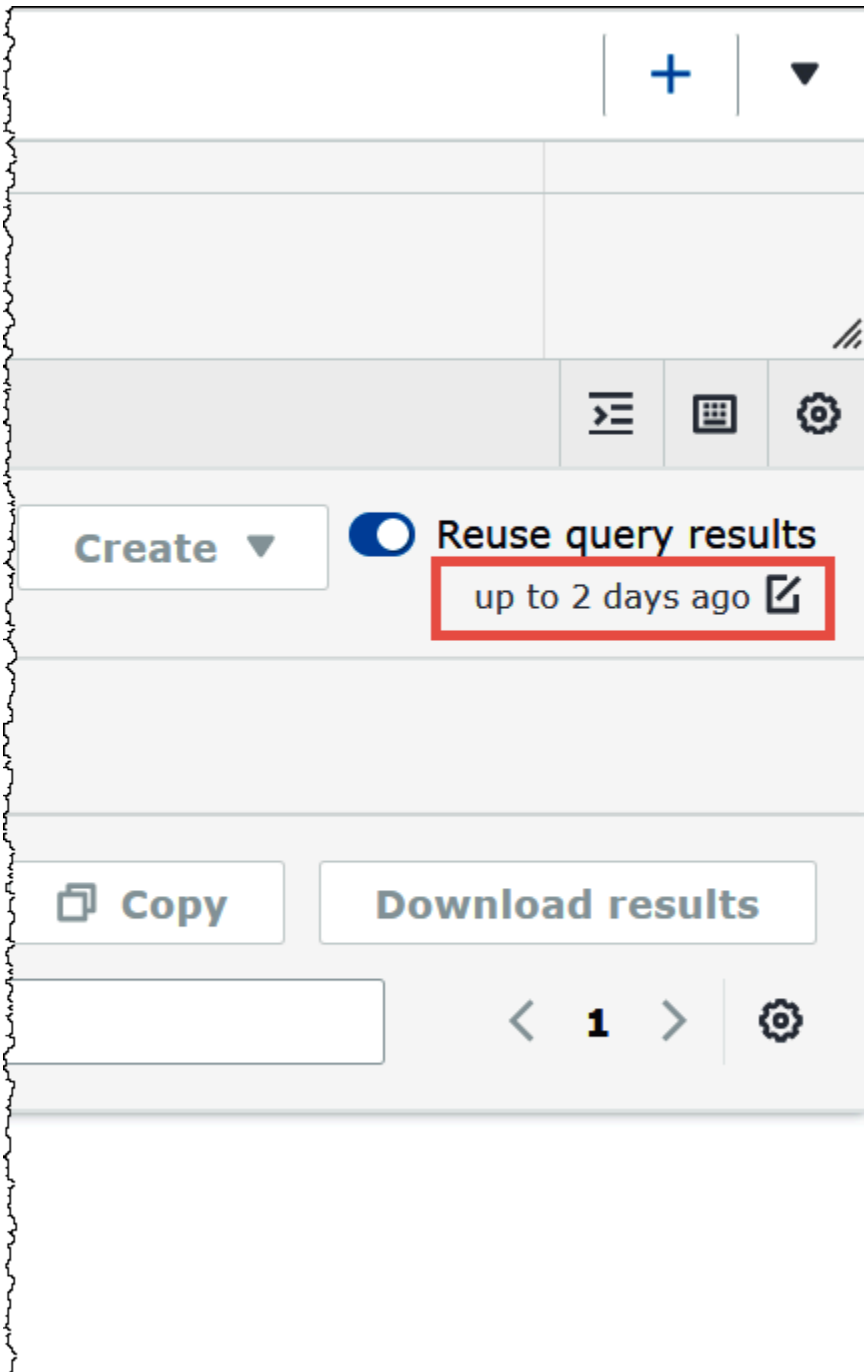
↕  ▼

Minimum: 1 day, Maximum: 7 days.

**Cancel** **Confirm**

4. Elija Confirmar.

Un banner confirma el cambio de configuración y la opción Reutilizar los resultados de las consultas muestra la nueva configuración.



## Visualización de estadísticas y detalles de ejecución de consultas completadas

Después de ejecutar una consulta, puede obtener estadísticas sobre los datos de entrada y salida procesados, ver una representación gráfica del tiempo necesario para cada fase de la consulta y explorar de forma interactiva los detalles de la ejecución.

## Cómo ver las estadísticas de consultas de una consulta completada

1. Después de ejecutar una consulta en el editor de consultas de Athena, elija la pestaña Estadísticas de consulta.

1 `SELECT * FROM "sampledb"."elb_logs" limit 10;`

SQL Ln 1, Col 46

**Run again** [Explain](#) [Cancel](#) [Save](#) [Clear](#) [Create](#)

Query results **Query stats**

### Data processed

Input rows	Input bytes	Output rows	Output bytes
26.43 K	9.00 MB	10	3.41 KB

### Total runtime - 1.4 seconds

[Execution details](#)

0 0.2 0.4 0.6 0.8 1 1.2 1.4

seconds

■ Queuing 17% ■ Planning 19% ■ Execution 58% ■ Service processing 6%

La pestaña Estadísticas de consulta proporciona la siguiente información:

- Datos procesados: muestra el número de filas y bytes de entrada procesados y el número de filas y bytes de salida.
- El tiempo de ejecución total: muestra la cantidad total de tiempo que la consulta tardó en ejecutarse y una representación gráfica de qué parte de ese tiempo se dedicó a la puesta en cola, la planificación, la ejecución y el procesamiento del servicio.

**Note**

La información sobre el recuento de filas y el tamaño de los datos de entrada y salida de nivel de etapa no se muestran cuando una consulta tiene filtros de nivel de fila definidos en Lake Formation.

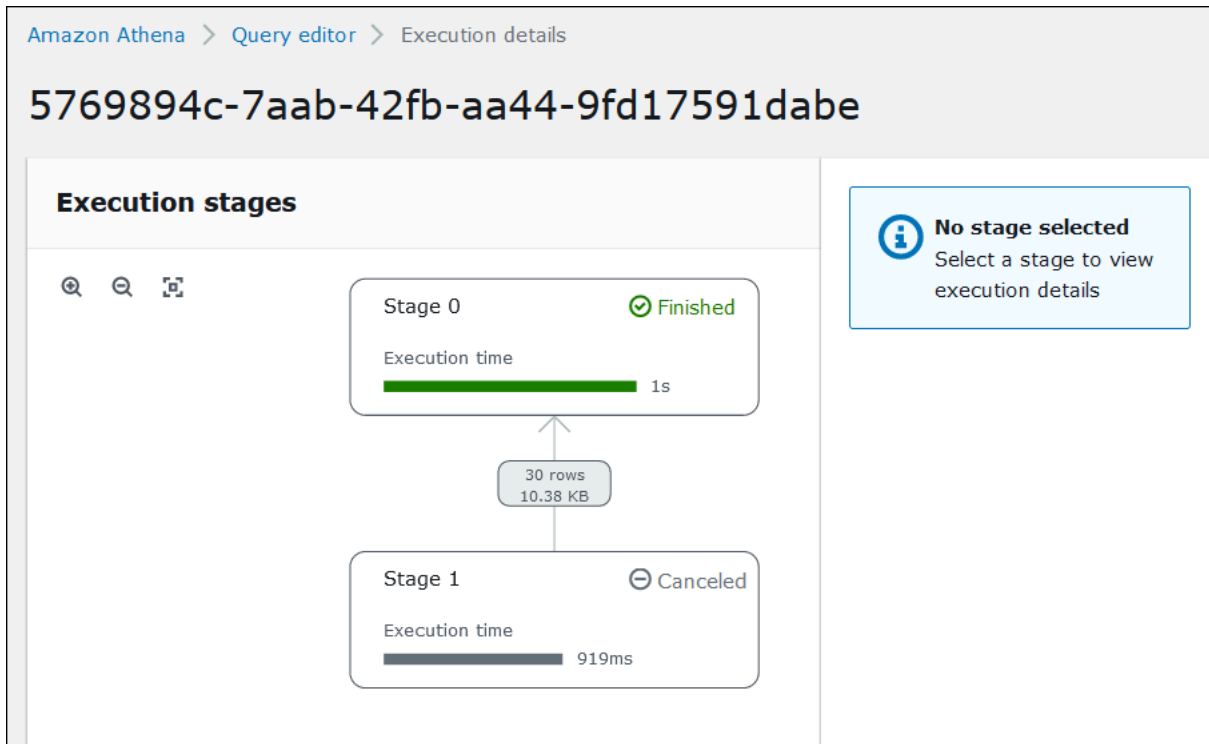
- Para explorar de forma interactiva información acerca de cómo se ejecutó la consulta, elija Detalles de ejecución.



La página Detalles de ejecución muestra el ID de ejecución de la consulta y un gráfico de las etapas de base cero de la consulta. Las etapas se ordenan de principio a fin de abajo hacia arriba. La etiqueta de cada etapa muestra la cantidad de tiempo que tardó la etapa en ejecutarse.

**Note**

El tiempo de ejecución total y el tiempo de ejecución de cada etapa de una consulta suelen variar de manera considerable. Por ejemplo, una consulta con un tiempo de ejecución total en minutos puede mostrar el tiempo de ejecución de una etapa en horas. Dado que una etapa es una unidad lógica de cómputo ejecutada en paralelo en muchas tareas, el tiempo de ejecución de una etapa es el tiempo de ejecución agregado de todas las tareas. A pesar de esta discrepancia, el tiempo de ejecución de las etapas puede ser útil como indicador relativo de qué etapa de una consulta fue más intensiva desde el punto de vista computacional.



Para navegar por el gráfico, utilice las siguientes opciones:

- Para ampliar o reducir la imagen, haga girar el ratón o utilice los iconos de la lupa.
  - Para ajustar el gráfico y que se ajuste a la pantalla, utilice el icono Aumentar para ajustar.
  - Para mover el gráfico, arrastre el puntero del ratón.
3. Para ver más detalles de una etapa, elija la etapa. El panel de detalles de etapa situado en la parte derecha muestra el número de filas y bytes de entrada y salida, y un árbol de operadores.

The screenshot displays the 'Execution stages' panel on the left and the 'Stage 0' details panel on the right. In the 'Execution stages' panel, Stage 0 is shown as 'Finished' with a green checkmark and a progress bar indicating 1s execution time. Stage 1 is shown as 'Canceled' with a gray minus sign and a progress bar indicating 919ms execution time. An arrow points from Stage 1 to Stage 0, with a box above it indicating '30 rows' and '10.38 KB'. The 'Stage 0' details panel shows the status as 'Finished' with a green checkmark. It lists 'Input rows' as 10 and 'Input bytes' as 3.31 KB, and 'Output rows' as 10 and 'Output bytes' as 3.31 KB. The 'Execution time' is 1.3 sec. Under 'Operators', there is a link to 'Expand all'. The 'Output' section is expanded, showing a list of fields: [request\_timestamp, elb\_name, backend\_port, request\_processing\_time, client\_response\_time, elb\_response\_time, received\_bytes, sent\_bytes, received\_bytes\_sent, ssl\_cipher, ssl\_protocol]. A red box highlights the expand icon in the top right corner of the 'Stage 0' details panel.

4. Para ver los detalles de la etapa en todo su ancho, elija el icono de ampliar en la parte superior derecha del panel de detalles.
5. Para obtener información sobre las partes del escenario, amplíe uno o más elementos en el árbol de operadores.

### Stage 0

Status  
✔ Finished

Input rows	Input bytes
10	3.31 KB
Output rows	Output bytes
10	3.31 KB

Execution time  
1.3 sec

Operators  
[Collapse all](#)

```
graph BT; RemoteSource[RemoteSource [1]] --> LocalExchange[LocalExchange [SINGLE] ()]; LocalExchange --> Limit[Limit [10]]; Limit --> Output[Output [request_timestamp, elb_name, request_ip, request_port, backend_ip, backend_port, request_processing_time, backend_processing_time, client_response_time, elb_response_code, backend_response_code, received_bytes, sent_bytes, request_verb, url, protocol, user_agent, ssl_cipher, ssl_protocol]];
```

The diagram shows a sequence of operators: RemoteSource [1] feeds into LocalExchange [SINGLE] (), which feeds into Limit [10], which finally feeds into Output. The Output operator lists the following fields: request\_timestamp, elb\_name, request\_ip, request\_port, backend\_ip, backend\_port, request\_processing\_time, backend\_processing\_time, client\_response\_time, elb\_response\_code, backend\_response\_code, received\_bytes, sent\_bytes, request\_verb, url, protocol, user\_agent, ssl\_cipher, and ssl\_protocol.

Para obtener más información acerca de los detalles de ejecución, consulte [Descripción de los resultados de la instrucción EXPLAIN de Athena](#).

## Recursos adicionales de

Para obtener más información, consulte los siguientes recursos.

[Ver planes de ejecución para consultas SQL](#)

[Uso de EXPLAIN y EXPLAIN ANALYZE en Athena](#)

## Uso de vistas

En Amazon Athena, una vista es una tabla lógica, no física. La consulta que define una vista se ejecuta cada vez que se hace referencia a la vista en una consulta.

Puede crear una vista de una consulta SELECT y, a continuación, hacer referencia a dicha vista en futuras consultas. Para obtener más información, consulte [CREATE VIEW](#).

### Temas

- [¿Cuándo debería utilizar vistas?](#)
- [Acciones compatibles con las vistas en Athena](#)
- [Consideraciones sobre las vistas](#)
- [Limitaciones de las vistas](#)
- [Uso de vistas en la consola](#)
- [Creación de vistas](#)
- [Ejemplos de vistas](#)
- [Uso de vistas del AWS Glue Data Catalog](#)

### ¿Cuándo debería utilizar vistas?

Puede crear vistas para lo siguiente:

- Consultar un subconjunto de datos. Por ejemplo, puede crear una vista con un subconjunto de columnas a partir de la tabla original para simplificar la consulta de datos.
- Combinar varias tablas en una consulta. Si tiene varias tablas y desea combinarlas con UNION ALL, puede crear una vista con esa expresión para simplificar las consultas en las tablas combinadas.
- Ocultar la complejidad de las consultas de base existentes y simplificar las consultas que ejecutan los usuarios. A menudo, las consultas de base incluyen combinaciones de tablas, expresiones en la lista de columnas y otra sintaxis SQL que dificultan su comprensión y depuración. Puede crear una vista que oculte la complejidad y simplifique las consultas.
- Experimentar con técnicas de optimización y crear consultas optimizadas. Por ejemplo, si encuentra una combinación de condiciones WHERE, orden JOIN u otras expresiones que consigan un mejor rendimiento, puede crear una vista con dichas cláusulas y expresiones. De este modo, las aplicaciones podrán realizar consultas relativamente sencillas a esta vista. Si posteriormente



encuentra una forma mejor de optimizar la consulta original, cuando vuelva a crear la vista, todas las aplicaciones aprovecharán inmediatamente la consulta de base optimizada.

- Ocultar la tabla subyacente y los nombres de las columnas, y minimizar los problemas de mantenimiento si dichos nombres cambian. En ese caso, vuelva a crear la vista con los nuevos nombres. Todas las consultas que usan la vista en lugar de las tablas subyacentes seguirán ejecutándose sin cambios.

## Acciones compatibles con las vistas en Athena

Athena admite las siguientes acciones con vistas. Puede ejecutar estos comandos en el editor de consultas.

Instrucción	Descripción
<a href="#"><u>CREATE VIEW</u></a>	<p>Crea una nueva vista de una consulta SELECT especificada. Para obtener más información, consulte <a href="#">Creación de vistas</a>.</p> <p>La cláusula opcional OR REPLACE le permite actualizar la vista existente sustituyéndola.</p>
<a href="#"><u>DESCRIBE VIEW</u></a>	Muestra la lista de columnas de la vista citada. Esto le permite examinar los atributos de una vista compleja.
<a href="#"><u>DROP VIEW</u></a>	Elimina una vista existente. La cláusula opcional IF EXISTS elimina el error si la vista no existe.
<a href="#"><u>SHOW CREATE VIEW</u></a>	Muestra la instrucción SQL que crea la vista especificada.
<a href="#"><u>SHOW VIEWS</u></a>	Muestra todas las vistas de la base de datos especificada o de la base de datos actual si se omite el nombre de la base de datos. Utilice la cláusula opcional LIKE con una expresión regular para limitar la lista de nombres de vista. También puede ver la lista de vistas en el panel izquierdo de la consola.
<a href="#"><u>SHOW COLUMNS</u></a>	Muestra las columnas del esquema de una vista.

## Consideraciones sobre las vistas

Las siguientes consideraciones se aplican a la creación y el uso de vistas en Athena:

- En Athena, puede obtener una vista previa y trabajar con vistas creadas en la consola de Athena, en AWS Glue Data Catalog, si ha migrado para utilizarlo, o con Presto en el clúster de Amazon EMR conectado al mismo catálogo. No se puede obtener vistas previas ni agregar a vistas de Athena creadas de otras formas.
- Si va a crear vistas a través del catálogo de datos de AWS Glue, debe incluir el parámetro `PartitionKeys` y establecer su valor en una lista vacía, tal como se indica a continuación: `"PartitionKeys": []`. De lo contrario, la consulta de la vista devolverá un error en Athena. El siguiente ejemplo muestra una vista creada desde el catálogo de datos con `"PartitionKeys": []`:

```
aws glue create-table
--database-name mydb
--table-input '{
"Name":"test",
"TableType": "EXTERNAL_TABLE",
"Owner": "hadoop",
"StorageDescriptor":{
"Columns":[{
"Name":"a","Type":"string"},{"Name":"b","Type":"string"}],
"Location":"s3://DOC-EXAMPLE-BUCKET/Oct2018/250ct2018/",
"InputFormat":"org.apache.hadoop.mapred.TextInputFormat",
"OutputFormat": "org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat",
"SerdeInfo":{"SerializationLibrary":"org.apache.hadoop.hive.serde2.OpenCSVSerde",
"Parameters":{"separatorChar": "|", "serialization.format":
"1"}}}, "PartitionKeys":[]}'
```

- Si ha creado vistas Athena en el catálogo de datos, este trata las vistas como tablas. Puede utilizar el control de acceso detallado de nivel de tablas en el catálogo de datos para [restringir el acceso](#) a estas vistas.
- Athena impide ejecutar vistas recursivas y muestra un mensaje de error en tales casos. Una vista recursiva es una consulta de vistas que hace referencia a sí misma.
- Athena muestra un mensaje de error cuando detecta vistas obsoletas. Se registra una vista obsoleta cuando se produce alguna de las siguientes situaciones:
  - La vista hace referencia a tablas o bases de datos que no existen.
  - Se realiza un cambio de esquema o metadatos en una tabla referenciada.

- Una tabla referenciada se elimina y se vuelve a crear con un esquema o configuración diferente.
- Puede crear y ejecutar vistas anidadas siempre y cuando la consulta detrás de la vista anidada sea válida y las tablas y las bases de datos existan.

## Limitaciones de las vistas

- Los nombres de vistas de Athena no pueden contener caracteres especiales distintos del guion bajo (\_). Para obtener más información, consulte [Nombres de tablas, bases de datos y columnas](#).
- Evite utilizar palabras clave reservadas para asignar nombres a las vistas. Si utiliza palabras clave reservadas, use comillas dobles para destacar las palabras clave reservadas en las consultas de vistas. Consulte [Palabras clave reservadas](#).
- No se pueden utilizar vistas creadas en Athena con almacenes de metadatos externos de Hive ni UDF. Para obtener información sobre el uso de vistas creadas de forma externa en Hive, consulte [Trabajo con vistas de Hive](#).
- No puede utilizar vistas con funciones geoespaciales.
- No se pueden utilizar vistas para administrar el control de acceso a los datos en Amazon S3. Para consultar una vista, necesita permisos de acceso a los datos almacenados en Amazon S3. Para obtener más información, consulte [Acceso a Amazon S3](#).
- Si bien tanto la versión 2 como la 3 del motor de Athena admiten la consulta de vistas entre cuentas, no se puede crear una vista que incluya un AWS Glue Data Catalog entre cuentas. Para obtener información sobre el acceso a catálogos de datos entre cuentas, consulte [Acceso entre cuentas a los catálogos de datos de AWS Glue](#).
- Las columnas de metadatos ocultas de Hive o Iceberg \$bucket, \$file\_modified\_time, \$file\_size y \$partition no se admiten en las vistas de Athena. Para obtener información sobre el uso de la columna de metadatos \$path en Athena, consulte [Obtención de las ubicaciones de archivos para los datos de origen en Simple Storage Service \(Amazon S3\)](#).

## Uso de vistas en la consola

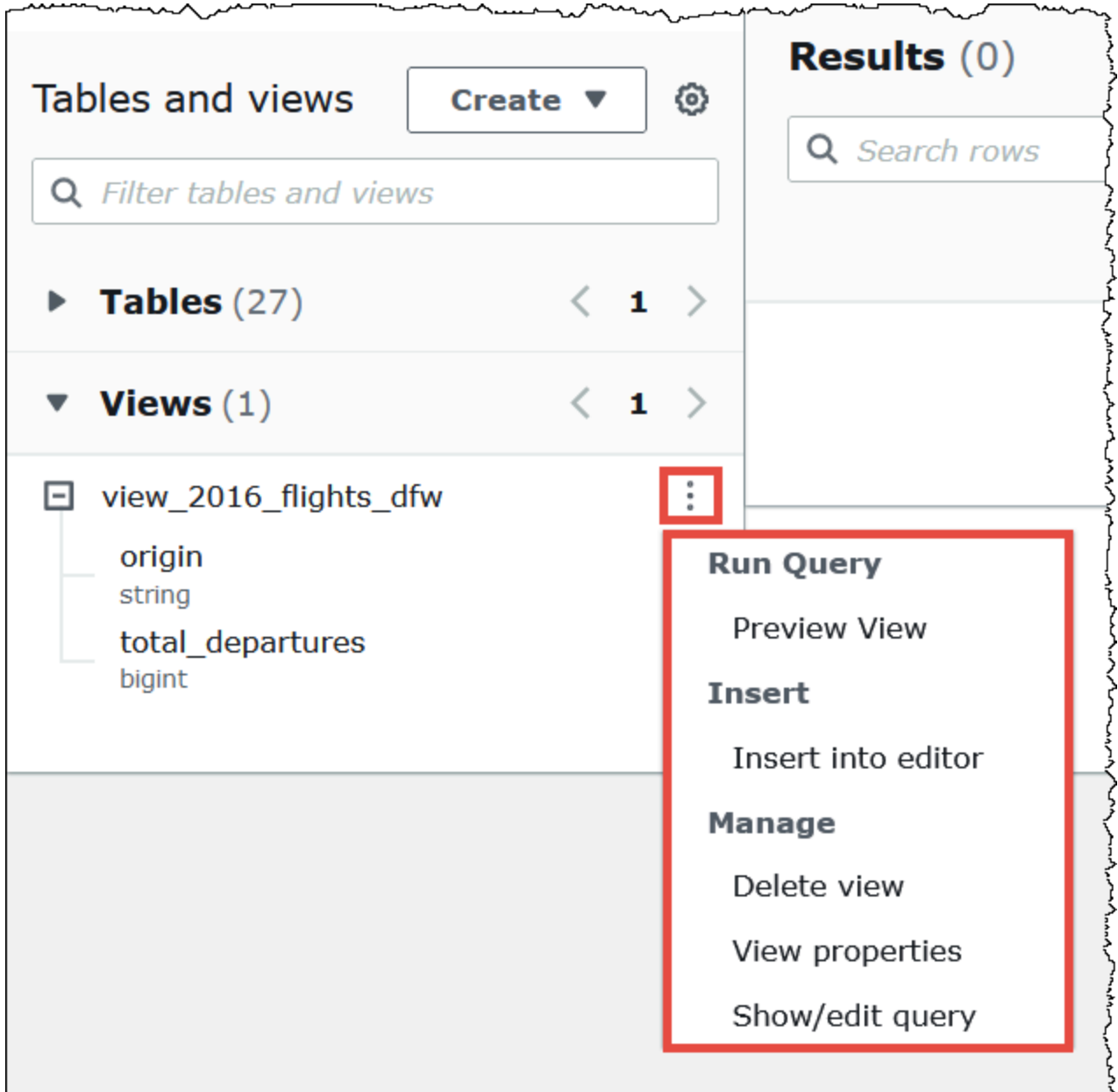
En la consola de Athena puede hacer lo siguiente:

- Ver todas las vistas en el panel izquierdo, donde se enumeran las tablas.
- Filtrar vistas.
- Obtener una vista previa de una vista, mostrar sus propiedades, editarla o eliminarla.

## Para mostrar las acciones de una vista

Las vistas se muestran en la consola únicamente si ya las ha creado.

1. En la consola de Athena, elija Vistas y, luego, elija una vista para expandirla y mostrar las columnas de la vista.
2. Haga clic en los tres puntos verticales junto a la vista para mostrar una lista de acciones.



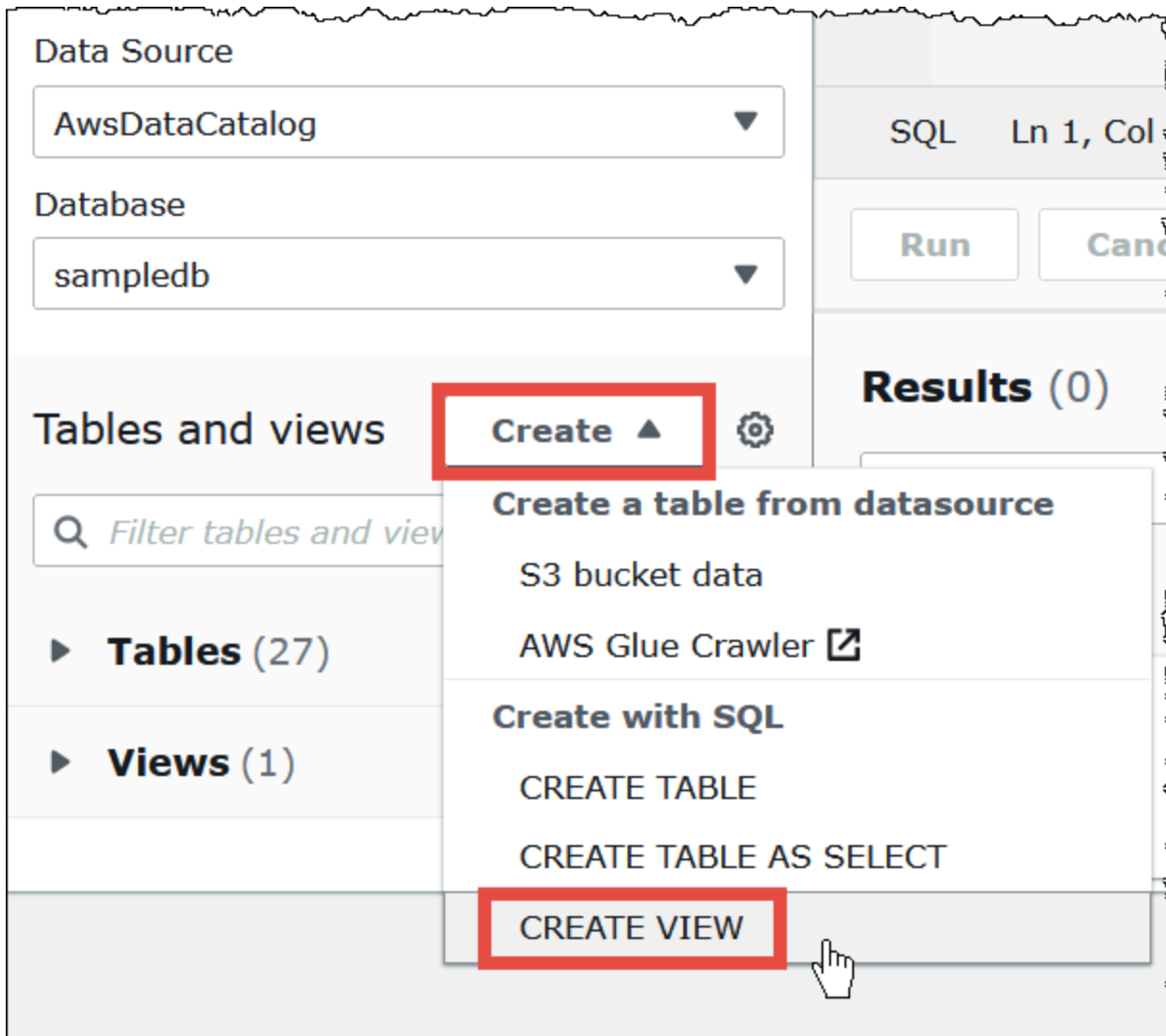
3. Abra este menú para obtener una vista previa de la vista, insertar el nombre de la vista en el editor de consultas, eliminar la vista, ver las propiedades de la vista o mostrar y editar la vista en el editor de consultas.

## Creación de vistas

Puede crear una vista en la consola de Athena mediante una plantilla o bien ejecutar una consulta existente.

Para usar una plantilla para crear una vista

1. En la consola de Athena, junto a Tablas y vistas, elija Crear y, a continuación, elija Crear vista.



Esta acción coloca una plantilla de vista editable en el editor de consultas.

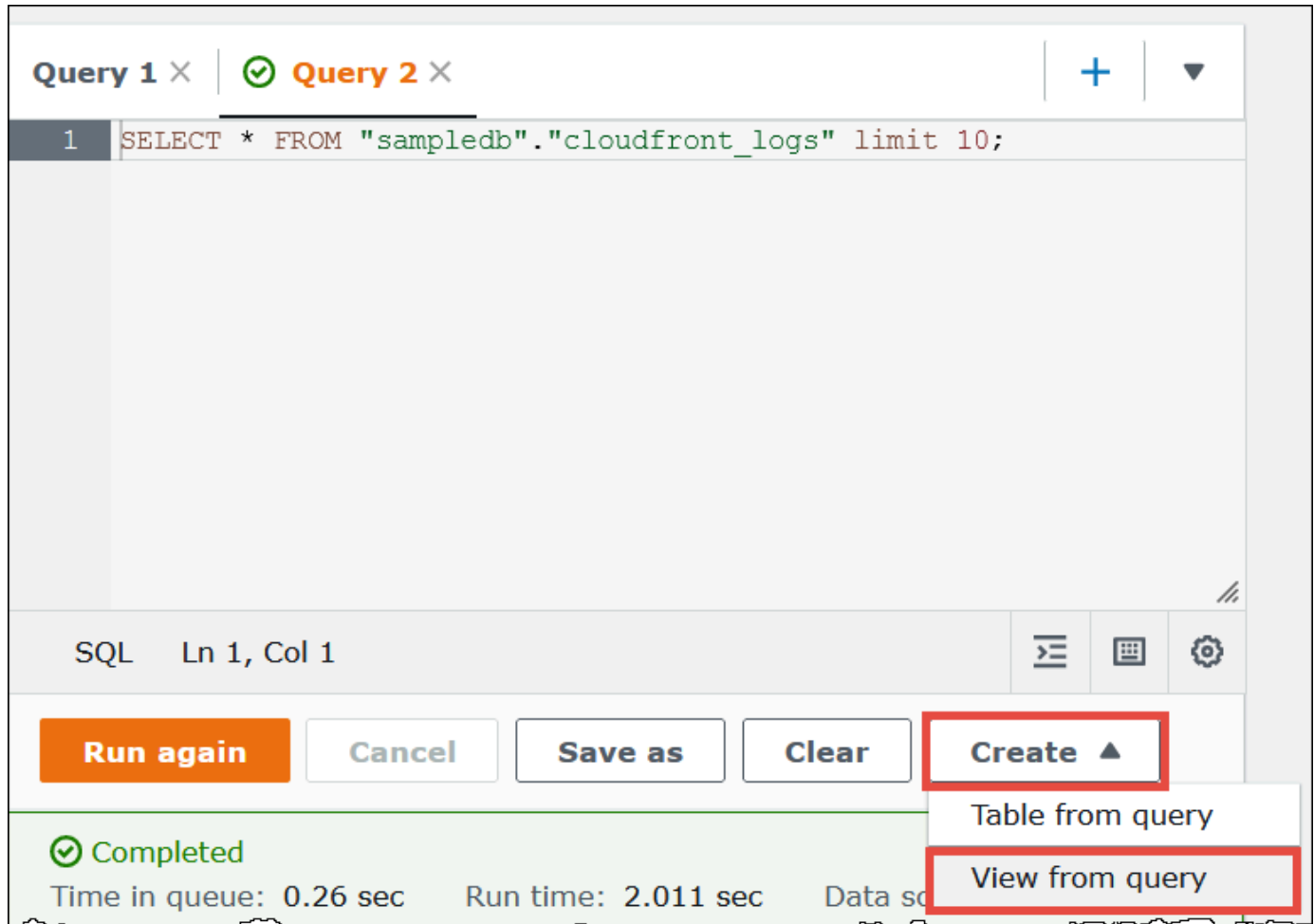
2. Edite la plantilla de vista según sus necesidades. Cuando ingrese un nombre para la vista en la instrucción, recuerde que los nombres de las vistas no pueden contener caracteres especiales que no sean guiones bajos (\_). Consulte [Nombres de tablas, bases de datos y columnas](#). Evite utilizar [Palabras clave reservadas](#) para asignar nombres a las vistas.

Para obtener más información sobre la creación de vistas, consulte [CREATE VIEW](#) y [Ejemplos de vistas](#).

3. Elija Ejecutar para crear la vista. La vista aparece en la lista de vistas de la consola de Athena.

Para crear una vista a partir de una consulta existente

1. Use el editor de consultas de Athena para ejecutar una consulta existente.
2. En la ventana del editor de consultas, elija Crear y, a continuación, Vista a partir de la consulta.



3. En el cuadro de diálogo Crear vista, ingrese un nombre para la vista y, luego, elija Crear. Los nombres de vistas no pueden contener caracteres especiales distintos del guion bajo (\_). Consulte [Nombres de tablas, bases de datos y columnas](#). Evite utilizar [Palabras clave reservadas](#) para asignar nombres a las vistas.

Athena agrega la vista a la lista de vistas en la consola y muestra la instrucción CREATE VIEW para la vista en el editor de consultas.

## Notas

- Si elimina una tabla en la que se basa la vista y luego intenta ejecutar la vista, Athena muestra un mensaje de error.
- Puede crear una vista anidada, que es una vista sobre una vista existente. Athena le impide ejecutar una vista recursiva que hace referencia a sí misma.

## Ejemplos de vistas

Para mostrar la sintaxis de la consulta de vista, utilice [SHOW CREATE VIEW](#).

### Example Ejemplo 1

Tenga en cuenta las siguientes dos tablas: una tabla `employees` con dos columnas, `id` y `name`, y una tabla `salaries`, con dos columnas, `id` y `salary`.

En este ejemplo, creamos una vista llamada `name_salary` como una consulta `SELECT` que obtiene una lista de ID mapeada con salarios de las tablas `employees` y `salaries`:

```
CREATE VIEW name_salary AS
SELECT
  employees.name,
  salaries.salary
FROM employees, salaries
WHERE employees.id = salaries.id
```

### Example Ejemplo 2

En el siguiente ejemplo, creamos una vista llamada `view1` que le permite ocultar la sintaxis de consultas más complejas.

Esta vista se ejecuta en dos tablas, `table1` y `table2`, donde cada tabla es una consulta `SELECT` distinta. La vista selecciona las columnas de `table1` y une los resultados con `table2`. Esta unión se basa en la columna `a` que está presente en ambas tablas.

```
CREATE VIEW view1 AS
WITH
  table1 AS (
    SELECT a,
```

```
        MAX(b) AS the_max
    FROM x
    GROUP BY a
),
table2 AS (
    SELECT a,
    AVG(d) AS the_avg
    FROM y
    GROUP BY a)
SELECT table1.a, table1.the_max, table2.the_avg
FROM table1
JOIN table2
ON table1.a = table2.a;
```

Para obtener información sobre cómo consultar vistas federadas, vea [Consulta de vistas federadas](#).

## Uso de vistas del AWS Glue Data Catalog

Esta característica está en versión preliminar y está sujeta a cambios. Para obtener más información, consulte la sección Betas y versiones preliminares del documento [Términos de servicio de AWS](#).

Utilice las vistas del AWS Glue Data Catalog cuando desee disponer de una única vista común entre los Servicios de AWS como Amazon Athena y Amazon Redshift. En las vistas del catálogo de datos, los permisos de acceso los define el usuario que creó la vista y no el usuario que consulta la vista. Este método de concesión de permisos se denomina semántica de definidor.

Los siguientes casos de uso muestran cómo puede usar las vistas del catálogo de datos.

- **Mayor control de acceso:** se crea una vista que restringe el acceso a los datos en función del nivel de permisos que requiera el usuario. Por ejemplo, puede usar las vistas del catálogo de datos para evitar que los empleados que no trabajan en el departamento de Recursos Humanos (RRHH) vean información de identificación personal.
- **Garantía de registros completos:** al aplicar determinados filtros a la vista del catálogo de datos, se asegura de que los registros de datos de una vista del catálogo de datos estén siempre completos.
- **Seguridad mejorada:** en las vistas del catálogo de datos, la definición de consulta que crea la vista debe estar intacta para poder crearla. Esto hace que las vistas del catálogo de datos sean menos susceptibles a los comandos SQL de actores malintencionados.



- **Impedimento de acceso a las tablas subyacentes:** la semántica del definidor permite a los usuarios acceder a una vista sin poner la tabla subyacente a su disposición. Solo el usuario que define la vista necesita acceder a las tablas.

Las definiciones de las vistas del catálogo de datos se almacenan en el AWS Glue Data Catalog. Esto significa que puede utilizar la AWS Lake Formation para conceder el acceso mediante concesiones de recursos, concesiones de columnas o controles de acceso basados en etiquetas. Para obtener más información sobre cómo conceder y revocar el acceso a Lake Formation, consulte [Concesión y revocación de permisos en los recursos del catálogo de datos](#) en la Guía para desarrolladores de AWS Lake Formation.

## Permisos

Las vistas del catálogo de datos requieren tres roles de Lake Formation: `Admin`, `Definer` y `Invoker`.

- **Lake Formation Admin:** tiene acceso para configurar todos los permisos de Lake Formation.
- **Definer:** crea la vista del catálogo de datos. El rol de `Definer` debe tener todos los permisos otorgables `SELECT` en todas las tablas subyacentes a las que hace referencia la definición de la vista.
- **Invoker:** puede consultar la vista del catálogo de datos o comprobar sus metadatos.

Las relaciones de confianza del rol de `Definer` deben permitir la acción `sts:AssumeRole` respecto de las entidades de servicios principales de AWS Glue y de Lake Formation, como se muestra en el siguiente ejemplo.

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

```
    }  
  ]  
}
```

También se requieren permisos de IAM para el acceso a Athena. Para obtener más información, consulte [Políticas administradas de AWS para Amazon Athena](#).

## Limitaciones

- Las vistas del catálogo de datos no pueden hacer referencia a otras vistas.
- Puede hacer referencia a un máximo de 10 tablas en la definición de vista.
- Las tablas subyacentes deben registrarse en Lake Formation.
- La entidad principal de DEFINER solo puede ser un rol de IAM.
- El rol de DEFINER debe tener todos los permisos (otorgables) SELECT en las tablas subyacentes.
- No se admiten las vistas del catálogo de datos UNPROTECTED.
- No se admiten en la definición de vista las funciones definidas por el usuario (UDF).
- Los orígenes de datos federados de Athena no se pueden utilizar en las vistas del catálogo de datos.
- Las vistas del catálogo de datos no son compatibles con los metaalmacenes externos de Hive.
- Athena muestra un mensaje de error cuando detecta vistas obsoletas. Se registra una vista obsoleta cuando se produce alguna de las siguientes situaciones:
  - La vista hace referencia a tablas o bases de datos que no existen.
  - Se realiza un cambio de esquema o metadatos en una tabla referenciada.
  - Una tabla referenciada se elimina y se vuelve a crear con un esquema o configuración diferente.

## Creación de una vista del catálogo de datos

El siguiente ejemplo de sintaxis muestra cómo un usuario del rol de Definer crea la vista del catálogo de datos de `orders_by_date`. En el ejemplo se supone que el rol de Definer tiene todos los permisos SELECT en la tabla `orders` de la base de datos `default`.

```
CREATE PROTECTED MULTI DIALECT VIEW orders_by_date  
SECURITY DEFINER  
AS  
SELECT orderdate, sum(totalprice) AS price
```

```
FROM orders
WHERE order_city = 'SEATTLE'
GROUP BY orderdate
```

## Consulta de la vista del catálogo de datos

Una vez creada la vista, el Admin de Lake Formation puede conceder permisos SELECT en la vista del catálogo de datos a las entidades principales del Invoker. A continuación, las entidades principales del Invoker pueden consultar la vista sin tener acceso a las tablas base subyacentes a las que hace referencia la vista. A continuación se muestra un ejemplo de consulta del Invoker.

```
SELECT * from orders_by_date where price > 5000
```

## Actualización de una vista del catálogo de datos

El Lake Formation Admin o el Definer pueden usar la sintaxis ALTER VIEW UPDATE DIALECT para actualizar la definición de la vista. El siguiente ejemplo modifica la definición de la vista para seleccionar columnas de la tabla returns en lugar de la tabla orders.

```
ALTER VIEW orders_by_date UPDATE DIALECT
AS
SELECT return_date, sum(totalprice) AS price
FROM returns
WHERE order_city = 'SEATTLE'
GROUP BY orderdate
```

Para obtener más información sobre la sintaxis para crear y administrar vistas del catálogo de datos, consulte [Sintaxis de vista del catálogo de datos de Glue](#).

## Sintaxis de vista del catálogo de datos de Glue

Esta característica está en versión preliminar y está sujeta a cambios. Para obtener más información, consulte la sección Betas y versiones preliminares del documento [Términos de servicio de AWS](#).

En esta sección se describen los comandos del lenguaje de definición de datos (DDL) para crear y administrar vistas del AWS Glue Data Catalog.

## ALTER VIEW DIALECT

Puede actualizar las vistas del catálogo de datos al añadir un dialecto del motor o actualizar o eliminar un dialecto del motor existente. Solo el Admin de Lake Formation y el Definer (el usuario que creó la vista) tienen permiso para usar la instrucción ALTER VIEW DIALECT en una vista del catálogo de datos.

### Sintaxis

```
ALTER VIEW name [ FORCE ] [ ADD|UPDATE ] DIALECT AS query
```

```
ALTER VIEW name [ DROP ] DIALECT
```

### FORCE

La palabra clave FORCE hace que la información entre los dialectos del motor en conflicto de una vista se sobrescriba con la nueva definición. La palabra clave FORCE resulta útil cuando una actualización de una vista del catálogo de datos da como resultado definiciones de vista contradictorias entre los dialectos del motor existentes. Supongamos que una vista del catálogo de datos tiene los dialectos Athena y Amazon Redshift y que la actualización provoca un conflicto con Amazon Redshift en la definición de la vista. En este caso, puede usar la palabra clave FORCE para permitir que se complete la actualización y marcar el dialecto de Amazon Redshift como obsoleto. Cuando los motores marcados como obsoletos consultan la vista, la consulta falla. Los motores hacen una excepción para impedir que se obtengan resultados obsoletos. Para corregir esto, actualice los dialectos obsoletos de la vista.

### ADD

Añade un nuevo dialecto del motor a la vista del catálogo de datos. El motor especificado no debe existir en la vista del catálogo de datos.

### UPDATE

Actualiza un dialecto del motor que ya existe en la vista del catálogo de datos.

### DROP

Elimina un dialecto del motor existente de la vista del catálogo de datos. Tras eliminar un motor de una vista del catálogo de datos, el motor que se descartó no podrá consultar la vista del catálogo de datos. Otros dialectos del motor de la vista aún pueden consultar la vista.

## DIALECT AS

Introduce una consulta SQL específica del motor.

### Ejemplos

```
ALTER VIEW orders_by_date FORCE ADD DIALECT
AS
SELECT orderdate, sum(totalprice) AS price
FROM orders
GROUP BY orderdate
```

```
ALTER VIEW orders_by_date FORCE UPDATE DIALECT
AS
SELECT orderdate, sum(totalprice) AS price
FROM orders
GROUP BY orderdate
```

```
ALTER VIEW orders_by_date DROP DIALECT
```

## CREATE PROTECTED MULTI DIALECT VIEW

Creación de una vista de catálogo de datos en el AWS Glue Data Catalog. Una vista de catálogo de datos es un esquema de vista única que funciona perfectamente en Athena y otros motores de SQL, como Amazon Redshift y Amazon EMR.

### Sintaxis

```
CREATE [ OR REPLACE ] PROTECTED MULTI DIALECT VIEW view_name
[ SECURITY DEFINER ]
AS query
```

## PROTECTED

Palabra clave necesaria. Especifica que la vista está protegida contra las filtraciones de datos. Las vistas de catálogo de datos solo se pueden crear como una vista PROTECTED.

## MULTI DIALECT

Especifica que la vista admite los dialectos SQL de diferentes motores de consulta y, por lo tanto, esos motores pueden leerla.

## SECURITY DEFINER

Especifica que la semántica del definidor está vigente en esta vista. La semántica de los definidores significa que los permisos de lectura en vigencia de las tablas subyacentes pertenecen a la entidad principal o al rol que definió la vista y no a la entidad principal que realiza la lectura real.

## OR REPLACE

No se puede reemplazar una vista del catálogo de datos si en la vista hay dialectos SQL de otros motores. Si el motor de llamada tiene el único dialecto SQL presente en la vista, la vista se puede reemplazar.

## Ejemplo

En el siguiente ejemplo se crea la vista del catálogo de datos de `orders_by_date` en función de una consulta de la tabla `orders`.

```
CREATE PROTECTED MULTI DIALECT VIEW orders_by_date
SECURITY DEFINER
AS
SELECT orderdate, sum(totalprice) AS price
FROM orders
WHERE order_city = 'SEATTLE'
GROUP BY orderdate
```

## DESCRIBE

Muestra la lista de columnas de la vista del catálogo de datos especificada. La instrucción `DESCRIBE` es similar a la instrucción `DESCRIBE` de los puntos de vista de Athena. A diferencia de las vistas de Athena, el resultado de la instrucción se controla mediante el control de acceso de Lake Formation. Por lo tanto, el resultado de esta consulta no son todas las columnas de la vista, sino solo las columnas a las que tiene acceso la persona que llama.

## Sintaxis

```
DESCRIBE [db_name.]view_name
```

## Ejemplos

```
DESCRIBE orders
```

## DROP VIEW

Elimina una vista del catálogo de datos solo si el dialecto del motor de llamada está presente en la vista del catálogo de datos. Por ejemplo, si un usuario llama a `DROP VIEW` desde Athena, la vista solo se elimina si el dialecto de Athena existe en la vista. De lo contrario, la operación no se llevará a cabo correctamente. Solo el administrador de Lake Formation y el definidor de vistas tienen permiso para usar la instrucción `DROP VIEW` en una vista del catálogo de datos.

### Sintaxis

```
DROP VIEW [ IF EXISTS ] view_name
```

### Ejemplos

```
DROP VIEW orders_by_date
```

```
DROP FORCE VIEW IF EXISTS orders_by_date
```

La cláusula `IF EXISTS` opcional provoca la eliminación del error si la vista no existe.

## SHOW COLUMNS

Solo muestra los nombres de las columnas de una única vista del catálogo de datos especificada. La instrucción `SHOW COLUMNS` es similar a la instrucción `SHOW COLUMNS` de los puntos de vista de Athena. A diferencia de las vistas de Athena, el resultado de la instrucción se controla mediante el control de acceso de Lake Formation. Por lo tanto, el resultado de esta consulta no son todas las columnas de la vista, sino solo las columnas a las que tiene acceso la persona que llama.

### Sintaxis

```
SHOW COLUMNS {FROM|IN} database_name.view_name
```

```
SHOW COLUMNS {FROM|IN} view_name [{FROM|IN} database_name]
```

## SHOW CREATE VIEW

Muestra la sintaxis SQL que creó la vista del catálogo de datos. El SQL devuelto muestra la sintaxis de creación de vistas utilizada en Athena. Solo el administrador de Lake Formation y las entidades

principales del definidor de vistas están autorizados a llamar a `SHOW CREATE VIEW` en una vista del catálogo de datos.

## Sintaxis

```
SHOW CREATE VIEW view_name
```

## Ejemplos

```
SHOW CREATE VIEW orders_by_date
```

## SHOW VIEWS

Muestra los nombres de todas las vistas de la base de datos. Se muestran todas las vistas del catálogo de datos de la base de datos que tienen el dialecto SQL del motor Athena. Se excluyen las demás vistas del catálogo de datos que no tienen el dialecto del motor Athena en la vista.

## Sintaxis

```
SHOW VIEWS [IN database_name] [LIKE 'regular_expression']
```

## Ejemplos

```
SHOW VIEWS
```

```
SHOW VIEWS IN marketing_analytics LIKE 'orders*'
```

## Uso de consultas guardadas

Puede utilizar la consola de Athena para guardar, editar, ejecutar, cambiar el nombre y eliminar las consultas creadas en el editor de consultas.

## Consideraciones y limitaciones

- Puede actualizar el nombre, la descripción y el texto de consulta de las consultas guardadas.
- Solo puede actualizar las consultas en su propia cuenta.
- No puede cambiar el grupo de trabajo ni la base de datos a la que pertenece la consulta.



- Athena no guarda un historial de modificaciones de consultas. Si desea conservar una versión concreta de una consulta, guárdela con otro nombre.

## Trabajo con consultas guardadas en la consola de Athena

Para guardar una consulta y ponerle un nombre

1. En el editor de consultas de la consola de Athena, ingrese o ejecute la consulta.
2. Sobre la ventana del editor de consultas, en la pestaña correspondiente a la consulta, seleccione los tres puntos verticales y, a continuación, Save as (Guardar como).
3. En el cuadro de diálogo Save query (Guardar consulta), ingrese un nombre para la consulta y una descripción opcional. Puede utilizar la ventana ampliable Preview SQL query (Vista previa de consultas SQL) para verificar el contenido de la consulta antes de guardarla.
4. Seleccione Save query (Guardar consulta).

En el editor de consultas, la pestaña de la consulta muestra el nombre que ha especificado.

Para ejecutar una consulta guardada

1. En la consola de Athena, elija la pestaña Saved queries (Consultas guardadas).
2. En la lista Saved queries (Consultas guardadas), elija el nombre de la consulta que quiera ejecutar.

El editor de consultas muestra la consulta que eligió.

3. Elija Ejecutar.

Para editar una consulta guardada

1. En la consola de Athena, elija la pestaña Saved queries (Consultas guardadas).
2. En la lista Saved queries (Consultas guardadas), elija la ID de la consulta que quiera editar.
3. En el editor de consultas, edite la consulta.
4. Lleve a cabo uno de los siguientes pasos:
  - Elija Run (Ejecutar) para ejecutar la consulta.
  - Para guardar la consulta, seleccione los tres puntos verticales en la pestaña de la consulta y, a continuación, seleccione Save (Guardar).

- Para guardar la consulta con otro nombre, elija los tres puntos verticales en la pestaña de la consulta y, a continuación, elija Save as (Guardar como).

Para cambiar el nombre de una consulta guardada que ya se muestra en el editor de consultas

1. Elija los tres puntos verticales de la pestaña de la consulta y, a continuación, elija Rename (Cambiar nombre) o Delete (Eliminar).
2. Siga las instrucciones para cambiar el nombre de la consulta o eliminarla.

Para cambiar el nombre de una consulta guardada que no se muestra en el editor de consultas

1. En la consola de Athena, elija la pestaña Saved queries (Consultas guardadas).
2. Seleccione la casilla de verificación de la consulta cuyo nombre quiera cambiar.
3. Elija Rename (Cambiar nombre).
4. En el cuadro de diálogo Rename query (Cambio del nombre de la consulta), edite el nombre y la descripción de la consulta. Puede utilizar la ventana ampliable Preview SQL query (Vista previa de consultas SQL) para verificar el contenido de la consulta antes de cambiarle el nombre.
5. Elija Rename query (Cambio del nombre de la consulta).

La consulta renombrada aparece en la lista Saved queries (Consultas guardadas).

Para eliminar una consulta guardada que no se muestra en el editor de consultas

1. En la consola de Athena, elija la pestaña Saved queries (Consultas guardadas).
2. Seleccione una o varias casillas de verificación correspondientes a las consultas que desee eliminar.
3. Elija Eliminar.
4. En el mensaje de confirmación, elija Delete (Eliminar).

Una o más consultas se eliminan de la lista Saved queries (Consultas guardadas).

## Uso de la API de Athena para actualizar las consultas guardadas

Para obtener información sobre cómo utilizar la API de Athena para actualizar una consulta guardada, consulte la acción [UpdateNamedQuery](#) en la Referencia de API de Athena.

## Uso de consultas parametrizadas

Puede usar consultas parametrizadas de Athena para volver a ejecutar la misma consulta con valores de parámetros diferentes en el momento de la ejecución y ayudar a evitar ataques de inyección de código SQL. En Athena, las consultas parametrizadas pueden adoptar la forma de parámetros de ejecución en cualquier consulta DML o instrucciones preparadas de SQL.

- Las consultas con parámetros de ejecución se pueden realizar en un solo paso y no son específicas del grupo de trabajo. Coloque signos de interrogación en cualquier consulta de DML para los valores que desea parametrizar. Al ejecutar la consulta, declara los valores de los parámetros de ejecución de manera secuencial. La declaración de parámetros y la asignación de valores para los parámetros se pueden hacer en la misma consulta, pero de manera desacoplada. A diferencia de las instrucciones preparadas, puede seleccionar el grupo de trabajo cuando envía una consulta con parámetros de ejecución.
- Las instrucciones preparadas requieren dos instrucciones SQL separadas: PREPARE y EXECUTE. Primero, defina los parámetros en la instrucción PREPARE. A continuación, ejecute una declaración EXECUTE que proporcione los valores de los parámetros que ha definido. Las instrucciones preparadas son específicas del grupo de trabajo; no puede ejecutarlas fuera del contexto del grupo de trabajo al que pertenecen.

### Consideraciones y limitaciones

- Las consultas parametrizadas se admiten en la versión 2 o una versión posterior del motor de Athena. Para obtener más información acerca de las versiones de motor Athena, consulte [Control de versiones del motor Athena](#).
- Actualmente, las consultas parametrizadas solo se admiten para instrucciones SELECT, INSERT INTO, CTAS y UNLOAD.
- En las consultas parametrizadas, los parámetros son posicionales y se indican mediante ?. A los parámetros se les asignan valores según su orden en la consulta. Los parámetros con nombre no se admiten.
- En la actualidad, los parámetros ? solo se pueden colocar en la cláusula WHERE. La sintaxis del tipo SELECT ? FROM table no es compatible.
- Los parámetros de interrogación no se pueden poner entre comillas simples o dobles (es decir, '?' y "?" no son sintaxis válidas).

- Para que los parámetros de ejecución de SQL se traten como cadenas, deben colocarse entre comillas simples en lugar de comillas dobles.
- Si es necesario, puede utilizar la función CAST al ingresar un valor para un término parametrizado. Por ejemplo, si tiene una columna del tipo date que ha parametrizado en una consulta y desea consultar la fecha 2014-07-05, al ingresar CAST( '2014-07-05' AS DATE) en el valor del parámetro obtendrá el resultado que busca.
- Las instrucciones preparadas son específicas del grupo de trabajo, y los nombres de las instrucciones preparadas deben ser únicos dentro del grupo de trabajo.
- Para las instrucciones preparadas se requieren permisos de IAM. Para obtener más información, consulte [Permitir acceso a instrucciones preparadas](#).
- Las consultas con parámetros de ejecución en la consola de Athena están limitadas a un máximo de 25 signos de interrogación.

## Consultar mediante parámetros de ejecución

Puede usar marcadores de posición de signo de interrogación en cualquier consulta DML para crear una consulta parametrizada sin crear primero una instrucción preparada. Para ejecutar estas consultas, puede utilizar la consola de Athena o la AWS CLI o el SDK de AWS y declarar las variables en el argumento `execution-parameters`.

### Ejecución de consultas con parámetros de ejecución en la consola de Athena

Cuando ejecuta una consulta parametrizada que tiene parámetros de ejecución (signos de interrogación) en la consola de Athena, se le solicitan los valores en el orden en que aparecen los signos de interrogación en la consulta.

Para ejecutar una consulta que tiene parámetros de ejecución

1. Ingrese una consulta con marcadores de posición de signo de interrogación en el editor de Athena, como se muestra en el siguiente ejemplo.

```
SELECT * FROM "my_database"."my_table"  
WHERE year = ? and month= ? and day= ?
```

2. Elija Ejecutar.
3. En el cuadro de diálogo Enter parameters (Ingresar parámetros), ingrese un valor en orden para cada uno de los signos de interrogación de la consulta.

The screenshot displays the Amazon Athena console interface. On the left, a SQL editor contains the following query:

```
1 SELECT * FROM "my_database"."my_table"  
2 WHERE year = ? and month= ? and day= ?
```

Below the editor, there are buttons for **Run**, **Cancel**, **Save** (with a dropdown arrow), **Clear**, and **Create** (with a dropdown arrow). The status bar indicates "SQL Ln 2, Col 15".

On the right, the "Enter parameters" dialog is open, showing three input fields:

- Parameter 1: 2020
- Parameter 2: (empty)
- Parameter 3: (empty)

At the bottom of the dialog are **Clear** and **Run** buttons.

Below the editor, the "Results (0)" section is visible, featuring a search bar, a "Copy" button, and a "Download results" button. The status indicates "No results" and "Run a query to view results".

4. Cuando haya terminado de ingresar los parámetros, seleccione Run (Ejecutar). El editor muestra los resultados de la consulta para los valores de parámetros que ha ingresado.

En este caso, puede elegir una de las siguientes opciones:

- Ingrese valores de parámetros diferentes para la misma consulta y, a continuación, elija Run again (Ejecutar de nuevo).
- Para borrar todos los valores que ha introducido a la vez, seleccione Clear (Eliminar).
- Para editar la consulta directamente (por ejemplo, para agregar o eliminar signos de interrogación), cierre el cuadro de diálogo Enter parameters (Ingresar parámetros) primero.
- Para guardar la consulta parametrizada para su uso posterior, seleccione Save (Guardar) o Save as (Guardar como) y, a continuación, asigne un nombre a la consulta. Para obtener más información acerca del uso de consultas guardadas, consulte [Uso de consultas guardadas](#).

Para su comodidad, el cuadro de diálogo Enter parameters (Ingresar parámetros) recuerda los valores que ha ingresado anteriormente para consultar siempre que utilice la misma pestaña en el editor de consultas.

## Ejecución de consultas con parámetros de ejecución con la AWS CLI

Para utilizar la AWS CLI para ejecutar consultas con parámetros de ejecución, utilice el comando `start-query-execution` y proporcione una consulta parametrizada en el argumento `query-string`. A continuación, en el argumento `execution-parameters`, proporcione los valores de los parámetros de ejecución. En el siguiente ejemplo, se ilustra esta técnica.

```
aws athena start-query-execution
--query-string "SELECT * FROM table WHERE x = ? AND y = ?"
--query-execution-context "Database"="default"
--result-configuration "OutputLocation"="s3://DOC-EXAMPLE-BUCKET;/..."
--execution-parameters "1" "2"
```

## Consultas con instrucciones preparadas

Puede utilizar una instrucción preparada para la ejecución repetida de la misma consulta con parámetros de consulta diferentes. Una instrucción preparada contiene marcadores de posición de parámetros cuyos valores se proporcionan en el momento de la ejecución.

### Note

El número máximo de instrucciones preparadas en un grupo de trabajo es 1000.

## Instrucciones SQL

Puede utilizar las instrucciones SQL `PREPARE`, `EXECUTE` y `DEALLOCATE PREPARE` para ejecutar consultas parametrizadas en el editor de consultas de la consola de Athena.

- Para especificar parámetros en los que normalmente usaría valores literales, utilice signos de interrogación en la instrucción `PREPARE`.
- Para reemplazar los parámetros con valores al ejecutar la consulta, utilice la cláusula `USING` en la instrucción `EXECUTE`.
- Para eliminar una instrucción preparada de las instrucciones preparadas en un grupo de trabajo, utilice la instrucción `DEALLOCATE PREPARE`.

En las siguientes secciones, se proporciona información adicional acerca de cada una de estas instrucciones.

## PREPARE

Prepara una instrucción que se ejecutará en un momento posterior. Las instrucciones preparadas se guardan en el grupo de trabajo actual con el nombre que especifique. La instrucción puede incluir parámetros en lugar de literales que se reemplazan cuando se ejecuta la consulta. Los parámetros que se van a reemplazar por valores se indican con signos de interrogación.

### Sintaxis

```
PREPARE statement_name FROM statement
```

En la siguiente tabla se describen estos parámetros.

Parámetro	Descripción
<i>statement_name</i>	Nombre de la instrucción a preparar. El nombre debe ser único dentro del grupo de trabajo.
<i>statement</i>	Una consulta SELECT, CTAS o INSERT INTO.

### Ejemplos de PREPARE

En los siguientes ejemplos se muestra el uso de la instrucción PREPARE. Los signos de interrogación indican los valores que debe proporcionar la instrucción EXECUTE cuando se ejecuta la consulta.

```
PREPARE my_select1 FROM  
SELECT * FROM nation
```

```
PREPARE my_select2 FROM  
SELECT * FROM "my_database"."my_table" WHERE year = ?
```

```
PREPARE my_select3 FROM  
SELECT order FROM orders WHERE productid = ? and quantity < ?
```

```
PREPARE my_insert FROM  
INSERT INTO cities_usa (city, state)  
SELECT city, state  
FROM cities_world
```

```
WHERE country = ?
```

```
PREPARE my_unload FROM  
UNLOAD (SELECT * FROM table1 WHERE productid < ?)  
TO 's3://DOC-EXAMPLE-BUCKET/'  
WITH (format='PARQUET')
```

## EXECUTE

Ejecuta una instrucción preparada. Los valores de los parámetros se especifican en la cláusula USING.

### Sintaxis

```
EXECUTE statement_name [USING value1 [ ,value2, ... ] ]
```

*statement\_name* es el nombre de la instrucción preparada. *value1* y *value2* son los valores que se especificarán para los parámetros de la instrucción.

### Ejemplos de EXECUTE

En el siguiente ejemplo se ejecuta la instrucción preparada `my_select1`, que no contiene parámetros.

```
EXECUTE my_select1
```

En el siguiente ejemplo se ejecuta la instrucción preparada `my_select2`, que contiene un único parámetro.

```
EXECUTE my_select2 USING 2012
```

En el siguiente ejemplo se ejecuta la instrucción preparada `my_select3`, que contiene dos parámetros.

```
EXECUTE my_select3 USING 346078, 12
```

En el siguiente ejemplo se proporciona un valor de cadena para un parámetro de la instrucción preparada `my_insert`.



```
EXECUTE my_insert USING 'usa'
```

En el siguiente ejemplo se proporciona un valor numérico para el parámetro `productid` de la instrucción preparada `my_unload`.

```
EXECUTE my_unload USING 12
```

## DEALLOCATE PREPARE

Elimina la instrucción preparada con el nombre especificado de la lista de instrucciones preparadas del grupo de trabajo actual.

### Sintaxis

```
DEALLOCATE PREPARE statement_name
```

*statement\_name* es el nombre de la instrucción preparada que se debe eliminar.

### Ejemplo

En el siguiente ejemplo, se elimina la instrucción preparada `my_select1` del grupo de trabajo actual.

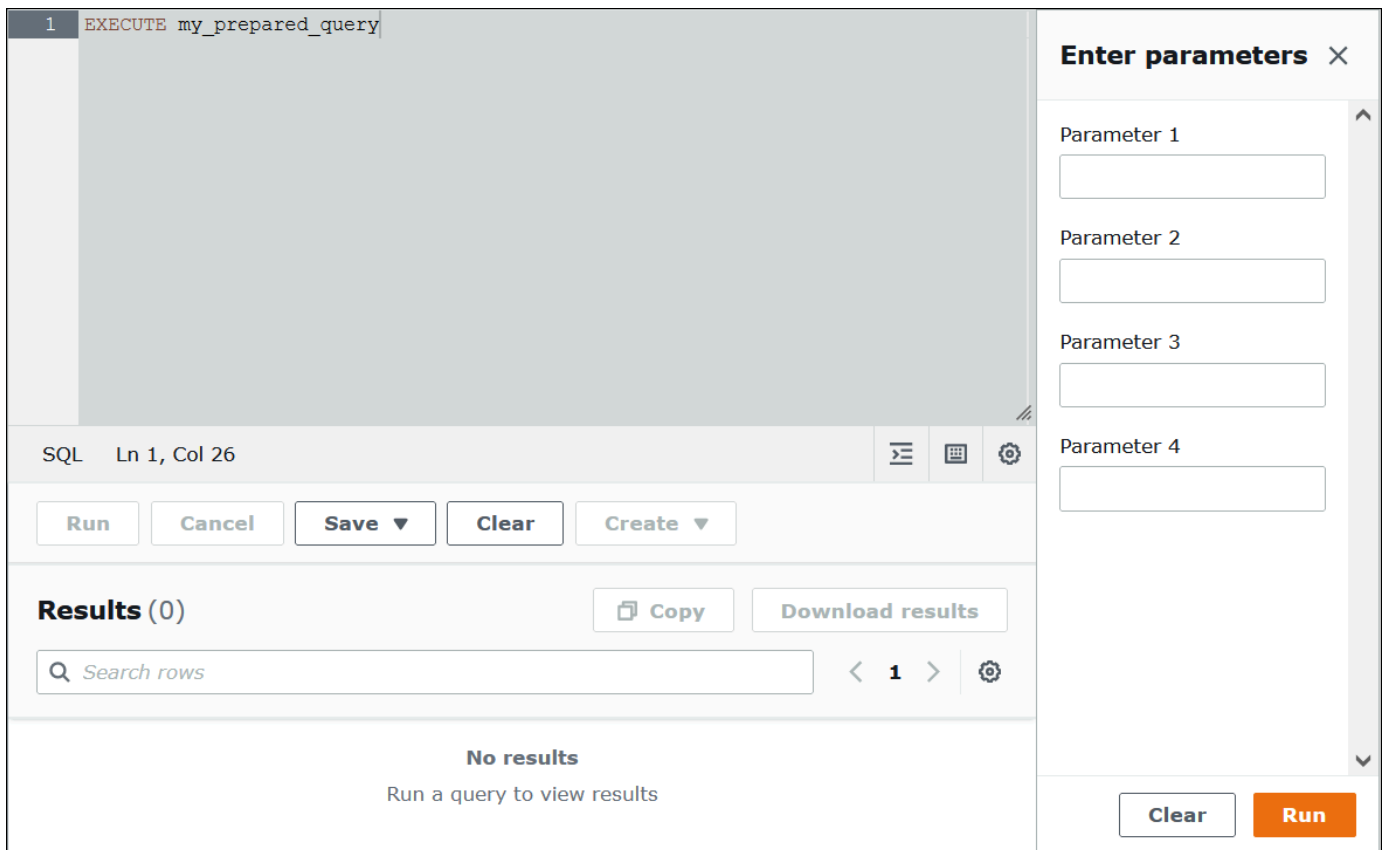
```
DEALLOCATE PREPARE my_select1
```

## Ejecución de declaraciones preparadas sin la cláusula USING en la consola de Athena

Si ejecuta una sentencia preparada existente con la sintaxis `EXECUTE prepared_statement` en el editor de consultas, Athena abre el cuadro de diálogo Enter parameters (Ingresar parámetros) para que pueda introducir los valores que normalmente aparecerían en la cláusula USING de la instrucción `EXECUTE ... USING`.

Para ejecutar una sentencia preparada mediante el cuadro de diálogo Enter parameters (Ingresar parámetros)

1. En el editor de consultas, en lugar de utilizar la sintaxis `EXECUTE prepared_statement USING Value1, Value2 ...`, utilice la sintaxis `EXECUTE prepared_statement`.
2. Elija Ejecutar. Aparece el cuadro de diálogo Enter parameters (Ingresar parámetros).



The screenshot displays the Amazon Athena console interface. At the top, a query editor shows the command `EXECUTE my_prepared_query`. Below the editor, there are buttons for **Run**, **Cancel**, **Save**, **Clear**, and **Create**. The **Results (0)** section is visible, featuring a search bar and navigation controls. A message states **No results** and *Run a query to view results*. On the right, the **Enter parameters** dialog is open, containing four input fields labeled **Parameter 1** through **Parameter 4**. At the bottom right of the dialog, there are **Clear** and **Run** buttons.

- Ingrese los valores en orden en el cuadro de diálogo Execution parameters (Ingresar parámetros). Como el texto original de la consulta no está visible, debe recordar el significado de cada parámetro posicional o tener la declaración preparada disponible como referencia.
- Elija Ejecutar.

### Creación de instrucciones preparadas mediante AWS CLI

Para utilizar la AWS CLI para crear una instrucción preparada, puede utilizar uno de los siguientes comandos de athena:

- Utilice el comando `create-prepared-statement` y proporcione una sentencia de consulta que tenga parámetros de ejecución.
- Utilice `start-query-execution` y proporcione una cadena de consulta que utilice la sintaxis `PREPARE`.

## Uso de create-prepared-statement

En un comando `create-prepared-statement`, defina el texto de la consulta en el argumento `query-statement`, como en el siguiente ejemplo.

```
aws athena create-prepared-statement
--statement-name PreparedStatement1
--query-statement "SELECT * FROM table WHERE x = ?"
--work-group athena-engine-v2
```

## Uso de start-query-execution y la sintaxis PREPARE

Utilice el comando `start-query-execution`. Coloque la instrucción `PREPARE` en el argumento `query-string`, como en el siguiente ejemplo:

```
aws athena start-query-execution
--query-string "PREPARE PreparedStatement1 FROM SELECT * FROM table WHERE x = ?"
--query-execution-context '{"Database": "default"}'
--result-configuration '{"OutputLocation": "s3://DOC-EXAMPLE-BUCKET/..."}'
```

## Ejecución de instrucciones preparadas mediante AWS CLI

Para ejecutar una declaración preparada con la AWS CLI, puede proporcionar valores para los parámetros mediante uno de los siguientes métodos:

- Utilice el argumento `execution-parameters`.
- Utilice la sintaxis SQL `EXECUTE ... USING` en el argumento `query-string`.

## Uso del argumento execution-parameters

En este enfoque, se utiliza `start-query-execution` y debe proporcionarse el nombre de una sentencia preparada existente en el argumento `query-string`. A continuación, en el argumento `execution-parameters`, proporcione los valores de los parámetros de ejecución. El siguiente ejemplo muestra este método.

```
aws athena start-query-execution
--query-string "Execute PreparedStatement1"
--query-execution-context "Database="default"
--result-configuration "OutputLocation"="s3://DOC-EXAMPLE-BUCKET/..."
```

```
--execution-parameters "1" "2"
```

## Uso de EXECUTE... Uso de sintaxis SQL

Para ejecutar una sentencia preparada existente con la sintaxis EXECUTE . . . USING, se utiliza el comando `start-query-execution` y debe colocarse el nombre de la sentencia preparada y los valores de los parámetros en el argumento `query-string`, como en el siguiente ejemplo:

```
aws athena start-query-execution
--query-string "EXECUTE PreparedStatement1 USING 1"
--query-execution-context '{"Database": "default"}'
--result-configuration '{"OutputLocation": "s3://DOC-EXAMPLE-BUCKET/...}"'
```

## Lista de instrucciones preparadas

A fin de enumerar las instrucciones preparadas para un grupo de trabajo específico, puede utilizar el comando de la AWS CLI [list-prepared-statements](#) o la acción de la API de Athena [ListPreparedStatements](#). El parámetro `--work-group` es obligatorio.

```
aws athena list-prepared-statements --work-group primary
```

## Recursos adicionales de

Consulte las siguientes publicaciones relacionadas en el Blog de macrodatos de AWS.

- [Mejore la reutilización y la seguridad mediante consultas parametrizadas de Amazon Athena](#)
- [Utilice consultas parametrizadas de Amazon Athena para proporcionar datos como servicio](#)

## Uso del optimizador basado en costes

Puede utilizar la característica de optimización basada en costes (CBO) de Athena SQL para optimizar sus consultas. Tiene la opción de solicitar a Athena que recopile estadísticas a nivel de tabla o columna para una de sus tablas en AWS Glue. Si todas las tablas de la consulta tienen estadísticas, Athena las utiliza para crear un plan de ejecución que determine que tiene el mejor rendimiento. El optimizador de consultas calcula planes alternativos en función de un modelo estadístico y, a continuación, le permite seleccionar el que probablemente sea más rápido para ejecutar la consulta.

Las estadísticas de las tablas AWS Glue se recopilan y almacenan en AWS Glue Data Catalog y se ponen a disposición de Athena para mejorar la planificación y ejecución de las consultas. Estas estadísticas son estadísticas a nivel de columna, como el número de valores distintos, nulos, máximos y mínimos en tipos de archivos como Parquet, ORC, JSON, ION, CSV y XML. Amazon Athena utiliza estas estadísticas para optimizar las consultas mediante la aplicación de los filtros más restrictivos lo antes posible en el procesamiento de consultas. Este filtrado limita el uso de memoria y la cantidad de registros que deben leerse para obtener los resultados de la consulta.

Junto con CBO, Athena utiliza una característica llamada optimizador basado en reglas (RBO). El RBO aplica de forma mecánica las reglas que se espera que mejoren el rendimiento de las consultas. El RBO suele ser beneficioso porque sus transformaciones tienen como objetivo simplificar el plan de consultas. Sin embargo, dado que RBO no realiza cálculos de costes ni comparaciones de planes, las consultas más complicadas dificultan que RBO cree un plan óptimo.

Por este motivo, Athena utiliza tanto RBO como CBO para optimizar sus consultas. Una vez que Athena identifica las oportunidades para mejorar la ejecución de las consultas, crea un plan óptimo. Para obtener más información acerca de los detalles del plan de ejecución, consulte [Ver planes de ejecución para consultas SQL](#). Para obtener información detallada sobre cómo funciona CBO, consulte la entrada del blog sobre el [optimizador basado en los costes](#) de macrodatos de AWS.

Para generar estadísticas para las tablas del catálogo de AWS Glue, puede utilizar la consola de Athena, la consola de AWS Glue o las API de AWS Glue. Como Athena está integrada con el catálogo de AWS Glue, obtendrá automáticamente las mejoras de rendimiento de las consultas correspondientes cuando ejecute consultas desde Amazon Athena.

## Consideraciones y limitaciones

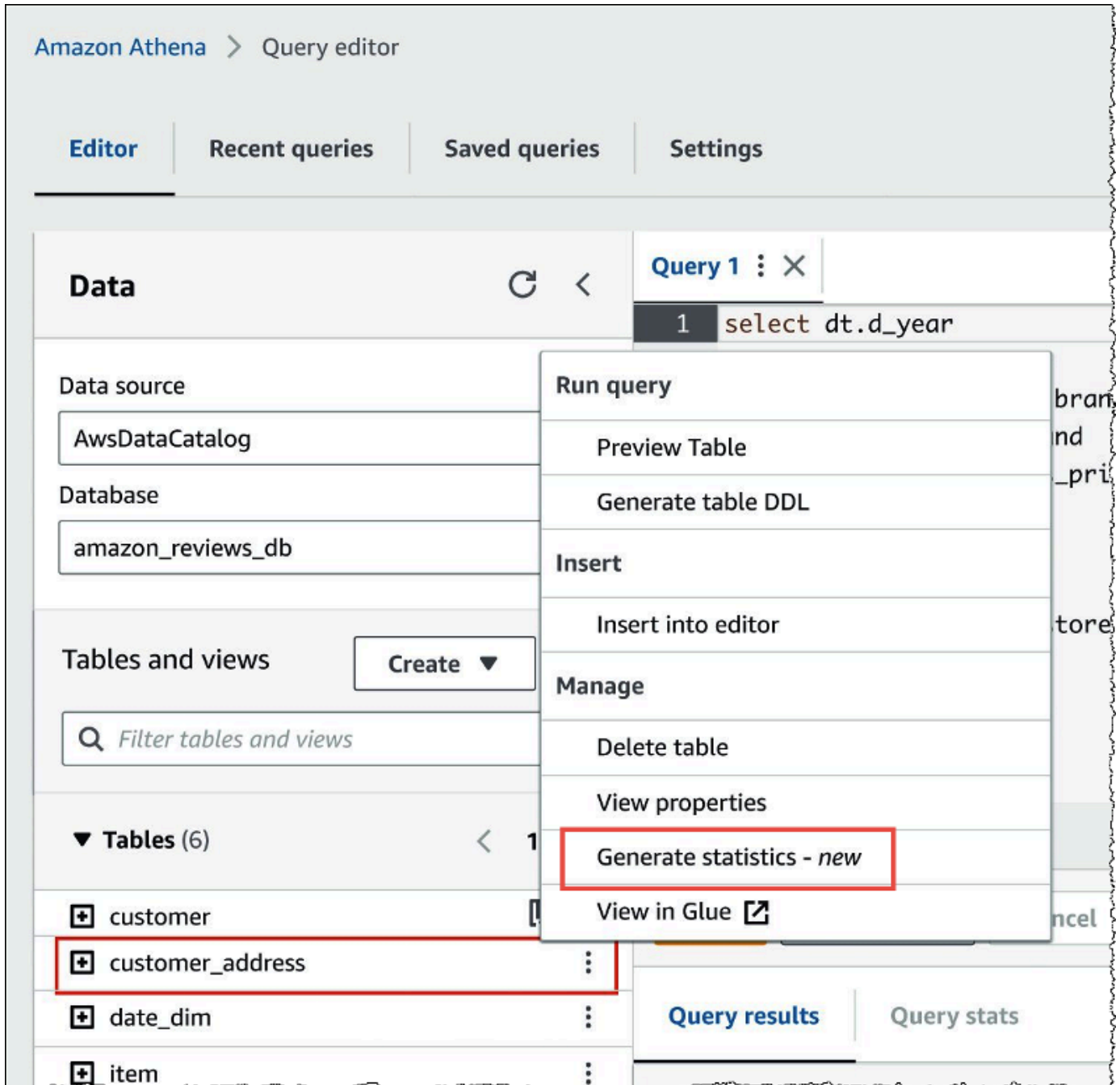
- Tipos de tablas: actualmente, la característica CBO de Athena solo admite las tablas Hive que se encuentran en AWS Glue Data Catalog.
- Athena para Spark: la característica CBO no está disponible en Athena for Spark.
- Precios: para obtener información sobre los precios, consulte la [página de precios de AWS Glue](#).

## Generación de tablas mediante la consola de Athena

En esta sección se describe cómo utilizar la consola de Athena para generar estadísticas a nivel de tablas o columnas para una tabla en AWS Glue. Para obtener información sobre cómo generar estadísticas de tablas mediante AWS Glue, consulte [Trabajar con estadísticas en columnas](#) en la Guía de desarrolladores de AWS Glue.

Para generar las estadísticas de una tabla mediante la consola de Athena

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. En la lista Tablas del editor de consultas de Athena, elija los tres puntos verticales de la tabla que desee y, a continuación, elija Generar estadísticas.



3. En el cuadro de diálogo Generar estadísticas, seleccione Todas las columnas para generar estadísticas para todas las columnas de la tabla o elija Columnas seleccionadas para seleccionar columnas específicas. Todas las columnas es el valor predeterminado.

## Generate statistics for customer\_address ✕

If statistics already exist, they will be updated.

Choose columns

All columns

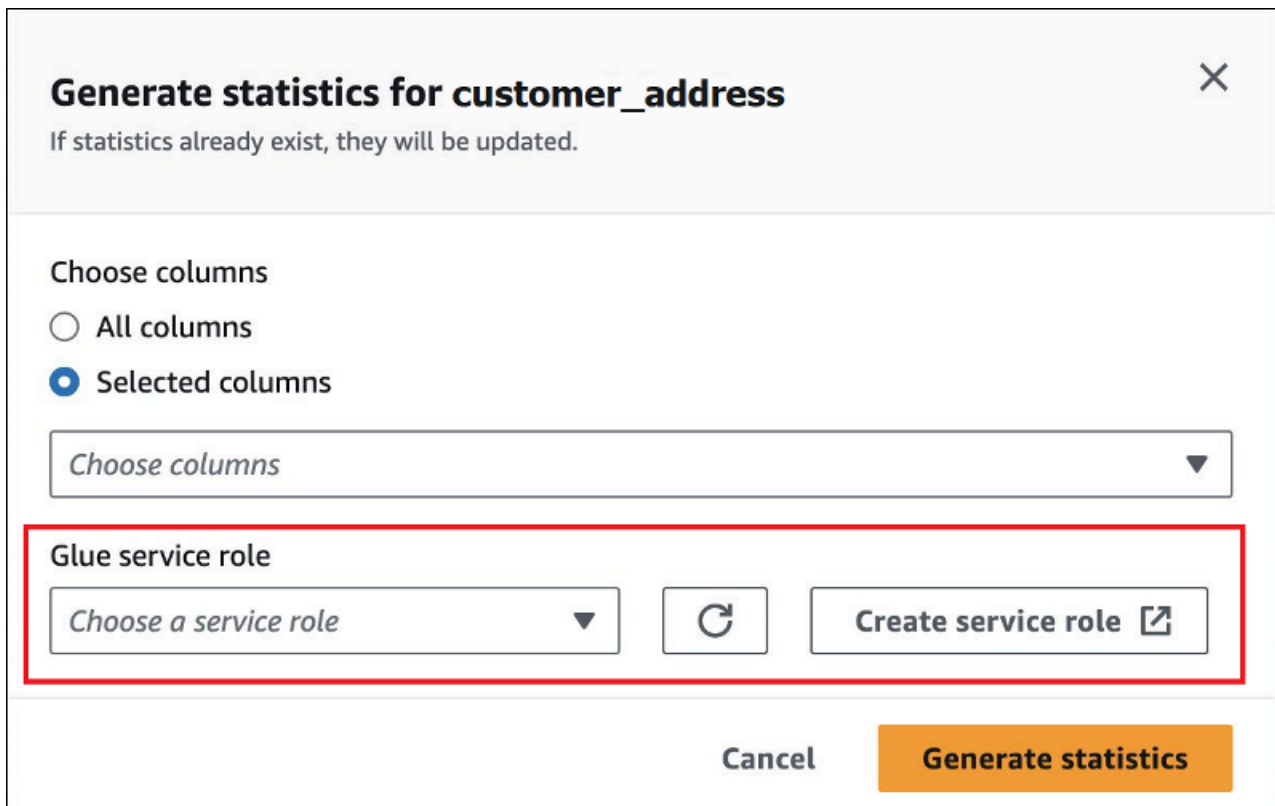
Selected columns

Choose one or more columns ▲

Q

<input checked="" type="checkbox"/>	<b>address_id</b>	string
<input checked="" type="checkbox"/>	<b>address</b>	string
<input type="checkbox"/>	<b>address2</b>	string
<input checked="" type="checkbox"/>	<b>city_id</b>	string
<input type="checkbox"/>	<b>location</b>	string
<input type="checkbox"/>	<b>phone</b>	int

4. Para el rol de servicio de AWS Glue, cree o seleccione un rol de servicio existente al que darle permiso a AWS Glue para generar estadísticas. El rol de servicio AWS Glue también requiere permisos [S3:GetObject](#) para el bucket de Amazon S3 que contiene los datos de la tabla.



**Generate statistics for customer\_address** ✕

If statistics already exist, they will be updated.

Choose columns

All columns

Selected columns

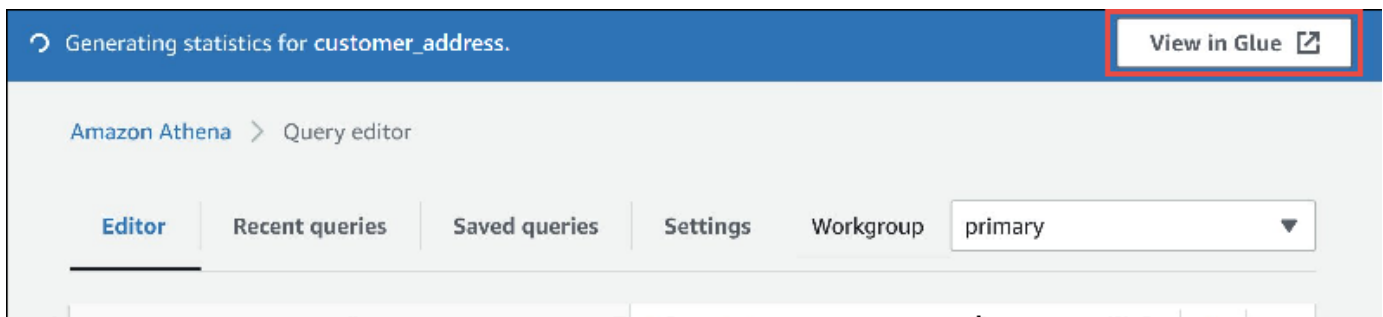
Choose columns ▼

**Glue service role**

Choose a service role ▼

Cancel

5. Elija Generar estadísticas. Un banner de notificación sobre la generación de estadísticas para **table\_name** muestra el estado de la tarea.











6. Para ver los detalles en la consola AWS Glue, seleccione Ver en Glue.

Para obtener información sobre cómo ver las estadísticas en la consola AWS Glue, consulte [Vista de estadísticas de las columnas](#) en la Guía para desarrolladores de AWS Glue.

7. Una vez generadas las estadísticas, las tablas y columnas que contienen estadísticas muestran la palabra Estadísticas entre paréntesis, como en la siguiente imagen.



▼ <b>Tables</b> (16)		< 1 >
 iris-json	<u>(Statistics)</u>	⋮
 iris-json-2.0	<u>(Statistics)</u>	⋮
 iris-json-3.0	<u>(Statistics)</u>	⋮
 iris-json-v2		⋮
 iris-json-v3		⋮
 iris-json-v4	<u>(Statistics)</u>	⋮
 iris-json-v5	<u>(Statistics)</u>	⋮
 iris-json-v6	<u>(Statistics)</u>	⋮

Ahora, cuando ejecute sus consultas, Athena realizará una optimización basada en los costes en las tablas y columnas para las que se generaron las estadísticas.

Recursos adicionales de

Para obtener información adicional, consulte el siguiente recurso.

## Consulta de datos de S3 Express One Zone

La clase de almacenamiento Amazon S3 Express One Zone es una clase de almacenamiento Amazon S3 de alto rendimiento que proporciona tiempos de respuesta de milisegundos de un solo dígito. Como tal, resulta útil para aplicaciones que acceden a los datos con frecuencia con cientos de miles de solicitudes por segundo.

S3 Express One Zone replica y almacena los datos dentro de la misma zona de disponibilidad para optimizar la velocidad y el costo. Esto difiere de las clases de almacenamiento regional de Amazon S3, que replican automáticamente los datos en un mínimo de tres zonas de disponibilidad de AWS dentro de una Región de AWS.

Para obtener más información, consulte [¿Qué es S3 Express One Zone?](#) en la Guía del usuario de Amazon S3.

### Requisitos previos

Antes de comenzar, confirme que se cumplen las siguientes condiciones:

- Versión 3 del motor Athena: para usar S3 Express One Zone con Athena SQL, su grupo de trabajo debe estar configurado para usar la versión 3 del motor Athena.
- Permisos de S3 Express One Zone: cuando S3 Express One Zone llama a una acción como GET, LIST o PUT en un objeto de Amazon S3, la clase de almacenamiento llama a `CreateSession` en su nombre. Por este motivo, su política de IAM debe permitir la acción `s3express:CreateSession`, lo que permite a Athena invocar la operación de API correspondiente.

### Consideraciones y limitaciones

Cuando realice una consulta sobre S3 Express One Zone con Athena, tenga en cuenta los siguientes puntos.

- Los buckets de S3 Express One Zone solo admiten el cifrado SSE\_S3. Los resultados de las consultas de Athena se escriben mediante cifrado SSE\_S3, independientemente de la opción que elija en la configuración del grupo de trabajo para cifrar los resultados de las consultas. Esta limitación incluye todos los escenarios en los que Athena escribe datos en buckets de S3 Express One Zone, lo que incluye `CREATE TABLE AS (CTAS)` y las instrucciones `INSERT INTO`.

- El rastreador de AWS Glue no es compatible para crear tablas con datos de S3 Express One Zone.
- No se admite la instrucción `MSCK REPAIR TABLE`. Como alternativa, utilice [ALTER TABLE ADD PARTITION](#).
- Los siguientes formatos de archivo y tabla no son compatibles o su compatibilidad es limitada. Si los formatos no aparecen en la lista, pero Athena los admite (como Parquet, ORC y JSON), también se admiten para su uso con el almacenamiento S3 Express One Zone.

Formato de archivo o tabla	Limitación
Apache Avro	No compatible
Registros de CloudTrail	No compatible
Apache Hudi	No compatible
Amazon Ion	No compatible
Registros de Logstash	No compatible
Registros de Apache WebServer	No compatible
Delta Lake	DDL no compatible. Para obtener información sobre la creación de una tabla de Delta Lake mediante un esquema ficticio, consulte <a href="#">Sincronización de metadatos de Delta Lake</a> . Se admiten consultas <code>SELECT</code> en la tabla.

## Introducción

La consulta de datos de S3 Express One Zone con Athena es sencilla. Para comenzar, siga el procedimiento detallado a continuación.

## Utilización de Athena SQL para consultar datos de S3 Express One Zone

1. Transfiera sus datos al almacenamiento de S3 Express One Zone. Para obtener más información, consulte [Configuración de la clase de almacenamiento de un objeto](#) en la Guía del usuario de Amazon S3.
2. Utilice una instrucción [CREATE TABLE](#) en Athena para catalogar sus datos en el AWS Glue Data Catalog. Para obtener más información sobre la creación de tablas en Athena, consulte la instrucción [Creación de tablas en Athena](#) y [CREATE TABLE](#).
3. (Opcional) Configure la ubicación de los resultados de la consulta de su grupo de trabajo de Athena para usar un bucket de directorio de Amazon S3. Los buckets de directorio de Amazon S3 tienen más rendimiento que los buckets generales y están diseñados para cargas de trabajo o aplicaciones de rendimiento crítico que requieren una latencia constante de milisegundos de un solo dígito. Para obtener más información, consulte [Información general de los buckets de directorio](#) en la Guía del usuario de Amazon S3.

## Consulta de objetos de Amazon S3 Glacier restaurados

Puede utilizar Athena para consultar objetos restaurados de las [clases de almacenamiento de Amazon S3](#) S3 Glacier Flexible Retrieval (anteriormente Glacier) y S3 Glacier Deep Archive. Debe habilitar esta capacidad por tabla. Si no habilita la característica en una tabla antes de ejecutar una consulta, Athena omite todos los objetos de S3 Glacier Flexible Retrieval y S3 Deep Archive de la tabla durante la ejecución de la consulta.

### Condiciones y limitaciones

- La consulta de objetos de Amazon S3 Glacier restaurados solo es compatible con la versión 3 del motor de Athena.
- La característica solo es compatible con las tablas de Apache Hive.
- Debe restaurar los objetos antes de consultar los datos; Athena no restaura los objetos en su nombre.

### Configuración de una tabla para utilizar objetos restaurados

Para configurar una tabla de Athena de modo que incluya objetos restaurados en las consultas, debe establecer su propiedad de tabla `read_restored_glacier_objects` en `true`. Para ello, puede

utilizar el editor de consultas de Athena o la consola de AWS Glue. También puede utilizar la [CLI de AWS Glue](#), la [API de AWS Glue](#) o el [SDK de AWS Glue](#).

### Uso del editor de consultas de Athena

En Athena, puede utilizar el comando [ALTER TABLE SET TBLPROPERTIES](#) para establecer la propiedad de la tabla, como en el siguiente ejemplo.

```
ALTER TABLE table_name SET TBLPROPERTIES ('read_restored_glacier_objects' = 'true')
```

### Mediante la consola de AWS Glue

En la consola de AWS Glue, realice los siguientes pasos para agregar la propiedad de tabla `read_restored_glacier_objects`.

Para configurar propiedades de tabla en la consola de AWS Glue

1. Inicie sesión en la AWS Management Console y abra la consola de AWS Glue en <https://console.aws.amazon.com/glue/>.
2. Realice una de las acciones siguientes:
  - Elija Ir al catálogo de datos.
  - En el panel de navegación, elija Tablas del catálogo de datos.
3. En la página Tablas, en la lista de tablas, elija el enlace de la tabla que desea editar.
4. Seleccione Acciones, Editar la tabla.
5. En la página Editar tabla, en la sección Propiedades de la tabla, agregue el siguiente par de clave-valor.
  - En Clave, añada `read_restored_glacier_objects`.
  - En Valor, introduzca `true`.
6. Elija Guardar.

### Uso de la AWS CLI

En la AWS CLI, puede utilizar el comando AWS Glue [update-table](#) y su argumento `--table-input` para redefinir la tabla y, al hacerlo, agregar la propiedad `read_restored_glacier_objects`. En el argumento `--table-input`, utilice la estructura `Parameters` para especificar la propiedad `read_restored_glacier_objects` y el valor de `true`. Tenga en cuenta que el argumento de

--table-input debe tener espacios y utilizar barras invertidas para evitar las comillas dobles. En el siguiente ejemplo, sustituya *my\_database* y *my\_table* por el nombre de la base de datos y la tabla.

```
aws glue update-table \  
  --database-name my_database \  
  --table-input="{\"Name\": \"my_table\", \"Parameters\": {\"read_restored_glacier_objects\": \"true\"}}"
```

### Important

El comando `update-table` de AWS Glue funciona en modo de sobrescritura, lo que significa que reemplaza la definición de tabla existente por la nueva definición especificada por el parámetro `table-input`. Por este motivo, asegúrese de especificar también todos los campos que quiera que estén presentes en la tabla del parámetro `table-input` cuando añada la propiedad `read_restored_glacier_objects`.

## Gestión de las actualizaciones de los esquemas

En esta sección, se proporciona orientación sobre cómo gestionar las actualizaciones de los esquemas con formatos de datos distintos. Athena es un motor de consultas de esquema sobre lectura. Esto significa que, al crear una tabla en Athena, aplica los esquemas al leer los datos. No cambia ni reescribe los datos subyacentes.

Si prevé cambios en los esquemas de las tablas, considere la posibilidad de crearlas en un formato de datos adecuado a sus necesidades. Su objetivo consiste en reutilizar las consultas de Athena existentes a medida que evolucionan los esquemas y evitar errores de discrepancia de esquemas al consultar tablas con particiones.

Para alcanzar este objetivo, elija un formato para los datos de una tabla basándose en la tabla del siguiente tema.

### Temas

- [Resumen: actualizaciones y formatos de datos en Athena](#)
- [Acceso de índice en ORC y Parquet](#)
- [Tipos de actualizaciones](#)
- [Actualizaciones en tablas con particiones](#)

## Resumen: actualizaciones y formatos de datos en Athena

En la siguiente tabla, se resumen los formatos de almacenamiento de datos y las manipulaciones de esquemas compatibles. Utilice esta tabla como ayuda para elegir el formato que le permitirá continuar utilizando las consultas de Athena incluso si los esquemas cambian a lo largo del tiempo.

En esta tabla, observe que Parquet y ORC son formatos de columna con diferentes métodos de acceso predeterminados a las columnas. De forma predeterminada, Parquet obtendrá acceso a las columnas por nombre y ORC por índice (valor ordinal). Por lo tanto, Athena proporciona una propiedad del SerDe definida al crear una tabla para cambiar el método de acceso predeterminado a las columnas, lo que permite una mayor flexibilidad en la evolución del esquema.

En Parquet, la propiedad `parquet.column.index.access` puede configurarse como `true`, lo que indica al método de acceso a las columnas que debe utilizar el número ordinal de la columna. Si establece esta propiedad en `false`, el método de acceso a las columnas utilizará el nombre de la columna. Del mismo modo, en ORC, utilice la propiedad `orc.column.index.access` para controlar el método de acceso a las columnas. Para obtener más información, consulte [Acceso de índice en ORC y Parquet](#).

CSV y TSV le permiten realizar todas las manipulaciones de esquemas, excepto la reordenación de columnas y la adición de columnas al principio de la tabla. Por ejemplo, si la evolución del esquema requiere únicamente cambiar el nombre de las columnas pero no eliminarlas, puede optar por crear las tablas en CSV o TSV. Si necesita eliminar columnas, no utilice CSV ni TSV, sino cualquiera de los demás formatos compatibles, preferiblemente un formato de columna, como Parquet u ORC.

### Actualizaciones de esquemas y formatos de datos en Athena

Tipo esperado de actualización de esquema	Resumen	CSV (con y sin encabezos) y TSV	JSO	AVF	PARQUE lectura por nombre (predeterminado)	PARQUE lectura por índice	ORC: lectura por índice (predeterminado)	ORC: lectura por nombre
<a href="#">Cambio del nombre de las columnas</a>	Almacene los datos en CSV y TSV, o en ORC y Parquet si se leen por índice.	Y	N	N	N	Y	Y	N

Tipo esperado de actualización de esquema	Resumen	CSV (con y sin encabezos) y TSV	JSON	AVRO	PARQUE lectura por nombre (predeterminado)	PARQUE lectura por índice	ORC: lectura por índice (predeterminado)	ORC: lectura por nombre
<a href="#">Adición de columnas al principio o en medio de la tabla</a>	Almacene los datos en JSON o AVRO, o en Parquet y ORC si se leen por nombre. No utilice CSV ni TSV.	N	Y	Y	Y	N	N	Y
<a href="#">Adición de columnas al final de la tabla</a>	Almacene los datos en CSV o TSV, JSON, AVRO, ORC o Parquet.	Y	Y	Y	Y	Y	Y	Y
<a href="#">Eliminación de columnas</a>	Almacene los datos en JSON o AVRO, o en Parquet y ORC si se leen por nombre. No utilice CSV ni TSV.	N	Y	Y	Y	N	N	Y
<a href="#">Reordenación de columnas</a>	Almacene los datos en AVRO, JSON u ORC y en Parquet si se leen por nombre.	N	Y	Y	Y	N	N	Y



Tipo esperado de actualización de esquema	Resumen	CSV (con y sin encabezos) y TSV	JSON	AVRO	PARQUET lectura por nombre (predeterminado)	PARQUET lectura por índice	ORC: lectura por índice (predeterminado)	ORC: lectura por nombre
<a href="#">Cambio del tipo de datos de una columna</a>	Almacene los datos en cualquier formato, pero pruebe la consulta en Athena para asegurarse de que los tipos de datos son compatibles. Para Parquet y ORC, el cambio de tipo de datos funciona solo para las tablas con particiones.	Y	Y	Y	Y	Y	Y	Y

## Acceso de índice en ORC y Parquet

PARQUET y ORC son formatos de almacenamiento de datos en columnas que se pueden leer por índice o por nombre. Si almacena los datos en cualquiera de estos formatos, podrá realizar todas las operaciones en los esquemas y ejecutar consultas de Athena sin errores de discrepancia de esquemas.

- Athena lee ORC por índice de forma predeterminada, tal como se define en SERDEPROPERTIES ( 'orc.column.index.access'='true' ). Para obtener más información, consulte [ORC: lectura por índice](#).
- Athena lee Parquet por nombre de forma predeterminada, tal y como se define en SERDEPROPERTIES ( 'parquet.column.index.access'='false' ). Para obtener más información, consulte [Parquet: lectura por nombre](#).

Dado que estos son valores predeterminados, es opcional especificar estas propiedades del SerDe en las consultas `CREATE TABLE`, ya que se utilizan implícitamente. Cuando se utilizan, le permiten ejecutar algunas operaciones de actualización de esquemas al mismo tiempo que impiden llevar a cabo otras operaciones similares. Para habilitar esas operaciones, ejecute otra consulta `CREATE TABLE` y cambie la configuración del SerDe.

#### Note

Las propiedades de SerDe no se propagan automáticamente en cada partición. Utilice declaraciones de `ALTER TABLE ADD PARTITION` para configurar las propiedades de SerDe para cada partición. Para automatizar este proceso, escriba un script que ejecute declaraciones de `ALTER TABLE ADD PARTITION`.

En las siguientes secciones se describen estos casos de manera detallada.

#### ORC: lectura por índice

Una tabla con formato ORC se lee por índice, de forma predeterminada. Esto lo define la siguiente sintaxis:

```
WITH SERDEPROPERTIES (  
  'orc.column.index.access'='true')
```

La lectura por índice le permite cambiar el nombre de las columnas. Sin embargo, perderá la capacidad de eliminar columnas o añadirlas en medio de la tabla.

Para que ORC lea por nombre, lo que le permitirá añadir columnas en medio de la tabla o eliminar columnas en ORC, establezca la propiedad del SerDe `orc.column.index.access` en `false` en la instrucción `CREATE TABLE`. En esta configuración, perderá la capacidad de cambiar el nombre de las columnas.

#### Note

En la versión 2 del motor de Athena, cuando las tablas ORC están configuradas para leer por nombre, Athena exige que todos los nombres de columna de los archivos ORC estén en minúsculas. Como Apache Spark no tiene nombres de campo en minúsculas cuando genera archivos ORC, es posible que Athena no pueda leer los datos así generados. La solución

consiste en cambiar el nombre de las columnas para que estén en minúsculas o usar la versión 3 del motor de Athena.

El siguiente ejemplo ilustra cómo cambiar el formato ORC para que lea por nombre:

```
CREATE EXTERNAL TABLE orders_orc_read_by_name (  
  `o_comment` string,  
  `o_orderkey` int,  
  `o_custkey` int,  
  `o_orderpriority` string,  
  `o_orderstatus` string,  
  `o_clerk` string,  
  `o_shippriority` int,  
  `o_orderdate` string  
)  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.q1.io.orc.OrcSerde'  
WITH SERDEPROPERTIES (  
  'orc.column.index.access'='false')  
STORED AS INPUTFORMAT  
  'org.apache.hadoop.hive.q1.io.orc.OrcInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive.q1.io.orc.OrcOutputFormat'  
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_orc/';
```

Parquet: lectura por nombre

Una tabla con formato Parquet se lee por nombre, de forma predeterminada. Esto lo define la siguiente sintaxis:

```
WITH SERDEPROPERTIES (  
  'parquet.column.index.access'='false')
```

La lectura por nombre le permite añadir columnas en medio de la tabla y eliminar columnas. Sin embargo, perderá la capacidad de cambiar el nombre de las columnas.

Para que Parquet lea por índice, lo que le permitirá cambiar el nombre de las columnas, debe crear una tabla con la propiedad del SerDe `parquet.column.index.access` establecida en `true`.

## Tipos de actualizaciones

En este tema se describen algunos de los cambios que puede realizar en el esquema en instrucciones CREATE TABLE sin alterar realmente los datos. Se examina cada tipo de actualización de esquema y se especifican los formatos de datos que permiten realizarlas en Athena. Para actualizar un esquema, en algunos casos se puede utilizar un comando ALTER TABLE, pero en otros no se modifica realmente una tabla existente. En su lugar, se crea una tabla con un nombre nuevo que modifica el esquema que utilizó en la instrucción CREATE TABLE original.

- [Adición de columnas al principio o en el medio de la tabla](#)
- [Adición de columnas al final de la tabla](#)
- [Eliminación de columnas](#)
- [Cambio del nombre de las columnas](#)
- [Cambio del orden de las columnas](#)
- [Cambio del tipo de datos de una columna](#)

Elija un formato de datos compatible que le permita seguir utilizando las consultas de Athena teniendo en cuenta la evolución prevista de los esquemas.

Supongamos que tiene una aplicación que lee información de pedidos de una tabla `orders` que existe en dos formatos: CSV y Parquet.

El ejemplo siguiente crea una tabla en Parquet:

```
CREATE EXTERNAL TABLE orders_parquet (  
  `orderkey` int,  
  `orderstatus` string,  
  `totalprice` double,  
  `orderdate` string,  
  `orderpriority` string,  
  `clerk` string,  
  `shippriority` int  
) STORED AS PARQUET  
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_ parquet/';
```

El ejemplo siguiente crea la misma tabla en CSV:

```
CREATE EXTERNAL TABLE orders_csv (  
  `orderkey` int,
```

```
`orderstatus` string,  
`totalprice` double,  
`orderdate` string,  
`orderpriority` string,  
`clerk` string,  
`shippriority` int  
)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','  
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_csv/';
```

En las secciones siguientes, examinaremos cómo afectan las actualizaciones de estas tablas a las consultas de Athena.

### Adición de columnas al principio o en el medio de la tabla

La adición de columnas es uno de los cambios de esquema más frecuentes. Por ejemplo, se puede añadir una columna para enriquecer una tabla con datos nuevos. O añadir una columna si el origen de una columna existente ha cambiado y se desea mantener la versión anterior de esta columna para que sigan funcionando las aplicaciones que dependen de ella.

Para añadir columnas al principio o en medio de la tabla y continuar ejecutando consultas en las tablas existentes, utilice AVRO, JSON, Parquet u ORC si su propiedad del SerDe está configurada para la lectura por nombre. Para obtener más información, consulte [Acceso de índice en ORC y Parquet](#).

No añada columnas al principio o en medio de la tabla en los formatos CSV y TSV, ya que estos dependen de la ordenación. La adición de una columna en estos casos dará lugar a errores de discrepancia de esquemas cuando cambie el esquema de las particiones.

En el siguiente ejemplo, se crea una tabla nueva que agrega una columna `o_comment` en el centro de una tabla en función de los datos JSON.

```
CREATE EXTERNAL TABLE orders_json_column_addition (  
  `o_orderkey` int,  
  `o_custkey` int,  
  `o_orderstatus` string,  
  `o_comment` string,  
  `o_totalprice` double,  
  `o_orderdate` string,  
  `o_orderpriority` string,  
  `o_clerk` string,  
  `o_shippriority` int,
```

```
)  
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'  
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_json/';
```

## Adición de columnas al final de la tabla

Si crea tablas en cualquiera de los formatos que admite Athena, como Parquet, ORC, Avro, JSON, CSV y TSV, puede usar la instrucción `ALTER TABLE ADD COLUMNS` para agregar columnas detrás de las columnas existentes pero antes de las columnas de partición.

En el ejemplo siguiente se agrega una columna `comment` al final de la tabla `orders_parquet` antes de cualquier columna de partición:

```
ALTER TABLE orders_parquet ADD COLUMNS (comment string)
```

### Note

Para ver una nueva columna de tabla en el Editor de consultas de Athena después de ejecutar `ALTER TABLE ADD COLUMNS`, actualice manualmente la lista de tablas en el editor y vuelva a expandirla.

## Eliminación de columnas

Es posible que sea necesario eliminar columnas de las tablas si ya no contienen datos o para restringir el acceso a los datos que contienen.

- Puede eliminar columnas de tablas en JSON, Avro, Parquet y ORC si se leen por nombre. Para obtener más información, consulte [Acceso de índice en ORC y Parquet](#).
- No recomendamos eliminar columnas de las tablas en CSV y TSV si se desea conservar las tablas ya creadas en Athena. La eliminación de una columna rompe el esquema y requiere que se vuelva a crear la tabla sin la columna eliminada.

En este ejemplo, se elimina una columna `totalprice`` de una tabla en Parquet y se ejecuta una consulta. En Athena, el formato Parquet se lee por nombre de forma predeterminada, por lo que omitimos la configuración `SERDEPROPERTIES` que especifica la lectura por nombre. Observe que la consulta siguiente se realiza correctamente, aunque haya cambiado el esquema:

```
CREATE EXTERNAL TABLE orders_parquet_column_removed (
```

```
`o_orderkey` int,  
`o_custkey` int,  
`o_orderstatus` string,  
`o_orderdate` string,  
`o_orderpriority` string,  
`o_clerk` string,  
`o_shippriority` int,  
`o_comment` string  
)  
STORED AS PARQUET  
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_parquet/';
```

## Cambio del nombre de las columnas

Es posible que desee cambiar el nombre de las columnas de las tablas para corregir errores de ortografía, hacer que los nombres de las columnas sean más descriptivos o reutilizar una columna existente para evitar la reordenación de columnas.

Puede cambiar el nombre de las columnas si almacena los datos en CSV y TSV, o en Parquet y ORC si están configurados para leer por índice. Para obtener más información, consulte [Acceso de índice en ORC y Parquet](#).

Athena lee los datos con formato CSV y TSV en el orden que tienen las columnas en el esquema y los devuelve en el mismo orden. No utiliza los nombres de las columnas para el mapeo de datos a una columna y por ello es posible cambiar el nombre de las columnas en CSV o TSV sin invalidar las consultas de Athena.

Una estrategia para cambiar el nombre de las columnas consiste en crear una nueva tabla basada en los mismos datos subyacentes, pero utilizando nuevos nombres de columna. En el siguiente ejemplo, se crea una nueva tabla `orders_parquet` llamada `orders_parquet_column_renamed`. En el ejemplo se cambia el nombre de la columna ``o_totalprice`` a ``o_total_price`` y, a continuación, se ejecuta una consulta en Athena:

```
CREATE EXTERNAL TABLE orders_parquet_column_renamed (  
  `o_orderkey` int,  
  `o_custkey` int,  
  `o_orderstatus` string,  
  `o_total_price` double,  
  `o_orderdate` string,  
  `o_orderpriority` string,  
  `o_clerk` string,
```

```
`o_shippriority` int,  
`o_comment` string  
)  
STORED AS PARQUET  
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_parquet/';
```

En el caso de la tabla en Parquet, la consulta siguiente se ejecuta, pero la columna cuyo nombre ha cambiado no muestra datos, ya que se estaba obteniendo acceso a ella por nombre (comportamiento predeterminado en Parquet) en lugar de por índice:

```
SELECT *  
FROM orders_parquet_column_renamed;
```

Una consulta con una tabla en CSV tiene un aspecto similar:

```
CREATE EXTERNAL TABLE orders_csv_column_renamed (  
  `o_orderkey` int,  
  `o_custkey` int,  
  `o_orderstatus` string,  
  `o_total_price` double,  
  `o_orderdate` string,  
  `o_orderpriority` string,  
  `o_clerk` string,  
  `o_shippriority` int,  
  `o_comment` string  
)  
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','  
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_csv/';
```

En el caso de la tabla en CSV, se ejecuta la consulta siguiente y los datos se muestran en todas las columnas, incluida aquella cuyo nombre se ha cambiado:

```
SELECT *  
FROM orders_csv_column_renamed;
```

### Cambio del orden de las columnas

Solo podrá reordenar las columnas de las tablas que tengan datos en formatos que lean por nombre, como JSON o Parquet, que lean por nombre de forma predeterminada. También puede hacer que ORC lea por nombre, si es necesario. Para obtener más información, consulte [Acceso de índice en ORC y Parquet](#).



En el siguiente ejemplo, se crea una tabla nueva con las columnas en un orden diferente:

```
CREATE EXTERNAL TABLE orders_parquet_columns_reordered (  
  `o_comment` string,  
  `o_orderkey` int,  
  `o_custkey` int,  
  `o_orderpriority` string,  
  `o_orderstatus` string,  
  `o_clerk` string,  
  `o_shippriority` int,  
  `o_orderdate` string  
)  
STORED AS PARQUET  
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_parquet/';
```

## Cambio del tipo de datos de una columna

Es posible que desee utilizar un tipo de columna diferente cuando el tipo existente ya no pueda contener la cantidad de información requerida. Por ejemplo, los valores de una columna de ID pueden superar el tamaño del tipo de datos INT y requerir el uso del tipo de datos BIGINT.

Cuando tenga previsto utilizar un tipo de datos diferente para una columna, tenga en cuenta los siguientes puntos:

- En la mayoría de los casos, no puede cambiar el tipo de datos de una columna de forma directa. En su lugar, se vuelve a crear la tabla de Athena y se define la columna con el tipo de datos nuevo.
- Solo es posible leer determinados tipos de datos como otros tipos de datos. Consulte en la tabla de esta sección los tipos de datos que se pueden gestionar de esa manera.
- En el caso de los datos en formato Parquet y ORC, no es posible utilizar un tipo de datos diferente para una columna si la tabla no se encuentre particionada.
- Para las tablas particionadas con formato Parquet y ORC, el tipo de las columnas de una partición puede ser diferente al que tienen en otra y Athena aplicará CAST para asignar el deseado, si es posible. Para obtener más información, consulte [Cómo evitar los errores de discrepancia de esquemas para las tablas con particiones](#).
- En el caso de las tablas que solo se crearon con [LazySimpleSerDe](#), es posible utilizar la instrucción ALTER TABLE REPLACE COLUMNS para sustituir las columnas existentes por un tipo de datos diferente, pero todas las columnas existentes que desee conservar también deben redefinirse en la instrucción o, de lo contrario, se eliminarán. Para obtener más información, consulte [ALTER TABLE REPLACE COLUMNS](#).

- Solo en el caso de las tablas de Apache Iceberg, puede utilizar la instrucción [ALTER TABLE CHANGE COLUMN](#) para cambiar el tipo de datos de una columna. ALTER TABLE REPLACE COLUMNS no es compatible con las tablas de Iceberg. Para obtener más información, consulte [Esquema de tabla de Iceberg en evolución](#).

#### Important

Recomendamos encarecidamente que compruebe y verifique las consultas antes de realizar traducciones de tipos de datos. Si Athena no puede utilizar el tipo de datos de destino, es posible que falle la consulta CREATE TABLE.

En la siguiente tabla se enumeran los tipos de datos que se gestionarán como otros tipos de datos:

#### Tipos de datos compatibles

Tipo de datos original	Tipos de datos de destino disponibles
STRING	BYTE, TINYINT, SMALLINT, INT, BIGINT
BYTE	TINYINT, SMALLINT, INT, BIGINT
TINYINT	SMALLINT, INT, BIGINT
SMALLINT	INT, BIGINT
INT	BIGINT
FLOAT	DOUBLE

En el siguiente ejemplo, se utiliza la instrucción CREATE TABLE de la tabla orders\_json original para crear una tabla nueva denominada orders\_json\_bigint. La tabla nueva utiliza BIGINT en lugar de INT como tipo de datos para la columna `o\_shippriority`.

```
CREATE EXTERNAL TABLE orders_json_bigint (
  `o_orderkey` int,
  `o_custkey` int,
  `o_orderstatus` string,
  `o_totalprice` double,
```

```
`o_orderdate` string,  
`o_orderpriority` string,  
`o_clerk` string,  
`o_shippriority` BIGINT  
)  
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'  
LOCATION 's3://DOC-EXAMPLE-BUCKET/orders_json';
```

La consulta siguiente se ejecuta correctamente, de forma similar a la consulta SELECT original, antes de cambiar el tipo de datos:

```
Select * from orders_json  
LIMIT 10;
```

## Actualizaciones en tablas con particiones

En Athena, una tabla y sus particiones deben utilizar los mismos formatos de datos, pero sus esquemas pueden ser diferentes. Al crear una partición, esta normalmente hereda el esquema de la tabla. Con el tiempo, los esquemas puede comenzar a ser diferentes. Los motivos pueden ser:

- Si se modifica el esquema de la tabla, los esquemas de las particiones no se actualizan para continuar sincronizados con el esquema de la tabla.
- El rastreador de AWS Glue permite descubrir datos en particiones con esquemas distintos. Esto significa que, si se crea una tabla en Athena con AWS Glue, cuando el rastreador finalice el procesamiento, los esquemas de la tabla y los de sus particiones podrán ser distintos.
- Si se agregan particiones directamente a través de una API de AWS.

Athena procesa correctamente las tablas con particiones si se cumplen las restricciones siguientes. Si no se cumplen estas restricciones, Athena genera un error `HIVE_PARTITION_SCHEMA_MISMATCH`.

- El esquema de cada partición es compatible con el esquema de la tabla.
- El formato de datos de la tabla permite del tipo de actualización que se desea realizar: añadir, eliminar, cambiar el orden de las columnas o cambiar el tipo de datos de una columna.

Por ejemplo, en el caso de los formato CSV y TSV, solo se puede cambiar el nombre de las columnas, añadir columnas al final de la tabla y cambiar el tipo de datos de una columna si los tipos son compatibles, pero no se pueden eliminar columnas. Para otros formatos, se pueden

añadir o eliminar columnas, o cambiar el tipo de datos de una columna por otro si los tipos son compatibles. Para obtener información, consulte [Resumen: Actualizaciones y formatos de datos en Athena](#).

## Cómo evitar los errores de discrepancia de esquemas para las tablas con particiones

Al principio de la ejecución de las consultas, Athena verifica el esquema de la tabla verificando que el tipo de datos de cada columna es compatible entre la tabla y la partición.

- Para los tipos de almacenamiento de datos Parquet y ORC, Athena tiene en cuenta los nombres de las columnas y los utiliza para la verificación de esquemas basada en el nombre de las columnas. Esto elimina los errores `HIVE_PARTITION_SCHEMA_MISMATCH` para las tablas con particiones en Parquet y ORC. (Esto es válido para ORC si la propiedad del `SerDe` está establecida para obtener acceso al índice por nombre: `orc.column.index.access=FALSE`. Parquet lee el índice por nombre de forma predeterminada).
- Para CSV, JSON y Avro, Athena utiliza una verificación de esquemas basada en índices. Esto significa que, si detecta un error de discrepancia de esquemas, debe eliminar la partición que está provocando la discrepancia y volver a crearla para que Athena pueda consultarla sin que se produzcan errores.

Athena compara el esquema de la tabla con los esquemas de las particiones. Si se crea una tabla con formato CSV, JSON y AVRO en Athena usando el rastreador de AWS Glue, cuando el rastreador finalice el procesamiento los esquemas de la tabla y los de sus particiones podrán ser distintos. Si hay una discrepancia entre el esquema de la tabla y los esquemas de las particiones, las consultas de Athena generan errores de verificación de esquema similares a este: `'crawler_test.click_avro' is declared as type 'string', but partition 'partition_0=2017-01-17' declared column 'col68' as type 'double'` ('crawler\_test.click\_avro' se ha declarado como de tipo 'cadena', pero la partición 'partition\_0=2017-01-17' ha declarado la columna 'col68' como de tipo 'doble').

Una solución típica para este tipo de errores consiste en eliminar la partición que provoca el error y volver a crearla. Para obtener más información, consulte [ALTER TABLE DROP PARTITION](#) y [ALTER TABLE ADD PARTITION](#).

## Matrices de consulta

Amazon Athena le permite crear matrices, concatenarlas, convertirlas en diferentes tipos de datos y también filtrarlas, aplanarlas y ordenarlas.

## Temas

- [Creación de matrices](#)
- [Concatenación de cadenas y matrices](#)
- [Conversión de tipos de datos de matrices](#)
- [Determinación de longitudes](#)
- [Acceso a los elementos de la matriz](#)
- [Aplanamiento de matrices anidadas](#)
- [Creación de matrices a partir de subconsultas](#)
- [Filtro de matrices](#)
- [Ordenación de matrices](#)
- [Uso de funciones de agregación con matrices](#)
- [Conversión de matrices en cadenas](#)
- [Uso de matrices para crear mapas](#)
- [Consultas de matrices con tipos complejos y estructuras anidadas](#)

## Creación de matrices

Para crear una matriz literal en Athena, utilice la palabra clave ARRAY seguida de paréntesis [ ] e incluya los elementos de matriz separados por comas.

### Ejemplos

Esta consulta crea una matriz con cuatro elementos.

```
SELECT ARRAY [1,2,3,4] AS items
```

Devuelve:

```
+-----+  
| items  |  
+-----+  
| [1,2,3,4] |  
+-----+
```

Esta consulta crea dos matrices.

```
SELECT ARRAY[ ARRAY[1,2], ARRAY[3,4] ] AS items
```

Devuelve:

```
+-----+
| items          |
+-----+
| [[1, 2], [3, 4]] |
+-----+
```

Para crear una matriz a partir de columnas seleccionadas de tipos compatibles, utilice una consulta tal y como se muestra en el ejemplo siguiente:

```
WITH
dataset AS (
  SELECT 1 AS x, 2 AS y, 3 AS z
)
SELECT ARRAY [x,y,z] AS items FROM dataset
```

Esta consulta devuelve:

```
+-----+
| items      |
+-----+
| [1,2,3]    |
+-----+
```

En el siguiente ejemplo, se seleccionan dos matrices y se devuelven como mensaje de bienvenida.

```
WITH
dataset AS (
  SELECT
    ARRAY ['hello', 'amazon', 'athena'] AS words,
    ARRAY ['hi', 'alexa'] AS alexa
)
SELECT ARRAY[words, alexa] AS welcome_msg
FROM dataset
```

Esta consulta devuelve:

```
+-----+
| welcome_msg |
+-----+
| [[hello, amazon, athena], [hi, alexa]] |
+-----+
```

Para crear una matriz de pares de clave-valor, utilice el operador MAP que toma una matriz de claves seguida de una matriz de valores, tal y como se muestra en ejemplo siguiente:

```
SELECT ARRAY[
  MAP(ARRAY['first', 'last', 'age'],ARRAY['Bob', 'Smith', '40']),
  MAP(ARRAY['first', 'last', 'age'],ARRAY['Jane', 'Doe', '30']),
  MAP(ARRAY['first', 'last', 'age'],ARRAY['Billy', 'Smith', '8'])
] AS people
```

Esta consulta devuelve:

```
+-----+
+
| people |
+-----+
+
| [{last=Smith, first=Bob, age=40}, {last=Doe, first=Jane, age=30}, {last=Smith,
first=Billy, age=8}] |
+-----+
+
```

## Concatenación de cadenas y matrices

### Concatenación de cadenas

Para concatenar dos cadenas, puede utilizar el operador de canalización doble ||, como en el siguiente ejemplo.

```
SELECT 'This' || ' is' || ' a' || ' test.' AS Concatenated_String
```

Esta consulta devuelve:

#	Concatenated_String
1	This is a test.

Puede utilizar la función `concat()` para lograr el mismo resultado.

```
SELECT concat('This', ' is', ' a', ' test.') AS Concatenated_String
```

Esta consulta devuelve:

#	Concatenated_String
1	This is a test.

Puede utilizar la función `concat_ws()` para concatenar cadenas con el separador especificado en el primer argumento.

```
SELECT concat_ws(' ', 'This', 'is', 'a', 'test.') as Concatenated_String
```

Esta consulta devuelve:

#	Concatenated_String
1	This is a test.

Para concatenar dos columnas del tipo de datos de cadena con un punto, haga referencia a las dos columnas con comillas dobles y agregue comillas simples al punto como una codificación rígida. Si una columna no es del tipo de datos de cadena, puede utilizar `CAST("column_name" as VARCHAR)` para convertir la columna primero.

```
SELECT "col1" || '.' || "col2" as Concatenated_String  
FROM my_table
```

Esta consulta devuelve:



#	Concatenated_String
1	<i>col1_string_value .col2_string_value</i>

## Concatenación de matrices

Puede utilizar las mismas técnicas para concatenar matrices.

Para concatenar varias matrices, utilice el operador de canalización doble `||`.

```
SELECT ARRAY [4,5] || ARRAY[ ARRAY[1,2], ARRAY[3,4] ] AS items
```

Esta consulta devuelve:

#	items
1	[[4, 5], [1, 2], [3, 4]]

Para combinar varias matrices en una sola, utilice el operador de canalización doble o la función `concat()`.

```
WITH
dataset AS (
  SELECT
    ARRAY ['Hello', 'Amazon', 'Athena'] AS words,
    ARRAY ['Hi', 'Alexa'] AS alexa
)
SELECT concat(words, alexa) AS welcome_msg
FROM dataset
```

Esta consulta devuelve:

#	welcome_msg
1	[Hello, Amazon, Athena, Hi, Alexa]

Para obtener más información sobre otras funciones de cadena `concat()`, consulte [String functions and operators](#) (Funciones y operadores de cadena) en la documentación de Trino.

## Conversión de tipos de datos de matrices

Para convertir datos de matrices en tipos de datos compatibles, utilice el operador `CAST` como `CAST(value AS type)`. Athena es compatible con todos los tipos de datos de Presto nativos.

```
SELECT
  ARRAY [CAST(4 AS VARCHAR), CAST(5 AS VARCHAR)]
AS items
```

Esta consulta devuelve:

```
+-----+
| items |
+-----+
| [4,5] |
+-----+
```

Cree dos matrices con elementos de pares de clave-valor, conviértalos a JSON y realice una concatenación, tal y como se muestra en este ejemplo:

```
SELECT
  ARRAY[CAST(MAP(ARRAY['a1', 'a2', 'a3'], ARRAY[1, 2, 3]) AS JSON)] ||
  ARRAY[CAST(MAP(ARRAY['b1', 'b2', 'b3'], ARRAY[4, 5, 6]) AS JSON)]
AS items
```

Esta consulta devuelve:

```
+-----+
| items |
+-----+
| [{"a1":1,"a2":2,"a3":3}, {"b1":4,"b2":5,"b3":6}] |
+-----+
```

## Determinación de longitudes

La función `cardinality` devuelve la longitud de una matriz, como en este ejemplo:

```
SELECT cardinality(ARRAY[1,2,3,4]) AS item_count
```

Esta consulta devuelve:

```
+-----+
| item_count |
+-----+
| 4          |
+-----+
```

## Acceso a los elementos de la matriz

Para obtener acceso a los elementos de una matriz, utilice el operador `[]` y especifique 1 para el primer elemento, 2 para el segundo elemento y así sucesivamente, tal y como se muestra en el ejemplo siguiente:

```
WITH dataset AS (
SELECT
  ARRAY[CAST(MAP(ARRAY['a1', 'a2', 'a3'], ARRAY[1, 2, 3]) AS JSON)] ||
  ARRAY[CAST(MAP(ARRAY['b1', 'b2', 'b3'], ARRAY[4, 5, 6]) AS JSON)]
AS items )
SELECT items[1] AS item FROM dataset
```

Esta consulta devuelve:

```
+-----+
| item          |
+-----+
| {"a1":1,"a2":2,"a3":3} |
+-----+
```

Para obtener acceso a los elementos de una matriz en una determinada posición (conocida como posición de índice), utilice la función `element_at()` y especifique el nombre de la matriz y la posición de índice:

- Si el índice es mayor que 0, `element_at()` devuelve el elemento que usted especifique, contando desde el principio hasta el final de la matriz. Actúa como el operador `[]`.
- Si el índice es inferior a 0, `element_at()` devuelve el elemento contando desde el final hasta el principio de la matriz.

La siguiente consulta crea una matriz `words` y selecciona el primer elemento `hello` de esta como `first_word`, el segundo elemento `amazon` (contando a partir del final de la matriz) como `middle_word` y el tercer elemento `athena`, como la `last_word`.

```
WITH dataset AS (
  SELECT ARRAY ['hello', 'amazon', 'athena'] AS words
)
SELECT
  element_at(words, 1) AS first_word,
  element_at(words, -2) AS middle_word,
  element_at(words, cardinality(words)) AS last_word
FROM dataset
```

Esta consulta devuelve:

```
+-----+
| first_word | middle_word | last_word |
+-----+
| hello      | amazon      | athena    |
+-----+
```

## Aplanamiento de matrices anidadas

Cuando trabaja con matrices anidadas, a menudo necesita ampliar elementos de matriz anidados en una única matriz o ampliar la matriz en varias filas.

### Ejemplos

Para aplanar los elementos de una matriz anidada en una única matriz de valores, utilice la función `flatten`. Esta consulta devuelve una fila para cada elemento de la matriz.

```
SELECT flatten(ARRAY[ ARRAY[1,2], ARRAY[3,4] ]) AS items
```

Esta consulta devuelve:

```
+-----+
| items    |
+-----+
| [1,2,3,4] |
+-----+
```

Para aplanar una matriz en varias filas, utilice CROSS JOIN junto con el operador UNNEST, tal y como se muestra en este ejemplo:

```
WITH dataset AS (
  SELECT
    'engineering' as department,
    ARRAY['Sharon', 'John', 'Bob', 'Sally'] as users
)
SELECT department, names FROM dataset
CROSS JOIN UNNEST(users) as t(names)
```

Esta consulta devuelve:

```
+-----+
| department | names |
+-----+
| engineering | Sharon |
+-----+
| engineering | John  |
+-----+
| engineering | Bob   |
+-----+
| engineering | Sally |
+-----+
```

Para aplanar una matriz de pares de clave-valor, transponga las claves seleccionadas en columnas, tal y como se muestra en este ejemplo:

```
WITH
dataset AS (
  SELECT
    'engineering' as department,
    ARRAY[
      MAP(ARRAY['first', 'last', 'age'],ARRAY['Bob', 'Smith', '40']),
      MAP(ARRAY['first', 'last', 'age'],ARRAY['Jane', 'Doe', '30']),
      MAP(ARRAY['first', 'last', 'age'],ARRAY['Billy', 'Smith', '8'])
    ] AS people
)
SELECT names['first'] AS
first_name,
names['last'] AS last_name,
department FROM dataset
```

```
CROSS JOIN UNNEST(people) AS t(names)
```

Esta consulta devuelve:

```
+-----+
| first_name | last_name | department |
+-----+
| Bob        | Smith    | engineering |
| Jane       | Doe      | engineering |
| Billy      | Smith    | engineering |
+-----+
```

Partiendo de una lista de empleados, seleccione el que tenga la puntuación combinada más alta. UNNEST puede utilizarse en la cláusula FROM sin ir precedido de CROSS JOIN, ya que es el operador de combinación predeterminado y está implícito.

```
WITH
dataset AS (
  SELECT ARRAY[
    CAST(ROW('Sally', 'engineering', ARRAY[1,2,3,4]) AS ROW(name VARCHAR, department
VARCHAR, scores ARRAY(INTEGER))),
    CAST(ROW('John', 'finance', ARRAY[7,8,9]) AS ROW(name VARCHAR, department VARCHAR,
scores ARRAY(INTEGER))),
    CAST(ROW('Amy', 'devops', ARRAY[12,13,14,15]) AS ROW(name VARCHAR, department
VARCHAR, scores ARRAY(INTEGER)))
  ] AS users
),
users AS (
  SELECT person, score
  FROM
    dataset,
    UNNEST(dataset.users) AS t(person),
    UNNEST(person.scores) AS t(score)
)
SELECT person.name, person.department, SUM(score) AS total_score FROM users
GROUP BY (person.name, person.department)
ORDER BY (total_score) DESC
LIMIT 1
```

Esta consulta devuelve:

```
+-----+
```

```

| name | department | total_score |
+-----+
| Amy  | devops     | 54          |
+-----+

```

En una lista de empleados seleccione el empleado con la máxima puntuación individual.

```

WITH
dataset AS (
  SELECT ARRAY[
    CAST(ROW('Sally', 'engineering', ARRAY[1,2,3,4]) AS ROW(name VARCHAR, department
  VARCHAR, scores ARRAY(INTEGER))),
    CAST(ROW('John', 'finance', ARRAY[7,8,9]) AS ROW(name VARCHAR, department VARCHAR,
  scores ARRAY(INTEGER))),
    CAST(ROW('Amy', 'devops', ARRAY[12,13,14,15]) AS ROW(name VARCHAR, department
  VARCHAR, scores ARRAY(INTEGER)))
  ] AS users
),
users AS (
  SELECT person, score
  FROM
    dataset,
    UNNEST(dataset.users) AS t(person),
    UNNEST(person.scores) AS t(score)
)
SELECT person.name, score FROM users
ORDER BY (score) DESC
LIMIT 1

```

Esta consulta devuelve:

```

+-----+
| name | score |
+-----+
| Amy  | 15    |
+-----+

```

### Consideraciones y limitaciones

Si se usa UNNEST en una o más matrices de la consulta, y una de las matrices es NULL, la consulta no devuelve filas. Si se usa UNNEST en una matriz que es una cadena vacía, se devuelve la cadena vacía.

Por ejemplo, en la siguiente consulta, dado que la segunda matriz es nula, la consulta no devuelve filas.

```
SELECT
  col1,
  col2
FROM UNNEST (ARRAY ['apples','oranges','lemons']) AS t(col1)
CROSS JOIN UNNEST (ARRAY []) AS t(col2)
```

En el siguiente ejemplo, la segunda matriz se modifica para que contenga una cadena vacía. Para cada fila, la consulta devuelve el valor de col1 y una cadena vacía para el valor de col2. La cadena vacía de la segunda matriz es necesaria para que se devuelvan los valores de la primera matriz.

```
SELECT
  col1,
  col2
FROM UNNEST (ARRAY ['apples','oranges','lemons']) AS t(col1)
CROSS JOIN UNNEST (ARRAY ['']) AS t(col2)
```

## Creación de matrices a partir de subconsultas

Cree una matriz a partir de una colección de filas.

```
WITH
dataset AS (
  SELECT ARRAY[1,2,3,4,5] AS items
)
SELECT array_agg(i) AS array_items
FROM dataset
CROSS JOIN UNNEST(items) AS t(i)
```

Esta consulta devuelve:

```
+-----+
| array_items |
+-----+
| [1, 2, 3, 4, 5] |
+-----+
```

Para crear una matriz de valores únicos a partir de un conjunto de filas, utilice la palabra clave `distinct`.



```
WITH
dataset AS (
  SELECT ARRAY [1,2,2,3,3,4,5] AS items
)
SELECT array_agg(distinct i) AS array_items
FROM dataset
CROSS JOIN UNNEST(items) AS t(i)
```

Esta consulta devuelve el resultado siguiente. Tenga en cuenta que el orden no está garantizado.

```
+-----+
| array_items |
+-----+
| [1, 2, 3, 4, 5] |
+-----+
```

Para obtener más información acerca del uso de la función `array_agg`, consulte [Aggregate functions](#) (Funciones de agrupación) en la documentación de Trino.

## Filtro de matrices

Cree una matriz a partir de una colección de filas si coinciden con los criterios de filtro.

```
WITH
dataset AS (
  SELECT ARRAY[1,2,3,4,5] AS items
)
SELECT array_agg(i) AS array_items
FROM dataset
CROSS JOIN UNNEST(items) AS t(i)
WHERE i > 3
```

Esta consulta devuelve:

```
+-----+
| array_items |
+-----+
| [4, 5]      |
+-----+
```

Filtre una matriz basándose en si uno de sus elementos contiene un valor específico, como 2, como en el ejemplo siguiente:

```
WITH
dataset AS (
  SELECT ARRAY
  [
    ARRAY[1,2,3,4],
    ARRAY[5,6,7,8],
    ARRAY[9,0]
  ] AS items
)
SELECT i AS array_items FROM dataset
CROSS JOIN UNNEST(items) AS t(i)
WHERE contains(i, 2)
```

Esta consulta devuelve:

```
+-----+
| array_items |
+-----+
| [1, 2, 3, 4] |
+-----+
```

La función de **filter**

```
filter(ARRAY [list_of_values], boolean_function)
```

Puede utilizar la función `filter` en una expresión ARRAY para crear una nueva matriz que sea el subconjunto de los elementos de la *list\_of\_values* (lista de valores) para la cual el valor de *boolean\_function* es true (verdadero). La función `filter` puede ser útil cuando no se pueda utilizar la función `UNNEST`.

En el siguiente ejemplo se filtran los valores mayores que cero de la matriz `[1, 0, 5, -1]`.

```
SELECT filter(ARRAY [1,0,5,-1], x -> x>0)
```

Resultados

```
[1, 5]
```

En el siguiente ejemplo se filtran los valores no nulos en la matriz `[-1, NULL, 10, NULL]`.

```
SELECT filter(ARRAY [-1, NULL, 10, NULL], q -> q IS NOT NULL)
```

Resultados

```
[-1,10]
```

## Ordenación de matrices

Para crear una matriz ordenada con valores únicos a partir de un conjunto de filas, puede utilizar la función [array\\_sort](#) como se muestra en el siguiente ejemplo.

```
WITH
dataset AS (
  SELECT ARRAY[3,1,2,5,2,3,6,3,4,5] AS items
)
SELECT array_sort(array_agg(distinct i)) AS array_items
FROM dataset
CROSS JOIN UNNEST(items) AS t(i)
```

Esta consulta devuelve:

```
+-----+
| array_items      |
+-----+
| [1, 2, 3, 4, 5, 6] |
+-----+
```

Para obtener información sobre cómo expandir una matriz en varias filas, consulte [Aplanamiento de matrices anidadas](#).

## Uso de funciones de agregación con matrices

- Para añadir valores dentro de una matriz, utilice `SUM`, como en el ejemplo siguiente.
- Para agregar varias filas de una misma matriz, utilice `array_agg`. Para obtener más información, consulte [Creación de matrices a partir de subconsultas](#).

**Note**

ORDER BY es compatible con funciones de agregación que comienzan en la versión 2 del motor Athena.

```
WITH
dataset AS (
  SELECT ARRAY
  [
    ARRAY[1,2,3,4],
    ARRAY[5,6,7,8],
    ARRAY[9,0]
  ] AS items
),
item AS (
  SELECT i AS array_items
  FROM dataset, UNNEST(items) AS t(i)
)
SELECT array_items, sum(val) AS total
FROM item, UNNEST(array_items) AS t(val)
GROUP BY array_items;
```

En la última instrucción SELECT, en lugar de utilizar sum() y UNNEST, puede utilizar reduce() para reducir el tiempo de procesamiento y la transferencia de datos, como en el ejemplo siguiente.

```
WITH
dataset AS (
  SELECT ARRAY
  [
    ARRAY[1,2,3,4],
    ARRAY[5,6,7,8],
    ARRAY[9,0]
  ] AS items
),
item AS (
  SELECT i AS array_items
  FROM dataset, UNNEST(items) AS t(i)
)
SELECT array_items, reduce(array_items, 0 , (s, x) -> s + x, s -> s) AS total
FROM item;
```

Ambas consultas devuelven los siguientes resultados. El orden de los resultados devueltos no está garantizado.

```
+-----+
| array_items | total |
+-----+
| [1, 2, 3, 4] | 10    |
| [5, 6, 7, 8] | 26    |
| [9, 0]       | 9     |
+-----+
```

## Conversión de matrices en cadenas

Para convertir una matriz en una única cadena, utilice la función `array_join`. En el siguiente ejemplo independiente, se crea una tabla llamada `dataset` que contiene una matriz con alias llamada `words`. La consulta utiliza `array_join` para unir los elementos de la matriz en `words`, separarlos con espacios y devolver la cadena resultante en una columna con alias llamada `welcome_msg`.

```
WITH
dataset AS (
  SELECT ARRAY ['hello', 'amazon', 'athena'] AS words
)
SELECT array_join(words, ' ') AS welcome_msg
FROM dataset
```

Esta consulta devuelve:

```
+-----+
| welcome_msg          |
+-----+
| hello amazon athena |
+-----+
```

## Uso de matrices para crear mapas

Los mapas son pares de clave-valor que consisten en tipos de datos disponibles en Athena. Para crear mapas, utilice el operador `MAP` y pásele dos matrices: la primera es la columna de nombres (clave) y la segunda corresponde a los valores. Todos los valores de las matrices deben ser del

mismo tipo. Si alguno de los elementos de la matriz de valores del mapa tiene que ser de otro tipo, puede convertirlo más adelante.

## Ejemplos

En este ejemplo se selecciona a un usuario de un conjunto de datos. Se utiliza el operador MAP y se le pasan dos matrices. La primera matriz incluye valores para nombres de columna, como "first", "last" y "age". La segunda matriz está formada por valores para cada una de estas columnas, como "Bob", "Smith", "35".

```
WITH dataset AS (
  SELECT MAP(
    ARRAY['first', 'last', 'age'],
    ARRAY['Bob', 'Smith', '35']
  ) AS user
)
SELECT user FROM dataset
```

Esta consulta devuelve:

```
+-----+
| user          |
+-----+
| {last=Smith, first=Bob, age=35} |
+-----+
```

Puede recuperar valores de Map seleccionando el nombre de campo seguido de [key\_name], como en este ejemplo:

```
WITH dataset AS (
  SELECT MAP(
    ARRAY['first', 'last', 'age'],
    ARRAY['Bob', 'Smith', '35']
  ) AS user
)
SELECT user['first'] AS first_name FROM dataset
```

Esta consulta devuelve:

```
+-----+
```

```
| first_name |
+-----+
| Bob      |
+-----+
```

## Consultas de matrices con tipos complejos y estructuras anidadas

A menudo sus datos de origen contienen matrices con tipos de datos complejos y estructuras anidadas. En los ejemplos de esta sección se muestra cómo cambiar el tipo de datos de un elemento, ubicar elementos dentro de matrices y encontrar palabras clave con consultas de Athena.

- [Creación de un ROW](#)
- [Cambio de los nombres de campos en matrices mediante CAST](#)
- [Filtro de matrices mediante la notación de .](#)
- [Filtro de matrices con valores anidados](#)
- [Filtro de matrices mediante UNNEST](#)
- [Búsqueda de palabras clave en matrices mediante regexp\\_like](#)

### Creación de un ROW

#### Note

En los ejemplos de esta sección se utiliza ROW como medio para crear datos de muestra con los que trabajar. Cuando consulta tablas en Athena, no es necesario que cree tipos de datos de ROW, puesto que ya se han creado a partir del origen de datos. Cuando se utiliza CREATE\_TABLE, Athena define un STRUCT en él, lo rellena con datos y crea el tipo de datos ROW automáticamente para cada fila del conjunto de datos. El tipo de datos ROW subyacente consiste en campos con nombre de cualquier tipo de datos SQL compatible.

```
WITH dataset AS (
  SELECT
    ROW('Bob', 38) AS users
)
SELECT * FROM dataset
```

Esta consulta devuelve:

```
+-----+
| users          |
+-----+
| {field0=Bob, field1=38} |
+-----+
```

## Cambio de los nombres de campos en matrices mediante **CAST**

Para cambiar el nombre de campo en una matriz que contiene valores ROW, puede ejecutar CAST en la declaración ROW:

```
WITH dataset AS (
  SELECT
    CAST(
      ROW('Bob', 38) AS ROW(name VARCHAR, age INTEGER)
    ) AS users
)
SELECT * FROM dataset
```

Esta consulta devuelve:

```
+-----+
| users          |
+-----+
| {NAME=Bob, AGE=38} |
+-----+
```

### Note

En el ejemplo anterior declara name como VARCHAR porque este es su tipo en Presto. Si declara este STRUCT en una instrucción CREATE TABLE, use el tipo String porque Hive define este tipo de datos como String.

Filtro de matrices mediante la notación de .

En el siguiente ejemplo, seleccione el campo accountId de la columna userIdentity de una tabla de registros de AWS CloudTrail utilizando la notación de punto . Para obtener más información, consulte [Consulta de registros de AWS CloudTrail](#).



```
SELECT
  CAST(useridentity.accountid AS bigint) as newid
FROM cloudtrail_logs
LIMIT 2;
```

Esta consulta devuelve:

```
+-----+
| newid      |
+-----+
| 112233445566 |
+-----+
| 998877665544 |
+-----+
```

Para consultar una matriz de valores, emita esta consulta:

```
WITH dataset AS (
  SELECT ARRAY[
    CAST(ROW('Bob', 38) AS ROW(name VARCHAR, age INTEGER)),
    CAST(ROW('Alice', 35) AS ROW(name VARCHAR, age INTEGER)),
    CAST(ROW('Jane', 27) AS ROW(name VARCHAR, age INTEGER))
  ] AS users
)
SELECT * FROM dataset
```

Devuelve este resultado:

```
+-----+
| users                                           |
+-----+
| [{NAME=Bob, AGE=38}, {NAME=Alice, AGE=35}, {NAME=Jane, AGE=27}] |
+-----+
```

### Filtro de matrices con valores anidados

Las matrices grandes suelen contener estructuras anidadas y usted debe poder filtrar o buscar valores dentro de ellas.

Para definir un conjunto de datos para una matriz de valores que contiene un valor `BOOLEAN` anidado, emita esta consulta:

```

WITH dataset AS (
  SELECT
    CAST(
      ROW('aws.amazon.com', ROW(true)) AS ROW(hostname VARCHAR, flaggedActivity
ROW(isNew BOOLEAN))
    ) AS sites
)
SELECT * FROM dataset

```

Devuelve este resultado:

```

+-----+
| sites                                     |
+-----+
| {HOSTNAME=aws.amazon.com, FLAGGEDACTIVITY={ISNEW=true}} |
+-----+

```

A continuación, para filtrar y obtener acceso al valor BOOLEAN de este elemento, siga utilizando la notación de punto .

```

WITH dataset AS (
  SELECT
    CAST(
      ROW('aws.amazon.com', ROW(true)) AS ROW(hostname VARCHAR, flaggedActivity
ROW(isNew BOOLEAN))
    ) AS sites
)
SELECT sites.hostname, sites.flaggedactivity.isnew
FROM dataset

```

Esta consulta selecciona los campos anidados y devuelve este resultado:

```

+-----+
| hostname      | isnew |
+-----+
| aws.amazon.com | true  |
+-----+

```

## Filtro de matrices mediante **UNNEST**

Para filtrar una matriz que contiene una estructura anidada por uno de sus elementos secundarios, emita una consulta con un operador UNNEST. Para obtener más información acerca de UNNEST, consulte [Aplane Matrices anidadas](#).

Por ejemplo, esta consulta encuentra nombres de host de sitios del conjunto de datos.

```
WITH dataset AS (
  SELECT ARRAY[
    CAST(
      ROW('aws.amazon.com', ROW(true)) AS ROW(hostname VARCHAR, flaggedActivity
ROW(isNew BOOLEAN))
    ),
    CAST(
      ROW('news.cnn.com', ROW(false)) AS ROW(hostname VARCHAR, flaggedActivity
ROW(isNew BOOLEAN))
    ),
    CAST(
      ROW('netflix.com', ROW(false)) AS ROW(hostname VARCHAR, flaggedActivity ROW(isNew
BOOLEAN))
    )
  ] as items
)
SELECT sites.hostname, sites.flaggedActivity.isNew
FROM dataset, UNNEST(items) t(sites)
WHERE sites.flaggedActivity.isNew = true
```

Devuelve:

```
+-----+
| hostname      | isnew |
+-----+
| aws.amazon.com | true  |
+-----+
```

## Búsqueda de palabras clave en matrices mediante **regexp\_like**

Los siguientes ejemplos ilustran cómo buscar en un conjunto de datos una palabra clave contenida en un elemento de una matriz mediante la función [regexp\\_like](#). Toma como entrada un patrón de expresión regular para evaluar o una lista de términos separados por una barra vertical (|), evalúa el patrón y determina si la cadena especificada lo contiene.

El patrón de expresión regular debe estar incluido dentro de la cadena y no debe coincidir con él. Para que coincida la cadena completa, escriba el patrón con `^` al principio del mismo y, `$` al final, como por ejemplo `^pattern$`.

Suponga que tiene una matriz de sitios que contienen su nombre de host y un elemento `flaggedActivity`. Este elemento incluye un elemento `ARRAY` que contiene a su vez varios elementos `MAP`, cada uno de ellos con una lista de diferentes palabras clave populares y su conteo de popularidad. Supongamos que desea encontrar una determinada palabra clave dentro de un elemento `MAP` de esta matriz.

Para buscar este conjunto de datos para sitios con una palabra clave específica, utilizamos `regexp_like` en lugar del operador `LIKE` de SQL similar, ya que la búsqueda de un gran número de palabras clave es más eficiente con `regexp_like`.

### Example Ejemplo 1: uso de `regexp_like`

La consulta en este ejemplo utiliza la función `regexp_like` para buscar términos `'politics|bigdata'`, que pueden encontrarse en valores dentro de matrices:

```
WITH dataset AS (
  SELECT ARRAY[
    CAST(
      ROW('aws.amazon.com', ROW(ARRAY[
        MAP(ARRAY['term', 'count'], ARRAY['bigdata', '10']),
        MAP(ARRAY['term', 'count'], ARRAY['serverless', '50']),
        MAP(ARRAY['term', 'count'], ARRAY['analytics', '82']),
        MAP(ARRAY['term', 'count'], ARRAY['iot', '74'])
      ])
    ) AS ROW(hostname VARCHAR, flaggedActivity ROW(flags ARRAY(MAP(VARCHAR,
  VARCHAR))) ))
  ),
  CAST(
    ROW('news.cnn.com', ROW(ARRAY[
      MAP(ARRAY['term', 'count'], ARRAY['politics', '241']),
      MAP(ARRAY['term', 'count'], ARRAY['technology', '211']),
      MAP(ARRAY['term', 'count'], ARRAY['serverless', '25']),
      MAP(ARRAY['term', 'count'], ARRAY['iot', '170'])
    ])
  ) AS ROW(hostname VARCHAR, flaggedActivity ROW(flags ARRAY(MAP(VARCHAR,
  VARCHAR))) ))
  ),
  CAST(
```

```

ROW('netflix.com', ROW(ARRAY[
  MAP(ARRAY['term', 'count'], ARRAY['cartoons', '1020']),
  MAP(ARRAY['term', 'count'], ARRAY['house of cards', '112042']),
  MAP(ARRAY['term', 'count'], ARRAY['orange is the new black', '342']),
  MAP(ARRAY['term', 'count'], ARRAY['iot', '4'])
]))
) AS ROW(hostname VARCHAR, flaggedActivity ROW(flags ARRAY(MAP(VARCHAR,
VARCHAR))) ))
)
] AS items
),
sites AS (
  SELECT sites.hostname, sites.flaggedactivity
  FROM dataset, UNNEST(items) t(sites)
)
SELECT hostname
FROM sites, UNNEST(sites.flaggedActivity.flags) t(flags)
WHERE regexp_like(flags['term'], 'politics|bigdata')
GROUP BY (hostname)

```

Esta consulta devuelve dos sitios:

```

+-----+
| hostname      |
+-----+
| aws.amazon.com |
+-----+
| news.cnn.com   |
+-----+

```

### Example Ejemplo 2: uso de **regexp\_like**

En la consulta del siguiente ejemplo se añade la puntuación de popularidad total de los sitios que coinciden con sus términos de búsqueda a la función `regexp_like` y, a continuación, se ordenan de puntuación más alta a más baja.

```

WITH dataset AS (
  SELECT ARRAY[
    CAST(
      ROW('aws.amazon.com', ROW(ARRAY[
        MAP(ARRAY['term', 'count'], ARRAY['bigdata', '10']),
        MAP(ARRAY['term', 'count'], ARRAY['serverless', '50']),

```

```

        MAP(ARRAY['term', 'count'], ARRAY['analytics', '82']),
        MAP(ARRAY['term', 'count'], ARRAY['iot', '74'])
    ])
    ) AS ROW(hostname VARCHAR, flaggedActivity ROW(flags ARRAY(MAP(VARCHAR,
VARCHAR))) ))
),
CAST(
    ROW('news.cnn.com', ROW(ARRAY[
        MAP(ARRAY['term', 'count'], ARRAY['politics', '241']),
        MAP(ARRAY['term', 'count'], ARRAY['technology', '211']),
        MAP(ARRAY['term', 'count'], ARRAY['serverless', '25']),
        MAP(ARRAY['term', 'count'], ARRAY['iot', '170'])
    ])
    ) AS ROW(hostname VARCHAR, flaggedActivity ROW(flags ARRAY(MAP(VARCHAR,
VARCHAR))) ))
),
CAST(
    ROW('netflix.com', ROW(ARRAY[
        MAP(ARRAY['term', 'count'], ARRAY['cartoons', '1020']),
        MAP(ARRAY['term', 'count'], ARRAY['house of cards', '112042']),
        MAP(ARRAY['term', 'count'], ARRAY['orange is the new black', '342']),
        MAP(ARRAY['term', 'count'], ARRAY['iot', '4'])
    ])
    ) AS ROW(hostname VARCHAR, flaggedActivity ROW(flags ARRAY(MAP(VARCHAR,
VARCHAR))) ))
)
] AS items
),
sites AS (
    SELECT sites.hostname, sites.flaggedactivity
    FROM dataset, UNNEST(items) t(sites)
)
SELECT hostname, array_agg(flags['term']) AS terms, SUM(CAST(flags['count'] AS
INTEGER)) AS total
FROM sites, UNNEST(sites.flaggedActivity.flags) t(flags)
WHERE regexp_like(flags['term'], 'politics|bigdata')
GROUP BY (hostname)
ORDER BY total DESC

```

Esta consulta devuelve dos sitios:

```

+-----+
| hostname      | terms      | total |

```

```
+-----+-----+
| news.cnn.com | politics | 241 |
+-----+-----+
| aws.amazon.com | bigdata | 10 |
+-----+-----+
```

## Consulta de datos geoespaciales

Los datos geoespaciales contienen identificadores que especifican la posición geográfica de un objeto. Este tipo de datos puede corresponder a informes meteorológicos, direcciones de mapas, tweets con posiciones geográficas, ubicaciones de almacenes y rutas de líneas aéreas. Los datos geoespaciales desempeñan un papel importante en el análisis, los informes y las previsiones de negocio.

Los identificadores geoespaciales, como la latitud y la longitud, le permiten convertir cualquier dirección postal en un conjunto de coordenadas geográficas.

### Temas

- [¿Qué es una consulta geoespacial?](#)
- [Formatos de los datos de entrada y tipos de datos geométricos](#)
- [Funciones geoespaciales admitidas](#)
- [Ejemplos: consultas geoespaciales](#)

### ¿Qué es una consulta geoespacial?

Las consultas geoespaciales son tipos de consultas SQL especializadas admitidas en Athena. Se diferencian de las consultas SQL que no son espaciales en lo siguiente:

- Utilizan los tipos de datos geométricos especializados siguientes: `point`, `line`, `multiline`, `polygon` y `multipolygon`.
- Expresan relaciones entre tipos de datos geométricos como `distance`, `equals`, `crosses`, `touches`, `overlaps`, `disjoint` y otros.

Con las consultas geoespaciales en Athena, puede ejecutar las operaciones siguientes y otras similares:

- Encontrar la distancia entre dos puntos.

- Comprobar si un área (polígono) contiene otra área.
- Comprobar si una línea cruza o toca otra línea u otro polígono.

Por ejemplo, para obtener un tipo de datos geométricos `point` a partir de valores de tipo `double` para las coordenadas geográficas del Monte Rainier en Athena, utilice la función geoespacial `ST_Point (longitude, latitude)`, como en el siguiente ejemplo.

```
ST_Point(-121.7602, 46.8527)
```

## Formatos de los datos de entrada y tipos de datos geométricos

Para utilizar funciones geoespaciales en Athena, introduzca los datos en formato WKT o bien utilice el SerDe JSON de Hive. También puede utilizar los tipos de datos geométricos compatibles con Athena.

### Formatos de los datos de entrada

Para gestionar las consultas geoespaciales, Athena admite datos de entrada con los formatos de datos siguientes:

- WKT (Well-known Text). En Athena, WKT se representa como `varchar(x)` o un tipo de datos `string`.
- Datos geoespaciales con codificación JSON. Para analizar archivos JSON con datos geoespaciales y crear tablas para ellos, Athena utiliza el [SerDe JSON de Hive](#). Para obtener más información sobre el uso de este SerDe en Athena, consulte [Bibliotecas de SerDe JSON](#).

### Tipos de datos geométricos

Para gestionar las consultas geoespaciales, Athena admite los tipos de datos geométricos especializados siguientes:

- `point`
- `line`
- `polygon`
- `multiline`
- `multipolygon`



## Funciones geospaciales admitidas

Las funciones geospaciales disponibles en Athena dependen de la versión del motor que utilice.

- Para obtener información sobre las funciones geospaciales de la versión 3 del motor Athena, consulte [Geospatial functions](#) (Funciones geospaciales) en la documentación de Trino.
- Para obtener una lista de los cambios de nombre de la función y las nuevas funciones en la versión 2 del motor Athena, consulte [Cambios de nombre de función geoespacial y nuevas funciones en la versión 2 del motor Athena](#).

Para obtener más información acerca del control de versiones del motor Athena, consulte [Control de versiones del motor Athena](#).

### Temas

- [Funciones geospaciales en la versión 3 del motor Athena](#)
- [Funciones geospaciales en la versión 2 del motor Athena](#)

### Funciones geospaciales en la versión 3 del motor Athena

Para obtener información sobre las funciones geospaciales de la versión 3 del motor Athena, consulte [Geospatial functions](#) (Funciones geospaciales) en la documentación de Trino.

### Funciones geospaciales en la versión 2 del motor Athena

En este tema se incluyen las funciones geospaciales ESRI compatibles con la versión 2 del motor Athena. Para obtener más información acerca de las versiones de motor Athena, consulte [Control de versiones del motor Athena](#).

### Cambios en la versión 2 del motor Athena

- Los tipos de entrada y salida de algunas funciones han cambiado. En particular, el tipo VARBINARY ya no se admite directamente para la entrada. Para obtener más información, consulte [Cambios en las funciones geospaciales](#).
- Los nombres de algunas funciones geospaciales han cambiado. Para obtener más información, consulte [Cambios de nombre de la función geoespacial en la versión 2 del motor Athena](#).
- Se han agregado nuevas funciones. Para obtener más información, consulte [Nuevas funciones geospaciales en la versión 2 del motor Athena](#).

Athena es compatible con los siguientes tipos de funciones geoespaciales:

- [Funciones del constructor](#)
- [Funciones de relaciones geoespaciales](#)
- [Funciones de operación](#)
- [Funciones del descriptor de acceso](#)
- [Funciones de agregación](#)
- [Funciones de mosaicos de Bing](#)

## Funciones del constructor

Utilice las funciones del constructor para obtener representaciones binarias de tipos de datos geométricos `point`, `line` o `polygon`. También puede utilizar estas funciones para convertir datos binarios a texto y obtener valores binarios para los datos geométricos que se expresan como Well-Known Text (WKT).

### **ST\_AsBinary(geometry)**

Devuelve un tipo de datos binarios de longitud variable (`varbinary`) que contiene la representación WKB de la geometría especificada. Ejemplo:

```
SELECT ST_AsBinary(ST_Point(-158.54, 61.56))
```

### **ST\_AsText(geometry)**

Convierte todos los [tipos de datos geométricos](#) especificados en texto. Devuelve un valor en un tipo de datos `varchar`, que es una representación WKT del tipo de datos geométricos. Ejemplo:

```
SELECT ST_AsText(ST_Point(-158.54, 61.56))
```

### **ST\_GeomAsLegacyBinary(geometry)**

Devuelve un `varbinary` heredado de la geometría especificada. Ejemplo:

```
SELECT ST_GeomAsLegacyBinary(ST_Point(-158.54, 61.56))
```

## **ST\_GeometryFromText(varchar)**

Convierte texto en formato WKT en un tipo de datos geométricos. Devuelve un valor en un tipo de datos geométricos. Ejemplo:

```
SELECT ST_GeometryFromText(ST_AsText(ST_Point(1, 2)))
```

## **ST\_GeomFromBinary(varbinary)**

Devuelve un objeto de tipo geométrico de una representación WKB. Ejemplo:

```
SELECT ST_GeomFromBinary(ST_AsBinary(ST_Point(-158.54, 61.56)))
```

## **ST\_GeomFromLegacyBinary(varbinary)**

Devuelve un objeto de tipo geométrico de un tipo de varbinary heredado. Ejemplo:

```
SELECT ST_GeomFromLegacyBinary(ST_GeomAsLegacyBinary(ST_Point(-158.54, 61.56)))
```

## **ST\_LineFromText(varchar)**

Devuelve un valor en la línea del [tipo de datos geométricos](#). Ejemplo:

```
SELECT ST_Line('linestring(1 1, 2 2, 3 3)')
```

## **ST\_LineString(array(point))**

Devuelve un tipo geométrico `LineString` formado a partir de una matriz de tipos geométricos de punto. Si hay menos de dos puntos no vacíos en la matriz especificada, se devuelve una `LineString` vacía. Arroja una excepción si cualquier elemento de la matriz es nulo, está vacío o es el mismo que el anterior. La geometría devuelta puede no ser simple. Según la entrada especificada, la geometría devuelta puede intersectarse a sí misma o contener vértices duplicados. Ejemplo:

```
SELECT ST_LineString(ARRAY[ST_Point(-158.54, 61.56), ST_Point(-158.55, 61.56)])
```

## **ST\_MultiPoint(array(point))**

Devuelve un objeto geométrico `MultiPoint` formado a partir de los puntos especificados. Devuelve un valor nulo si la matriz especificada está vacía. Arroja una excepción si cualquier elemento de la

matriz es nulo o está vacío. La geometría devuelta puede no ser simple y puede contener puntos duplicados si la matriz especificada tiene duplicados. Ejemplo:

```
SELECT ST_MultiPoint(ARRAY[ST_Point(-158.54, 61.56), ST_Point(-158.55, 61.56)])
```

### **ST\_Point(double, double)**

Devuelve un objeto `point` de tipo geométrico. Para los valores de datos de entrada a esta función, utilice valores geométricos, como, por ejemplo, el sistema de coordenadas cartesiano universal transversal de Mercator (UTM) o unidades de mapas geográficos (longitud y latitud) en grados decimales. Los valores de longitud y latitud utilizan el sistema geodésico mundial, también conocido como WGS 1984 o EPSG:4326. WGS 1984 es el sistema de coordenadas utilizado por el Sistema de posicionamiento global (GPS).

Por ejemplo, en la siguiente notación, las coordenadas del mapa se especifican en longitud y latitud, y el valor `.072284`, que es la distancia del búfer, se especifica en unidades angulares como grados decimales:

```
SELECT ST_Buffer(ST_Point(-74.006801, 40.705220), .072284)
```

### Sintaxis:

```
SELECT ST_Point(longitude, latitude) FROM earthquakes LIMIT 1
```

En el ejemplo siguiente se utilizan coordenadas de longitud y de latitud específicas:

```
SELECT ST_Point(-158.54, 61.56)
FROM earthquakes
LIMIT 1
```

En el siguiente ejemplo se utilizan coordenadas de longitud y de latitud específicas:

```
SELECT ST_Point(-74.006801, 40.705220)
```

El siguiente ejemplo utiliza la función `ST_AsText` para obtener la geometría de WKT:

```
SELECT ST_AsText(ST_Point(-74.006801, 40.705220)) AS WKT
```

## ST\_Polygon(varchar)

Usando la secuencia de las ordenadas proporcionadas en el sentido de las agujas del reloj, de izquierda a derecha, devuelve un `polyon` de [tipo de datos geométrico](#). Desde la versión 2 del motor Athena, solo se aceptan polígonos como entradas. Ejemplo:

```
SELECT ST_Polygon('polygon ((1 1, 1 4, 4 4, 4 1))')
```

## to\_geometry(sphericalGeography)

Devuelve un objeto geométrico del objeto geográfico esférico especificado. Ejemplo:

```
SELECT to_geometry(to_spherical_geography(ST_Point(-158.54, 61.56)))
```

## to\_spherical\_geography(geometry)

Devuelve un objeto geográfico esférico de la geometría especificada. Utilice esta función para convertir un objeto geométrico en un objeto geográfico esférico en el campo del radio de la Tierra. Esta función se puede utilizar solo en geometrías POINT, MULTIPOINT, LINESTRING, MULTILINESTRING, POLYGON y MULTIPOLYGON definidas en un espacio 2D o un GEOMETRYCOLLECTION de tales geometrías. Para cada punto de la geometría especificada, la función verifica que `point.x` esté dentro de `[-180.0, 180.0]` y `point.y` dentro de `[-90.0, 90.0]`. La función utiliza estos puntos como grados de longitud y latitud para construir la forma del resultado `sphericalGeography`.

Ejemplo:

```
SELECT to_spherical_geography(ST_Point(-158.54, 61.56))
```

## Funciones de relaciones geoespaciales

Las siguientes funciones expresan relaciones entre dos geometrías diferentes que especifica como entrada y que devuelven resultados de tipo `boolean`. El orden en que especifique el par de geometrías es importante: el primer valor de geometría se denomina geometría izquierda, mientras que el segundo valor de geometría se denomina geometría derecha.

Estas funciones devuelven:

- TRUE si y solo si se cumple la relación descrita por la función.

- FALSE si y solo si no se cumple la relación descrita por la función.

### **ST\_Contains(geometry, geometry)**

Devuelve TRUE si y solo si la geometría izquierda contiene la geometría derecha. Ejemplos:

```
SELECT ST_Contains('POLYGON((0 2,1 1,0 -1,0 2))', 'POLYGON((-1 3,2 1,0 -3,-1 3))')
```

```
SELECT ST_Contains('POLYGON((0 2,1 1,0 -1,0 2))', ST_Point(0, 0))
```

```
SELECT ST_Contains(ST_GeometryFromText('POLYGON((0 2,1 1,0 -1,0 2))'),  
ST_GeometryFromText('POLYGON((-1 3,2 1,0 -3,-1 3))'))
```

### **ST\_Crosses(geometry, geometry)**

Devuelve TRUE si y solo si la geometría izquierda cruza la geometría derecha. Ejemplo:

```
SELECT ST_Crosses(ST_Line('linestring(1 1, 2 2)'), ST_Line('linestring(0 1, 2 2)'))
```

### **ST\_Disjoint(geometry, geometry)**

Devuelve TRUE si y solo si la intersección de la geometría izquierda y la geometría derecha está vacía. Ejemplo:

```
SELECT ST_Disjoint(ST_Line('linestring(0 0, 0 1)'), ST_Line('linestring(1 1, 1 0)'))
```

### **ST\_Equals(geometry, geometry)**

Devuelve TRUE si y solo si la geometría izquierda es igual a la geometría derecha. Ejemplo:

```
SELECT ST_Equals(ST_Line('linestring( 0 0, 1 1)'), ST_Line('linestring(1 3, 2 2)'))
```

### **ST\_Intersects(geometry, geometry)**

Devuelve TRUE si y solo si la geometría izquierda hace intersección con la geometría derecha. Ejemplo:

```
SELECT ST_Intersects(ST_Line('linestring(8 7, 7 8)'), ST_Polygon('polygon((1 1, 4 1, 4  
4, 1 4))'))
```

## **ST\_Overlaps(geometry, geometry)**

Devuelve TRUE si y solo si la geometría izquierda se solapa con la geometría derecha. Ejemplo:

```
SELECT ST_Overlaps(ST_Polygon('polygon((2 0, 2 1, 3 1))'), ST_Polygon('polygon((1 1, 1 4, 4 4, 4 1))'))
```

## **ST\_Relate(geometry, geometry, varchar)**

Devuelve TRUE si y solo si la geometría izquierda tiene la relación especificada de modelo de nueve intersecciones dimensionalmente extendido ([DE-9IM](#)) con la geometría derecha. La tercera entrada (varchar) toma la relación. Ejemplo:

```
SELECT ST_Relate(ST_Line('linestring(0 0, 3 3)'), ST_Line('linestring(1 1, 4 4)'), 'T*****')
```

## **ST\_Touches(geometry, geometry)**

Devuelve TRUE si y solo si la geometría izquierda toca la geometría derecha.

Ejemplo:

```
SELECT ST_Touches(ST_Point(8, 8), ST_Polygon('polygon((1 1, 1 4, 4 4, 4 1))'))
```

## **ST\_Within(geometry, geometry)**

Devuelve TRUE si y solo si la geometría izquierda está dentro de la geometría derecha.

Ejemplo:

```
SELECT ST_Within(ST_Point(8, 8), ST_Polygon('polygon((1 1, 1 4, 4 4, 4 1))'))
```

## Funciones de operación

Utilice funciones de operación para llevar a cabo operaciones en valores de tipos de datos de geometría. Por ejemplo, puede obtener los límites de un único tipo de datos geométrico; las intersecciones entre dos tipos de datos geométricos; la diferencia entre las geometrías izquierda y derecha, donde cada geometría es del mismo tipo de datos; o un búfer o anillo exterior en torno a un tipo de datos geométrico concreto.

## **geometry\_union(array(geometry))**

Devuelve una geometría que representa la unión del conjunto de puntos de las geometrías especificadas. Ejemplo:

```
SELECT geometry_union(ARRAY[ST_Point(-158.54, 61.56), ST_Point(-158.55, 61.56)])
```

## **ST\_Boundary(geometry)**

Toma como entrada uno de los tipos de datos geométricos y devuelve el tipo de datos geométrico boundary.

Ejemplos:

```
SELECT ST_Boundary(ST_Line('linestring(0 1, 1 0)'))
```

```
SELECT ST_Boundary(ST_Polygon('polygon((1 1, 1 4, 4 4, 4 1))'))
```

## **ST\_Buffer(geometry, double)**

Toma como una entrada uno de los tipos de datos geométricos, como, por ejemplo, punto, línea, polígono, varias líneas o varios polígonos y una distancia como el tipo double. Devuelve el tipo de datos geométricos delimitado por la distancia especificada (o radio). Ejemplo:

```
SELECT ST_Buffer(ST_Point(1, 2), 2.0)
```

En el siguiente ejemplo, las coordenadas del mapa se especifican en longitud y latitud, y el valor .072284, que es la distancia del búfer, se especifica en unidades angulares como grados decimales:

```
SELECT ST_Buffer(ST_Point(-74.006801, 40.705220), .072284)
```

## **ST\_Difference(geometry, geometry)**

Devuelve una geometría de la diferencia entre la geometría izquierda y la geometría derecha.

Ejemplo:

```
SELECT ST_AsText(ST_Difference(ST_Polygon('polygon((0 0, 0 10, 10 10, 10 0))'),  
ST_Polygon('polygon((0 0, 0 5, 5 5, 5 0))')))
```



## ST\_Envelope(geometry)

Toma como entrada line, polygon, multiline y tipos de datos geométricos multipolygon. No admite el tipo de datos geométricos point. Devuelve el sobre como una geometría, en la que el sobre es un rectángulo que rodea el tipo de datos geométricos especificado. Ejemplos:

```
SELECT ST_Envelope(ST_Line('linestring(0 1, 1 0)'))
```

```
SELECT ST_Envelope(ST_Polygon('polygon((1 1, 1 4, 4 4, 4 1))'))
```

## ST\_EnvelopeAsPts(geometry)

Devuelve una matriz de dos puntos que representan las esquinas inferior izquierda y superior derecha del polígono rectangular delimitador de una geometría. Devuelve un valor nulo si la geometría especificada está vacía. Ejemplo:

```
SELECT ST_EnvelopeAsPts(ST_Point(-158.54, 61.56))
```

## ST\_ExteriorRing(geometry)

Devuelve la geometría del anillo exterior del tipo de entrada polygon. Desde la versión 2 del motor Athena, los polígonos son las únicas geometrías aceptadas como entradas. Ejemplos:

```
SELECT ST_ExteriorRing(ST_Polygon(1,1, 1,4, 4,1))
```

```
SELECT ST_ExteriorRing(ST_Polygon('polygon ((0 0, 8 0, 0 8, 0 0), (1 1, 1 5, 5 1, 1 1))'))
```

## ST\_Intersection(geometry, geometry)

Devuelve la geometría de la intersección de la geometría izquierda y la geometría derecha. Ejemplos:

```
SELECT ST_Intersection(ST_Point(1,1), ST_Point(1,1))
```

```
SELECT ST_Intersection(ST_Line('linestring(0 1, 1 0)'), ST_Polygon('polygon((1 1, 1 4, 4 4, 4 1))'))
```

```
SELECT ST_AsText(ST_Intersection(ST_Polygon('polygon((2 0, 2 3, 3 0))'),
  ST_Polygon('polygon((1 1, 4 1, 4 4, 1 4))'))))
```

### **ST\_SymDifference(geometry, geometry)**

Devuelve la geometría de la diferencia geoméricamente simétrica entre la geometría izquierda y la geometría derecha. Ejemplo:

```
SELECT ST_AsText(ST_SymDifference(ST_Line('linestring(0 2, 2 2)'),
  ST_Line('linestring(1 2, 3 2)')))
```

### **ST\_Union(geometry, geometry)**

Devuelve un tipo de datos geométricos que representa la unión del conjunto de puntos de las geometrías especificadas. Ejemplo:

```
SELECT ST_Union(ST_Point(-158.54, 61.56),ST_LineString(array[ST_Point(1,2),
  ST_Point(3,4)]))
```

### Funciones del descriptor de acceso

Las funciones del descriptor de acceso son útiles para obtener valores de los tipos `varchar`, `bigint` o `double` a partir de diferentes tipos de datos `geometry`, donde `geometry` es cualquiera de los tipos de datos geométricos compatibles con Athena: `point`, `line`, `polygon`, `multiline` y `multipolygon`. Por ejemplo, puede obtener una zona de un tipo de datos `polygon`, valores X e Y máximos y mínimos de un tipo de datos geométrico especificado, obtener la longitud de una `line` o recibir el número de puntos de un tipo de datos geométrico especificado.

### **geometry\_invalid\_reason(geometry)**

Devuelve, en un tipo de datos `varchar`, el motivo por el que la geometría especificada no es válida o no es simple. Si la geometría especificada no es válida ni simple, devuelve el motivo por el que no es válida. Si la geometría especificada es válida y simple, devuelve un valor nulo. Ejemplo:

```
SELECT geometry_invalid_reason(ST_Point(-158.54, 61.56))
```

### **great\_circle\_distance(latitude1, longitude1, latitude2, longitude2)**

Devuelve, como doble, la distancia de círculo grande entre dos puntos de la superficie de la Tierra en kilómetros. Ejemplo:

```
SELECT great_circle_distance(36.12, -86.67, 33.94, -118.40)
```

### **line\_locate\_point(lineString, point)**

Devuelve un doble entre 0 y 1 que representa la ubicación del punto más cercano en la cadena de línea especificada al punto especificado como una fracción de la longitud total de línea 2d.

Devuelve un valor nulo si la cadena de línea o el punto especificado está vacío o nulo. Ejemplo:

```
SELECT line_locate_point(ST_GeometryFromText('LINESTRING (0 0, 0 1)'), ST_Point(0, 0.2))
```

### **simplify\_geometry(geometry, double)**

Utiliza el [algoritmo Ramer-Douglas-Peucker](#) para devolver un tipo de datos geométricos que sea una versión simplificada de la geometría especificada. Evita la creación de geometrías derivadas (en particular, polígonos) que no sean válidas. Ejemplo:

```
SELECT simplify_geometry(ST_GeometryFromText('POLYGON ((1 0, 2 1, 3 1, 3 1, 4 1, 1 0))'), 1.5)
```

### **ST\_Area(geometry)**

Toma como entrada un tipo de datos geométrico y devuelve una zona en tipo `double`. Ejemplo:

```
SELECT ST_Area(ST_Polygon('polygon((1 1, 4 1, 4 4, 1 4))'))
```

### **ST\_Centroid(geometry)**

Toma como entrada un `polygon` de [tipo de datos geométricos](#) y devuelve un `point` que es el centro del sobre del polígono. Ejemplos:

```
SELECT ST_Centroid(ST_GeometryFromText('polygon ((0 0, 3 6, 6 0, 0 0))'))
```

```
SELECT ST_AsText(ST_Centroid(ST_Envelope(ST_GeometryFromText('POINT (53 27)'))))
```

### **ST\_ConvexHull(geometry)**

Devuelve un tipo de datos geométricos que es la geometría convexa más pequeña que encierra todas las geometrías de la entrada especificada. Ejemplo:

```
SELECT ST_ConvexHull(ST_Point(-158.54, 61.56))
```

### **ST\_CoordDim(geometry)**

Toma como entrada uno de los [tipos de datos geométricos](#) admitidos y devuelve el recuento de componentes de coordenadas en el tipo `tinyint`. Ejemplo:

```
SELECT ST_CoordDim(ST_Point(1.5,2.5))
```

### **ST\_Dimension(geometry)**

Toma como entrada uno de los [tipos de datos geométricos](#) admitidos y devuelve la dimensión espacial de una geometría en tipo `tinyint`. Ejemplo:

```
SELECT ST_Dimension(ST_Polygon('polygon((1 1, 4 1, 4 4, 1 4))'))
```

### **ST\_Distance(geometry, geometry)**

Devuelve, en función de la referencia espacial, un doble que contiene la distancia cartesiana mínima bidimensional entre dos geometrías en unidades proyectadas. Desde la versión 2 del motor Athena, se devuelve un valor nulo si una de las entradas es una geometría vacía. Ejemplo:

```
SELECT ST_Distance(ST_Point(0.0,0.0), ST_Point(3.0,4.0))
```

### **ST\_Distance(sphericalGeography, sphericalGeography)**

Devuelve, como doble, la distancia de círculo grande entre dos puntos geográficos esféricos en metros. Ejemplo:

```
SELECT ST_Distance(to_spherical_geography(ST_Point(61.56, -86.67)),to_spherical_geography(ST_Point(61.56, -86.68)))
```

### **ST\_EndPoint(geometry)**

Devuelve el último punto de un tipo de datos geométricos `line` en un tipo de datos geométricos `point`. Ejemplo:

```
SELECT ST_EndPoint(ST_Line('linestring(0 2, 2 2)'))
```

## ST\_Geometries(geometry)

Devuelve una matriz de geometrías en la colección especificada. Si la geometría especificada no es una geometría múltiple, devuelve una matriz de un elemento. Si la geometría especificada está vacía, devuelve un valor nulo.

Por ejemplo, dado un objeto `MultiLineString`, `ST_Geometries` crea una matriz de objetos `LineString`. Dado un objeto `GeometryCollection`, `ST_Geometries` devuelve una matriz no aplanada de sus componentes. Ejemplo:

```
SELECT ST_Geometries(GEOMETRYCOLLECTION(MULTIPOINT(0 0, 1 1),
GEOMETRYCOLLECTION(MULTILINESTRING((2 2, 3 3)))))
```

Resultado:

```
array[MULTIPOINT(0 0, 1 1),GEOMETRYCOLLECTION(MULTILINESTRING((2 2, 3 3)))]
```

## ST\_GeometryN(geometry, index)

Devuelve, como tipo de datos geométricos, el elemento geométrico en un índice entero especificado. Los índices comienzan en 1. Si la geometría especificada es una colección de geometrías (por ejemplo, un objeto `GEOMETRYCOLLECTION` o `MULTI*`), devuelve la geometría en el índice especificado. Si el índice especificado es menor que 1 o mayor que el número total de elementos de la colección, devuelve un valor nulo. Para encontrar el número total de elementos, utilice [ST\\_NumGeometries](#). Geometrías singulares (por ejemplo, `POINT`, `LINestring` o `POLYGON`) se tratan como colecciones de un elemento. Las geometrías vacías se tratan como colecciones vacías. Ejemplo:

```
SELECT ST_GeometryN(ST_Point(-158.54, 61.56),1)
```

## ST\_GeometryType(geometry)

Devuelve, como `varchar`, el tipo de la geometría. Ejemplo:

```
SELECT ST_GeometryType(ST_Point(-158.54, 61.56))
```

## ST\_InteriorRingN(geometry, index)

Devuelve el elemento de anillo interior en el índice especificado (los índices comienzan en 1). Si el índice dado es menor que 1 o mayor que el número total de anillos internos en la geometría

especificada, devuelve un valor nulo. Arroja un error si la geometría especificada no es un polígono. Para encontrar el número total de elementos, utilice [ST\\_NumInteriorRing](#). Ejemplo:

```
SELECT ST_InteriorRingN(st_polygon('polygon ((0 0, 1 0, 1 1, 0 1, 0 0))'),1)
```

### **ST\_InteriorRings(geometry)**

Devuelve una matriz geométrica de todos los anillos interiores encontrados en la geometría especificada, o una matriz vacía si el polígono no tiene anillos interiores. Si la geometría especificada está vacía, devuelve un valor nulo. Arroja un error si la geometría especificada no es un polígono.

Ejemplo:

```
SELECT ST_InteriorRings(st_polygon('polygon ((0 0, 1 0, 1 1, 0 1, 0 0))'))
```

### **ST\_IsClosed(geometry)**

Toma como entrada solo los [tipos de datos geométricos](#) line y multiline. Devuelve TRUE (tipo boolean) si y solo si la línea está cerrada. Ejemplo:

```
SELECT ST_IsClosed(ST_Line('linestring(0 2, 2 2)'))
```

### **ST\_IsEmpty(geometry)**

Toma como entrada solo los [tipos de datos geométricos](#) line y multiline. Devuelve TRUE (tipo boolean) si y solo si la geometría especificada está vacía, es decir, cuando los valores de inicio y finalización de line coinciden. Ejemplo:

```
SELECT ST_IsEmpty(ST_Point(1.5, 2.5))
```

### **ST\_IsRing(geometry)**

Devuelve TRUE (tipo boolean) si y solo si el tipo line está cerrado y es simple. Ejemplo:

```
SELECT ST_IsRing(ST_Line('linestring(0 2, 2 2)'))
```

### **ST\_IsSimple(geometry)**

Devuelve verdadero si la geometría especificada no tiene puntos geométricos anómalos (por ejemplo, autointersección o autotangencia). Utilice [geometry\\_invalid\\_reason\(\)](#) para determinar por qué la geometría no es simple. Ejemplo:

```
SELECT ST_IsSimple(ST_LineString(array[ST_Point(1,2), ST_Point(3,4)]))
```

## **ST\_IsValid(geometry)**

Devuelve verdadero si y solo si la geometría especificada está bien formada. Utilice [geometry\\_invalid\\_reason\(\)](#) para determinar por qué la geometría no está bien formada.

Ejemplo:

```
SELECT ST_IsValid(ST_Point(61.56, -86.68))
```

## **ST\_Length(geometry)**

Devuelve la longitud de line en tipo double. Ejemplo:

```
SELECT ST_Length(ST_Line('linestring(0 2, 2 2)'))
```

## **ST\_NumGeometries(geometry)**

Devuelve, como número entero, el número de geometrías de la colección. Si la geometría es una colección de geometrías (por ejemplo, un objeto GEOMETRYCOLLECTION o MULTI\*), devuelve el número de geometrías. Las geometrías simples devuelven 1; las geometrías vacías devuelven 0. Una geometría vacía en un objeto GEOMETRYCOLLECTION se cuenta como una geometría. Por ejemplo, el siguiente ejemplo se evalúa en 1:

```
ST_NumGeometries(ST_GeometryFromText('GEOMETRYCOLLECTION(MULTIPOINT EMPTY)'))
```

## **ST\_NumInteriorRing(geometry)**

Devuelve el número de anillos interiores de la geometría polygon en tipo bigint. Ejemplo:

```
SELECT ST_NumInteriorRing(ST_Polygon('polygon ((0 0, 8 0, 0 8, 0 0), (1 1, 1 5, 5 1, 1 1))'))
```

## **ST\_NumPoints(geometry)**

Devuelve el número de puntos de la geometría en tipo bigint. Ejemplo:

```
SELECT ST_NumPoints(ST_Point(1.5, 2.5))
```

## **ST\_PointN(lineString, index)**

Devuelve, como tipo de datos geométricos de punto, el vértice de la cadena de línea especificada en el índice entero especificado. Los índices comienzan en 1. Si el índice dado es menor que 1 o mayor que el número total de elementos de la colección, devuelve un valor nulo. Para encontrar el número total de elementos, utilice [ST\\_NumPoints](#). Ejemplo:

```
SELECT ST_PointN(ST_LineString(array[ST_Point(1,2), ST_Point(3,4)]),1)
```

## **ST\_Points(geometry)**

Devuelve una matriz de puntos del objeto de geometría de cadena de línea especificado. Ejemplo:

```
SELECT ST_Points(ST_LineString(array[ST_Point(1,2), ST_Point(3,4)]))
```

## **ST\_StartPoint(geometry)**

Devuelve el primer punto de un tipo de datos geométricos `line` en un tipo de datos geométricos `point`. Ejemplo:

```
SELECT ST_StartPoint(ST_Line('linestring(0 2, 2 2)'))
```

## **ST\_X(point)**

Devuelve la coordenada X de un punto en tipo `double`. Ejemplo:

```
SELECT ST_X(ST_Point(1.5, 2.5))
```

## **ST\_XMax(geometry)**

Devuelve la coordenada X máxima de una geometría en tipo `double`. Ejemplo:

```
SELECT ST_XMax(ST_Line('linestring(0 2, 2 2)'))
```

## **ST\_XMin(geometry)**

Devuelve la coordenada X mínima de una geometría en tipo `double`. Ejemplo:



```
SELECT ST_XMin(ST_Line('linestring(0 2, 2 2)'))
```

### **ST\_Y(point)**

Devuelve la coordenada Y de un punto en tipo `double`. Ejemplo:

```
SELECT ST_Y(ST_Point(1.5, 2.5))
```

### **ST\_YMax(geometry)**

Devuelve la coordenada Y máxima de una geometría en tipo `double`. Ejemplo:

```
SELECT ST_YMax(ST_Line('linestring(0 2, 2 2)'))
```

### **ST\_YMin(geometry)**

Devuelve la coordenada Y mínima de una geometría en tipo `double`. Ejemplo:

```
SELECT ST_YMin(ST_Line('linestring(0 2, 2 2)'))
```

## Funciones de agregación

### **convex\_hull\_agg(geometry)**

Devuelve la geometría convexa mínima que encierra todas las geometrías pasadas como entrada.

### **geometry\_union\_agg(geometry)**

Devuelve una geometría que representa la unión del conjunto de puntos de las geometrías pasadas como entrada.

## Funciones de mosaicos de Bing

Las siguientes funciones convierten geometrías y mosaicos en el [sistema de mosaicos de Bing Maps](#) de Microsoft.

### **bing\_tile(x, y, zoom\_level)**

Devuelve un objeto de mosaico de Bing a partir de coordenadas enteras x y y y el nivel de zoom especificado. El nivel de zoom debe ser un entero del 1 al 23. Ejemplo:

```
SELECT bing_tile(10, 20, 12)
```

### **bing\_tile(quadKey)**

Devuelve un objeto de mosaico de Bing de una clave cuádruple (quadkey). Ejemplo:

```
SELECT bing_tile(bing_tile_quadkey(bing_tile(10, 20, 12)))
```

### **bing\_tile\_at(latitude, longitude, zoom\_level)**

Devuelve un objeto de mosaico de Bing en la latitud, longitud y nivel de zoom especificados. La latitud debe estar entre -85.05112878 y 85.05112878. La longitud debe estar entre -180 y 180. Los valores `latitude` y `longitude` deben ser un entero `double` y `zoom_level`. Ejemplo:

```
SELECT bing_tile_at(37.431944, -122.166111, 12)
```

### **bing\_tiles\_around(latitude, longitude, zoom\_level)**

Devuelve una matriz de mosaicos de Bing que indican el punto de latitud y longitud especificado en el nivel de zoom especificado. Ejemplo:

```
SELECT bing_tiles_around(47.265511, -122.465691, 14)
```

### **bing\_tiles\_around(latitude, longitude, zoom\_level, radius\_in\_km)**

Devuelve, en el nivel de zoom especificado, una matriz de mosaicos de Bing. La matriz contiene el conjunto mínimo de mosaicos de Bing que cubre un círculo del radio especificado en kilómetros alrededor de la latitud y longitud especificadas. Los valores `latitude`, `longitude` y `radius_in_km` son `double`; el nivel de zoom es un `integer`. Ejemplo:

```
SELECT bing_tiles_around(37.8475, 112.596667, 10, .5)
```

### **bing\_tile\_coordinates(tile)**

Devuelve las coordenadas x y y del mosaico de Bing especificado. Ejemplo:

```
SELECT bing_tile_coordinates(bing_tile_at(37.431944, -122.166111, 12))
```

## **bing\_tile\_polygon(tile)**

Devuelve la representación poligonal del mosaico de Bing especificado. Ejemplo:

```
SELECT bing_tile_polygon(bing_tile_at(47.265511, -122.465691, 4))
```

## **bing\_tile\_quadkey(tile)**

Devuelve la clave cuádruple del mosaico de Bing especificado. Ejemplo:

```
SELECT bing_tile_quadkey(bing_tile(52, 143, 10))
```

## **bing\_tile\_zoom\_level(tile)**

Devuelve el nivel de zoom del mosaico de Bing especificado como un entero. Ejemplo:

```
SELECT bing_tile_zoom_level(bing_tile(52, 143, 10))
```

## **geometry\_to\_bing\_tiles(geometry, zoom\_level)**

Devuelve el conjunto mínimo de mosaicos de Bing que cubre completamente la geometría especificada en el nivel de zoom especificado. Se admiten niveles de zoom de 1 a 23. Ejemplo:

```
SELECT geometry_to_bing_tiles(ST_Point(61.56, 58.54), 10)
```

Cambios de nombre de función geoespacial y nuevas funciones en la versión 2 del motor Athena

En esta sección se enumeran los cambios de nombre de funciones geoespaciales y funciones geoespaciales que son nuevas en la versión 2 del motor Athena. Para obtener información acerca de otros cambios en la versión 2 del motor Athena, consulte [Versión 2 del motor Athena](#).

Para obtener más información acerca del control de versiones del motor Athena, consulte [Control de versiones del motor Athena](#).

Cambios de nombre de la función geoespacial en la versión 2 del motor Athena

Los nombres de las siguientes funciones han cambiado. En algunos casos, los tipos de entrada y salida también cambiaron. Para obtener más información, visite los enlaces correspondientes.

Nombre de la función anterior	Nombre de la función desde la versión 2 del motor Athena
st_coordinate_dimension	<a href="#">ST_CoordDim</a>
st_end_point	<a href="#">ST_EndPoint</a>
st_exterior_ring	<a href="#">ST_ExteriorRing</a>
st_interior_ring_number	<a href="#">ST_NumInteriorRing</a>
st_geometry_from_text	<a href="#">ST_GeometryFromText</a>
st_is_closed	<a href="#">ST_IsClosed</a>
st_is_empty	<a href="#">ST_IsEmpty</a>
st_is_ring	<a href="#">ST_IsRing</a>
st_max_x	<a href="#">ST_XMax</a>
st_max_y	<a href="#">ST_YMax</a>
st_min_x	<a href="#">ST_XMin</a>
st_min_y	<a href="#">ST_YMin</a>
st_point_number	<a href="#">ST_NumPoints</a>
st_start_point	<a href="#">ST_StartPoint</a>
st_symmetric_difference	<a href="#">ST_SymDifference</a>

## Nuevas funciones geospaciales en la versión 2 del motor Athena

Las siguientes funciones geospaciales se introdujeron en la versión 2 del motor Athena. Para obtener más información, visite los enlaces correspondientes.

### Funciones del constructor

- [ST\\_AsBinary](#)
- [ST\\_GeomAsLegacyBinary](#)

- [ST\\_GeomFromBinary](#)
- [ST\\_GeomFromLegacyBinary](#)
- [ST\\_LineString](#)
- [ST\\_MultiPoint](#)
- [to\\_geometry](#)
- [to\\_spherical\\_geography](#)

### Funciones de operación

- [geometry\\_union](#)
- [ST\\_EnvelopeAsPts](#)
- [ST\\_Union](#)

### Funciones del descriptor de acceso

- [geometry\\_invalid\\_reason](#)
- [great\\_circle\\_distance](#)
- [line\\_locate\\_point](#)
- [simplify\\_geometry](#)
- [ST\\_ConvexHull](#)
- [ST\\_Distance \(geometría esférica\)](#)
- [ST\\_Geometries](#)
- [ST\\_GeometryN](#)
- [ST\\_GeometryType](#)
- [ST\\_InteriorRingN](#)
- [ST\\_InteriorRings](#)
- [ST\\_IsSimple](#)
- [ST\\_IsValid](#)
- [ST\\_NumGeometries](#)
- [ST\\_PointN](#)
- [ST\\_Points](#)

## Funciones de agregación

- [convex\\_hull\\_agg](#)
- [geometry\\_union\\_agg](#)

## Funciones de mosaicos de Bing

- [bing\\_tile](#)
- [bing\\_tile \(quadkey\)](#)
- [bing\\_tile\\_at](#)
- [bing\\_tiles\\_around](#)
- [bing\\_tiles\\_around \(radius\)](#)
- [bing\\_tile\\_coordinates](#)
- [bing\\_tile\\_polygon](#)
- [bing\\_tile\\_quadkey](#)
- [bing\\_tile\\_zoom\\_level](#)
- [geometry\\_to\\_bing\\_tiles](#)

## Ejemplos: consultas geoespaciales

Los ejemplos de este tema crean dos tablas a partir de datos de ejemplo disponibles en GitHub y consultan las tablas en función de los datos. Los datos de ejemplo, que son solo para fines ilustrativos y no se garantiza que sean exactos, se encuentran en los siguientes archivos:

- [earthquakes.csv](#): enumera los terremotos que se han producido en California. La tabla de ejemplo earthquakes utiliza campos de estos datos.
- [california-counties.json](#): enumera los datos del condado para el estado de California en [formato GeoJSON compatible con ESRI](#). Los datos incluyen muchos campos tales como AREA, PERIMETER, STATE, COUNTY y NAME, pero la table counties de ejemplo solo utiliza dos: Name (cadena) y BoundaryShape (binario).

### Note

Athena utiliza el `com.esri.json.hadoop.EnclosedEsriJsonInputFormat` para convertir los datos JSON a formato binario geoespacial.

El siguiente ejemplo de código crea una tabla llamada earthquakes:

```
CREATE external TABLE earthquakes
(
  earthquake_date string,
  latitude double,
  longitude double,
  depth double,
  magnitude double,
  magtype string,
  mbstations string,
  gap string,
  distance string,
  rms string,
  source string,
  eventid string
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION 's3://DOC-EXAMPLE-BUCKET/my-query-log/csv/';
```

El siguiente ejemplo de código crea una tabla llamada counties:

```
CREATE external TABLE IF NOT EXISTS counties
(
  Name string,
  BoundaryShape binary
)
ROW FORMAT SERDE 'com.esri.hadoop.hive.serde.EsriJsonSerDe'
STORED AS INPUTFORMAT 'com.esri.json.hadoop.EnclosedEsriJsonInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://DOC-EXAMPLE-BUCKET/my-query-log/json/';
```

La siguiente consulta de ejemplo utiliza la función `CROSS JOIN` en las tablas `counties` y `earthquake`. En el ejemplo se utiliza `ST_CONTAINS` para consultar los condados cuyos límites incluyen ubicaciones de terremotos, que se especifican con `ST_POINT`. La consulta agrupa los condados por su nombre, los ordena por su número y los devuelve en orden descendente.

#### Note

Desde la versión 2 del motor Athena, funciones como `ST_CONTAINS` ya no son compatibles con el tipo `VARBINARY` como entrada. Por este motivo, el ejemplo utiliza

la función [ST\\_GeomFromLegacyBinary\(varbinary\)](#) para convertir el valor binario boundaryshape en una geometría. Para obtener más información consulte [Cambios en las funciones geoespaciales](#) en la Referencia [Versión 2 del motor Athena](#).

```
SELECT counties.name,
       COUNT(*) cnt
FROM counties
CROSS JOIN earthquakes
WHERE ST_CONTAINS (ST_GeomFromLegacyBinary(counties.boundaryshape),
  ST_POINT(earthquakes.longitude, earthquakes.latitude))
GROUP BY counties.name
ORDER BY cnt DESC
```

Esta consulta devuelve:

```
+-----+
| name          | cnt |
+-----+
| Kern          | 36  |
+-----+
| San Bernardino | 35  |
+-----+
| Imperial      | 28  |
+-----+
| Inyo          | 20  |
+-----+
| Los Angeles   | 18  |
+-----+
| Riverside     | 14  |
+-----+
| Monterey      | 14  |
+-----+
| Santa Clara   | 12  |
+-----+
| San Benito    | 11  |
+-----+
| Fresno        | 11  |
+-----+
| San Diego     | 7   |
+-----+
| Santa Cruz    | 5   |
```



```
+-----+
| Ventura      | 3  |
+-----+
| San Luis Obispo | 3  |
+-----+
| Orange       | 2  |
+-----+
| San Mateo    | 1  |
+-----+
```

## Recursos adicionales de

Para obtener más ejemplos de consultas geoespaciales, consulte estas publicaciones de blog:

- [Amplíe las consultas geoespaciales en Amazon Athena con UDF y AWS Lambda](#)
- [Visualice más de 200 años de datos climáticos globales con Amazon Athena y Amazon QuickSight.](#)
- [Consulta de OpenStreetMap con Amazon Athena](#)

## Consulta de JSON

Amazon Athena le permite analizar valores con codificación JSON, extraer datos de JSON, buscar valores y determinar la longitud y el tamaño de las matrices JSON.

### Temas

- [Prácticas recomendadas para la lectura de datos JSON](#)
- [Extracción de datos de JSON](#)
- [Búsqueda de valores en matrices JSON](#)
- [Obtención de la longitud y el tamaño de las matrices JSON](#)
- [Solución de problemas de consultas JSON](#)

## Prácticas recomendadas para la lectura de datos JSON

JavaScript Object Notation (JSON) es una forma habitual de codificar estructuras de datos como texto. Numerosas aplicaciones y herramientas generan datos con codificación JSON.

En Amazon Athena puede crear tablas a partir de datos externos e incluir en ellas datos con codificación JSON. Para dichos tipos de datos de origen, utilice Athena junto con [Bibliotecas de SerDe JSON](#).

Utilice los siguientes consejos para leer datos con codificación JSON:

- Elija el SerDe adecuado, un SerDe JSON nativo, `org.apache.hive.hcatalog.data.JsonSerDe` o un SerDe OpenX, `org.openx.data.jsonserde.JsonSerDe`. Para obtener más información, consulte [Bibliotecas de SerDe JSON](#).
- Asegúrese de que cada registro codificado en JSON se represente en una línea separada, no en un formato de impresión.

#### Note

El SerDe espera que cada documento JSON esté en una sola línea de texto sin caracteres de terminación de línea que separen los campos del registro. Si el texto JSON está en formato de impresión, puede recibir un mensaje de error como `HIVE_CURSOR_ERROR: la fila no es un objeto JSON válido` o `HIVE_CURSOR_ERROR: JsonParseException: fin de entrada inesperado: marcador de cierre esperado para OBJECT cuando intenta consultar la tabla después de crearla`. Para obtener más información, consulte los [JSON Data Files](#) (Archivos de datos JSON) en la documentación de OpenX SerDE en GitHub.

- Genere los datos con codificación JSON en columnas que no distingan entre mayúsculas y minúsculas.
- Proporcione una opción para omitir los registros con formato incorrecto, como en este ejemplo.

```
CREATE EXTERNAL TABLE json_table (  
  column_a string,  
  column_b int  
)  
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'  
WITH SERDEPROPERTIES ('ignore.malformed.json' = 'true')  
LOCATION 's3://DOC-EXAMPLE-BUCKET/path/';
```

- Convierta los campos de los datos de origen que tengan un esquema indeterminado en cadenas con codificación JSON en Athena.

Cuando Athena crea tablas basadas en datos JSON, analiza los datos en función del esquema predefinido y existente. Sin embargo, puede que no todos los datos tengan un esquema predefinido. Para simplificar la administración del esquema en estos casos, a menudo es útil convertir los campos de los datos de origen que tengan un esquema indeterminado en cadenas JSON en Athena y, a continuación, usar [Bibliotecas de SerDe JSON](#).

Por ejemplo, imaginemos una aplicación IoT que publica eventos con campos comunes de diferentes sensores. Uno de los campos debe almacenar una carga personalizada específica del sensor que envía el evento. En este caso, habida cuenta de que no conocemos el esquema, recomendamos almacenar la información como una cadena codificada en JSON. Para ello, convierta los datos de la tabla de Athena a JSON, como en el ejemplo siguiente. También puede convertir datos con codificación JSON a los tipos de datos de Athena.

- [Conversión de los tipos de datos de Athena a JSON](#)
- [Conversión de JSON a los tipos de datos de Athena](#)

## Conversión de los tipos de datos de Athena a JSON

Para convertir los tipos de datos de Athena a JSON, utilice CAST.

```
WITH dataset AS (
  SELECT
    CAST('HELLO ATHENA' AS JSON) AS hello_msg,
    CAST(12345 AS JSON) AS some_int,
    CAST(MAP(ARRAY['a', 'b'], ARRAY[1,2]) AS JSON) AS some_map
)
SELECT * FROM dataset
```

Esta consulta devuelve:

```
+-----+
| hello_msg      | some_int | some_map      |
+-----+
| "HELLO ATHENA" | 12345    | {"a":1,"b":2} |
+-----+
```

## Conversión de JSON a los tipos de datos de Athena

Para convertir los datos JSON a tipos de Athena, utilice CAST.

**Note**

En este ejemplo, para indicar que las cadenas tienen codificación JSON, comience con la palabra clave JSON y use comillas simples, por ejemplo JSON '12345'

```
WITH dataset AS (
  SELECT
    CAST(JSON '"HELLO ATHENA"' AS VARCHAR) AS hello_msg,
    CAST(JSON '12345' AS INTEGER) AS some_int,
    CAST(JSON '{"a":1,"b":2}' AS MAP(VARCHAR, INTEGER)) AS some_map
)
SELECT * FROM dataset
```

Esta consulta devuelve:

```
+-----+
| hello_msg | some_int | some_map |
+-----+
| HELLO ATHENA | 12345 | {a:1,b:2} |
+-----+
```

## Extracción de datos de JSON

Es posible que tenga datos de origen que contengan cadenas con codificación JSON que no necesariamente desee deserializar en una tabla en Athena. En este caso, puede seguir ejecutando operaciones SQL en estos datos utilizando las funciones de JSON disponibles en Presto.

Considere esta cadena JSON como un conjunto de datos de ejemplo.

```
{"name": "Susan Smith",
"org": "engineering",
"projects":
  [
    {"name":"project1", "completed":false},
    {"name":"project2", "completed":true}
  ]
}
```

## Ejemplos: extracción de propiedades

Para extraer las propiedades `name` y `projects` de la cadena de JSON, utilice la función `json_extract` tal y como se muestra en el ejemplo siguiente. La función `json_extract` toma la columna que contiene la cadena de JSON y realiza una búsqueda en ella utilizando una expresión del tipo `JSONPath` con la notación de punto `.`

### Note

`JSONPath` realiza un sencillo recorrido del árbol. Utiliza el signo `$` para indicar la raíz del documento JSON, seguido de un punto y un elemento anidado directamente en la raíz como, por ejemplo, `$.name`.

```
WITH dataset AS (
  SELECT '{"name": "Susan Smith",
         "org": "engineering",
         "projects": [{"name":"project1", "completed":false},
                     {"name":"project2", "completed":true}]}'
        AS myblob
)
SELECT
  json_extract(myblob, '$.name') AS name,
  json_extract(myblob, '$.projects') AS projects
FROM dataset
```

El valor devuelto es una cadena con codificación JSON y no un tipo de datos nativo de Athena.

```
+-----+-----+
+
| name           | projects
|
+-----+-----+
+
| "Susan Smith" | [{"name":"project1","completed":false},
{"name":"project2","completed":true}] |
+-----+-----+
+
```

Para extraer el valor escalar de la cadena JSON, utilice la función `json_extract_scalar`. Es similar a `json_extract`, pero devuelve únicamente valores escalares (booleanos, números o cadenas).

### Note

No utilice la función `json_extract_scalar` en matrices, mapas o estructuras.

```
WITH dataset AS (
  SELECT '{"name": "Susan Smith",
         "org": "engineering",
         "projects": [{"name": "project1", "completed": false}, {"name": "project2",
         "completed": true}]}'
  AS myblob
)
SELECT
  json_extract_scalar(myblob, '$.name') AS name,
  json_extract_scalar(myblob, '$.projects') AS projects
FROM dataset
```

Esta consulta devuelve:

```
+-----+
| name          | projects |
+-----+
| Susan Smith  |          |
+-----+
```

Para obtener el primer elemento de la propiedad `projects` de la matriz de ejemplo, utilice la función `json_array_get` y especifique la posición del índice.

```
WITH dataset AS (
  SELECT '{"name": "Bob Smith",
         "org": "engineering",
         "projects": [{"name": "project1", "completed": false}, {"name": "project2",
         "completed": true}]}'
  AS myblob
)
SELECT json_array_get(json_extract(myblob, '$.projects'), 0) AS item
```

```
FROM dataset
```

Devuelve el valor en la posición de índice especificada en la matriz codificada con JSON.

```
+-----+
| item          |
+-----+
| {"name":"project1","completed":false} |
+-----+
```

Para obtener un tipo de cadena de Athena, utilice el operador [] dentro de una expresión JSONPath y, a continuación, utilice la función `json_extract_scalar`. Para obtener más información acerca de [], consulte [Acceso a los elementos de la matriz](#).

```
WITH dataset AS (
  SELECT '{"name": "Bob Smith",
         "org": "engineering",
         "projects": [{"name":"project1", "completed":false}, {"name":"project2",
         "completed":true}]}'
         AS myblob
)
SELECT json_extract_scalar(myblob, '$.projects[0].name') AS project_name
FROM dataset
```

Devuelve este resultado:

```
+-----+
| project_name |
+-----+
| project1     |
+-----+
```

## Búsqueda de valores en matrices JSON

Para determinar si un valor concreto existe dentro de una matriz con codificación JSON, utilice la función `json_array_contains`.

En la consulta siguiente se enumeran los nombres de los usuarios que participan en "project2".

```
WITH dataset AS (
  SELECT * FROM (VALUES
```

```

    (JSON '{"name": "Bob Smith", "org": "legal", "projects": ["project1"]}' ),
    (JSON '{"name": "Susan Smith", "org": "engineering", "projects": ["project1",
"project2", "project3"]}' ),
    (JSON '{"name": "Jane Smith", "org": "finance", "projects": ["project1",
"project2"]}' )
  ) AS t (users)
)
SELECT json_extract_scalar(users, '$.name') AS user
FROM dataset
WHERE json_array_contains(json_extract(users, '$.projects'), 'project2')

```

Esta consulta devuelve una lista de usuarios.

```

+-----+
| user   |
+-----+
| Susan Smith |
+-----+
| Jane Smith  |
+-----+

```

El siguiente ejemplo de consulta muestra los nombres de los usuarios que han completado proyectos junto con el número total de proyectos completados. Realiza las acciones siguientes:

- Utiliza instrucciones SELECT anidadas para facilitar la claridad.
- Extrae la matriz de proyectos.
- Convierte la matriz en una matriz nativa de pares de clave-valor utilizando CAST.
- Extrae cada elemento individual de la matriz utilizando el operador UNNEST.
- Filtra los valores obtenidos por proyectos completados y los cuenta.

#### Note

Cuando utilice CAST para ejecutar MAP, puede especificar el elemento clave como VARCHAR (cadena nativa en Presto), pero dejar el valor como JSON, ya que los valores de MAP son de tipos diferentes: cadena para el primer par clave-valor y booleano para el segundo.

```
WITH dataset AS (
```



```

SELECT * FROM (VALUES
  (JSON '{"name": "Bob Smith",
        "org": "legal",
        "projects": [{"name":"project1", "completed":false}]}' ),
  (JSON '{"name": "Susan Smith",
        "org": "engineering",
        "projects": [{"name":"project2", "completed":true},
                     {"name":"project3", "completed":true}]}' ),
  (JSON '{"name": "Jane Smith",
        "org": "finance",
        "projects": [{"name":"project2", "completed":true}]}' )
) AS t (users)
),
employees AS (
  SELECT users, CAST(json_extract(users, '$.projects') AS
    ARRAY(MAP(VARCHAR, JSON))) AS projects_array
  FROM dataset
),
names AS (
  SELECT json_extract_scalar(users, '$.name') AS name, projects
  FROM employees, UNNEST (projects_array) AS t(projects)
)
SELECT name, count(projects) AS completed_projects FROM names
WHERE cast(element_at(projects, 'completed') AS BOOLEAN) = true
GROUP BY name

```

Esta consulta devuelve el resultado siguiente:

```

+-----+
| name          | completed_projects |
+-----+
| Susan Smith  | 2                  |
+-----+
| Jane Smith   | 1                  |
+-----+

```

## Obtención de la longitud y el tamaño de las matrices JSON

### Ejemplo: `json_array_length`

Para obtener la longitud de una matriz con codificación JSON, use la función `json_array_length`.

```

WITH dataset AS (

```

```

SELECT * FROM (VALUES
  (JSON '{"name":
    "Bob Smith",
    "org":
    "legal",
    "projects": [{"name":"project1", "completed":false}]}' ),
  (JSON '{"name": "Susan Smith",
    "org": "engineering",
    "projects": [{"name":"project2", "completed":true},
      {"name":"project3", "completed":true}]}' ),
  (JSON '{"name": "Jane Smith",
    "org": "finance",
    "projects": [{"name":"project2", "completed":true}]}' )
) AS t (users)
)
SELECT
  json_extract_scalar(users, '$.name') as name,
  json_array_length(json_extract(users, '$.projects')) as count
FROM dataset
ORDER BY count DESC

```

Esta consulta devuelve este resultado:

```

+-----+
| name          | count |
+-----+
| Susan Smith  | 2     |
+-----+
| Bob Smith    | 1     |
+-----+
| Jane Smith   | 1     |
+-----+

```

### Ejemplo: `json_size`

Para obtener el tamaño de una matriz u objeto con codificación JSON, utilice la función `json_size` y especifique la columna que contiene la cadena JSON y la expresión JSONPath de la matriz u objeto.

```

WITH dataset AS (
  SELECT * FROM (VALUES
    (JSON '{"name": "Bob Smith", "org": "legal", "projects": [{"name":"project1",
      "completed":false}]}' ),

```

```

    (JSON '{"name": "Susan Smith", "org": "engineering", "projects":
[{"name":"project2", "completed":true},{ "name":"project3", "completed":true}]}' ),
    (JSON '{"name": "Jane Smith", "org": "finance", "projects": [{"name":"project2",
"completed":true}]}' )
) AS t (users)
)
SELECT
    json_extract_scalar(users, '$.name') as name,
    json_size(users, '$.projects') as count
FROM dataset
ORDER BY count DESC

```

Esta consulta devuelve este resultado:

```

+-----+
| name          | count |
+-----+
| Susan Smith  | 2     |
+-----+
| Bob Smith    | 1     |
+-----+
| Jane Smith   | 1     |
+-----+

```

## Solución de problemas de consultas JSON

Para obtener ayuda sobre la solución de problemas con consultas relacionadas con JSON, consulte [Errores relacionados con JSON](#) o los siguientes recursos:

- [Recibo errores cuando intento leer datos JSON en Amazon Athena](#)
- [¿Cómo soluciono el error “HIVE\\_CURSOR\\_ERROR: Row is not a valid JSON object - JSONException: Duplicate key” \(HIVE\\_CURSOR\\_ERROR: la fila no es un objeto JSON válido - JSONException: clave duplicada\) cuando se leen archivos a partir de AWS Config en Athena?](#)
- [La consulta SELECT COUNT en Amazon Athena devuelve solo un registro aunque el archivo JSON de entrada tenga varios registros.](#)
- [¿Cómo puedo ver el archivo de origen de Amazon S3 de una fila en una tabla de Athena?](#)

Véase también [Consideraciones y limitaciones de las consultas SQL en Amazon Athena.](#)

## Uso de Machine Learning (ML) con Amazon Athena

Machine Learning (ML) con Amazon Athena puede escribir instrucciones SQL que ejecutan la inferencia de Machine Learning (ML) mediante Amazon SageMaker. Esta característica simplifica el acceso a los modelos ML para el análisis de datos y elimina la necesidad de utilizar métodos de programación complejos para ejecutar la inferencia.

Para utilizar ML con Athena, defina una función de ML con Athena con la cláusula `USING EXTERNAL FUNCTION`. La función apunta al punto de enlace del modelo SageMaker que desea utilizar y especifica los nombres de variables y los tipos de datos para pasar al modelo. Las cláusulas posteriores de la consulta hacen referencia a la función para pasar valores al modelo. El modelo ejecuta la inferencia basada en los valores que pasa la consulta y, a continuación, devuelve resultados de inferencia. Para obtener más información acerca de SageMaker y cómo funcionan los puntos de enlace de SageMaker, consulte la [Guía para desarrolladores de Amazon SageMaker](#).

Para ver un ejemplo que utiliza ML con la inferencia de Athena y SageMaker para detectar un valor anómalo en un conjunto de resultados, consulte el artículo del blog de big data de AWS [Detección de valores anómalos invocando la función de inferencia de machine learning de Amazon Athena](#).

### Consideraciones y limitaciones

- **Regiones disponibles:** la característica de ML de Athena está disponible en las Regiones de AWS en las que se admite la versión 2 o una versión posterior del motor Athena.
- El punto de conexión del modelo de SageMaker debe aceptar y devolver **text/csv**: para obtener más información sobre los formatos de datos, consulte [Formatos de datos comunes para inferencia](#) en la Guía para desarrolladores de Amazon SageMaker.
- **Athena no envía encabezados CSV:** si su punto de conexión de SageMaker es `text/csv`, su controlador de entrada no debe suponer que la primera línea de la entrada es un encabezado CSV. Debido a que Athena no envía encabezados CSV, el resultado devuelto a Athena contará con una fila menos de lo esperado y provocará un error.
- **Escalado de puntos de enlace de SageMaker:** asegúrese de que el punto de enlace del modelo de SageMaker al que se hace referencia esté suficientemente escalado para las llamadas de Athena al punto de enlace. Para obtener más información, consulte [Escalado automático de modelos de SageMaker](#) en la Guía para desarrolladores de Amazon SageMaker y [CreateEndpointConfig](#) en la Referencia de la API de Amazon SageMaker.
- **Permisos de IAM:** para ejecutar una consulta que especifica una función de ML con Athena, la entidad principal de IAM que ejecuta la consulta debe tener permiso para realizar la acción

`sagemaker:InvokeEndpoint` para el punto de enlace del modelo de SageMaker al que se hace referencia. Para obtener más información, consulte [Permiso de acceso para ML con Athena](#).

- Las funciones de ML con Athena no pueden utilizarse en las cláusulas **GROUP BY** directamente.

## ML con sintaxis de Athena

La cláusula `USING EXTERNAL FUNCTION` especifica una función o varias funciones de ML con Athena a las que se puede hacer referencia mediante una instrucción `SELECT` posterior en la consulta. Defina el nombre de la función, los nombres de las variables y los tipos de datos para las variables y los valores de devolución.

### Sinopsis

La siguiente sintaxis muestra una cláusula `USING EXTERNAL FUNCTION` que especifica una función de ML con Athena.

```
USING EXTERNAL FUNCTION ml_function_name (variable1 data_type [, variable2 data_type]  
[,...])  
RETURNS data_type  
SAGEMAKER 'sagemaker_endpoint'  
SELECT ml_function_name()
```

### Parámetros

`USING EXTERNAL FUNCTION ml_function_name (variable1 data_type [, variable2 data_type][,...])`

*ML\_Function\_name* define el nombre de la función, que se puede utilizar en las cláusulas de consulta posteriores. Cada *variable data\_type* especifica una variable con nombre con su tipo de datos correspondiente, que el modelo de SageMaker puede aceptar como entrada. El tipo de datos especificado debe ser un tipo de datos de Athena admitido.

`RETURNS data_type`

*data\_type* especifica el tipo de datos SQL que *ml\_function\_name* devuelve a la consulta como salida del modelo de SageMaker.

`SAGEMAKER 'sagemaker_endpoint'`

*sagemaker\_endpoint* especifica el punto de enlace del modelo de SageMaker.

```
SELECT [...] ml_function_name(expression) [...]
```

La consulta SELECT que pasa valores a variables de función y al modelo de SageMaker para devolver un resultado. *ml\_function\_name* especifica la función definida anteriormente en la consulta, seguida de una *expresión* que se evalúa para pasar valores. Los valores que se pasan y se devuelven deben coincidir con los tipos de datos correspondientes especificados para la función en la cláusula USING EXTERNAL FUNCTION.

## Ejemplo

En el ejemplo siguiente se muestra una consulta mediante ML con Athena.

## Example

```
USING EXTERNAL FUNCTION predict_customer_registration(age INTEGER)
  RETURNS DOUBLE
  SAGEMAKER 'xgboost-2019-09-20-04-49-29-303'
SELECT predict_customer_registration(age) AS probability_of_enrolling, customer_id
  FROM "sampledb"."ml_test_dataset"
  WHERE predict_customer_registration(age) < 0.5;
```

## Ejemplos de usos de clientes

En los siguientes videos, que utilizan la versión preliminar de Machine Learning (ML) con Amazon Athena, se muestran formas en las que puede utilizar SageMaker con Athena.

### Predicción del abandono de clientes

En el siguiente video se muestra cómo combinar Athena con las capacidades de machine learning de Amazon SageMaker para predecir el abandono del cliente.

[Predict customer churn using Amazon Athena and Amazon SageMaker](#) (Predecir el abandono de clientes con Amazon Athena y Amazon SageMaker)

### Detección de botnets

En el siguiente video se muestra cómo una empresa utiliza Amazon Athena y Amazon SageMaker para detectar botnets.

[Detect botnets using Amazon Athena and Amazon SageMaker](#) (Detectar botnets con Amazon Athena y Amazon SageMaker)

## Consulta con funciones definidas por el usuario

Las funciones definidas por el usuario (UDF) en Amazon Athena le permiten crear funciones personalizadas para procesar registros o grupos de registros. Una UDF acepta parámetros, realiza el trabajo y, a continuación, devuelve un resultado.

Para utilizar una UDF en Athena, escriba una cláusula `USING EXTERNAL FUNCTION` antes de una instrucción `SELECT` en una consulta SQL. La instrucción `SELECT` hace referencia a la UDF y define las variables que se pasan a la UDF cuando se ejecuta la consulta. La consulta SQL invoca una función de Lambda mediante el tiempo de ejecución de Java cuando llama a la UDF. Las UDF se definen dentro de la función de Lambda como métodos en un paquete de implementación de Java. Se pueden definir varias UDF en el mismo paquete de implementación de Java para una función de Lambda. También se especifica el nombre de la función de Lambda en la cláusula `USING EXTERNAL FUNCTION`.

Tiene dos opciones para implementar una función de Lambda para UDF de Athena. Puede implementar la función directamente mediante Lambda, o puede usar AWS Serverless Application Repository. Para buscar funciones de Lambda existentes para UDF, puede buscar en el repositorio público AWS Serverless Application Repository o su repositorio privado y, a continuación, realizar la implementación en Lambda. También puede crear o modificar el código fuente Java, empaquetarlo en un archivo JAR e implementarlo mediante Lambda o AWS Serverless Application Repository. Para ver un ejemplo de código fuente Java y paquetes para empezar, consulte [Creación e implementación de una UDF mediante Lambda](#). Para obtener más información acerca de Lambda, consulte la [Guía para desarrolladores de AWS Lambda](#). Para obtener más información sobre AWS Serverless Application Repository, consulte la [Guía para desarrolladores de AWS Serverless Application Repository](#).

Para ver un ejemplo que utiliza UDF con Athena para traducir y analizar texto, consulte el artículo del blog de Machine Learning de AWS [Traducir y analizar texto mediante funciones SQL con Amazon Athena, Amazon Translate y Amazon Comprehend](#), o vea el [video](#).

Para ver un ejemplo del uso de UDF a fin de ampliar las consultas geoespaciales en Amazon Athena, consulte [Extend geospatial queries in Amazon Athena with UDFs and AWS Lambda](#) en el Blog de macrodatos de AWS.

## Consideraciones y limitaciones

- Funciones de Athena integradas: las funciones integradas en Athena se han diseñado para tener un alto rendimiento. Se recomienda utilizar funciones prediseñadas sobre UDF cuando sea

posible. Para obtener más información sobre funciones prediseñadas, consulte [Funciones en Amazon Athena](#).

- Únicamente UDF escalares: Athena solo admite UDF escalares, que procesan una fila a la vez y devuelven un único valor de columna. Athena pasa un lote de filas, potencialmente en paralelo, a la UDF cada vez que invoca Lambda. Al diseñar UDF y consultas, tenga en cuenta el impacto potencial en el tráfico de red de este procesamiento.
- Funciones del controlador de la FDU utilizan un formato abreviado: utilice el formato abreviado (no formato completo), para las funciones de la FDU (por ejemplo, `package .Class` en lugar de `package .Class : :method`).
- Los métodos de la FDU deben ir en minúscula: los métodos de la FDU deben estar en minúsculas; no se permite la combinación entre mayúsculas y minúsculas.
- Los métodos de UDF requieren parámetros: los métodos de UDF deben tener al menos un parámetro de entrada. Si se intenta invocar una UDF definida sin parámetros de entrada, se produce una excepción en el tiempo de ejecución. Las UDF se han diseñado para desarrollar funciones con registros de datos, pero una UDF sin argumentos no acepta datos, por lo que se produce una excepción.
- Soporte de tiempo de ejecución en Java: actualmente, las UDF de Athena admiten los tiempos de ejecución de Java 8 y Java 11 para Lambda. Para obtener más información, consulte [Creación de funciones de Lambda con Java](#) en la Guía para desarrolladores de AWS Lambda.
- Permisos de IAM: para ejecutar y crear instrucciones de consulta UDF en Athena, la entidad principal de IAM que ejecuta la consulta debe tener permiso para realizar acciones además de las funciones de Athena. Para obtener más información, consulte [Ejemplos de políticas de permisos de IAM para permitir funciones definidas por el usuario \(UDF\) de Amazon Athena](#).
- Cuotas de Lambda: las cuotas de Lambda se aplican a las UDF. Para obtener más información, consulte [Cuotas de Lambda](#) en la Guía para desarrolladores de AWS Lambda.
- Filtrado a nivel de fila: el filtrado a nivel de fila de Lake Formation no es compatible con las UDF.
- Vistas: no se pueden utilizar vistas con UDF.
- Problemas conocidos: para obtener la lista más actualizada de problemas conocidos, consulte [Limitaciones y problemas](#) en la sección `awslabs/aws-athena-query-federation` de GitHub.


## Videos

Mire los siguientes videos para obtener más información acerca del uso de las UDF en Athena.

Video: presentación de las funciones definidas por el usuario (UDF) en Amazon Athena



En el siguiente video, se muestra cómo utilizar UDF en Amazon Athena para redactar información confidencial.


 Note

La sintaxis de este video es una versión preliminar, pero los conceptos son los mismos. Utilice Athena sin el grupo de trabajo AmazonAthenaPreviewFunctionality.

### [Presentación de las funciones definidas por el usuario \(UDF\) en Amazon Athena](#)

Video : Traducir, analizar y redactar campos de texto mediante consultas SQL en Amazon Athena

En el siguiente video, se muestra cómo utilizar UDF en Amazon Athena junto con otros Servicios de AWS para traducir y analizar texto.

 Note

La sintaxis de este video es una versión preliminar, pero los conceptos son los mismos. Para obtener la sintaxis correcta, consulte la publicación del blog relacionada [Traducir, redactar y analizar texto mediante funciones SQL con Amazon Athena, Amazon Translate y Amazon Comprehend](#) en el blog de Machine Learning de AWS.

### [Traducir, analizar y redactar campos de texto mediante consultas SQL en Amazon Athena](#)

#### Sintaxis de la consulta de UDF

La cláusula `USING EXTERNAL FUNCTION` especifica una UDF o varias UDF a las que se puede hacer referencia mediante una instrucción `SELECT` posterior en la consulta. Necesita el nombre del método para la UDF y el nombre de la función de Lambda que aloja la UDF. En lugar del nombre de la función de Lambda, puede utilizar el ARN de Lambda. En escenarios de varias cuentas, se requiere el ARN de Lambda.

#### Sinopsis

```
USING EXTERNAL FUNCTION UDF_name(variable1 data_type[, variable2 data_type][,...])  
RETURNS data_type  
LAMBDA 'lambda_function_name_or_ARN'  
[, EXTERNAL FUNCTION UDF_name2(variable1 data_type[, variable2 data_type][,...])
```

```

RETURNS data_type
LAMBDA 'lambda_function_name_or_ARN'[,...]]
SELECT [...] UDF_name(expression) [, UDF_name2(expression)] [...]

```

## Parámetros

USING EXTERNAL FUNCTION ***UDF\_name(variable1 data\_type*** [, ***variable2 data\_type***][,...])

***UDF\_name*** especifica el nombre de la UDF, que debe corresponderse con un método Java dentro de la función de Lambda de referencia. Cada ***variable data\_type*** especifica una variable con nombre con su tipo de datos correspondiente, que la UDF acepta como entrada. El ***data\_type*** debe ser uno de los tipos de datos de Athena admitidos que aparecen en la siguiente tabla y asignarse al tipo de datos Java correspondiente.

Tipo de datos de Athena	Tipo de datos Java
MARCA DE TIEMPO	java.time.LocalDateTime (UTC)
FECHA	java.time.LocalDate (UTC)
TINYINT	java.lang.Byte
SMALLINT	java.lang.Short
REAL	java.lang.Float
DOBLE	java.lang.Double
DECIMAL (consulte la nota RETURNS)	java.math.BigDecimal
BIGINT	java.lang.Long
INTEGER	java.lang.Int
VARCHAR	java.lang.String
VARBINARY	byte[]

Tipo de datos de Athena	Tipo de datos Java
BOOLEAN	java.lang.Boolean
ARRAY	java.util.List
ROW	java.util.Map<String, Object>

## RETURNS *data\_type*

*data\_type* especifica el tipo de datos SQL que la UDF devuelve a la consulta como salida. Los tipos de datos de Athena enumerados en la tabla anterior son compatibles. Para el tipo de datos DECIMAL, utilice la sintaxis RETURNS DECIMAL(*precision*, *scale*) donde *precision* y *scale* son enteros.

## LAMBDA '*lambda\_function*'

*lambda\_function* especifica el nombre de la función de Lambda que se va a invocar cuando se ejecuta la UDF.

## SELECT [...] *UDF\_name*(*expression*) [...]

La consulta SELECT que pasa valores a la UDF y devuelve un resultado. *UDF\_name* especifica la UDF que se va a utilizar, seguida de una *expression* (expresión) que se evalúa para pasar valores. Los valores que se pasan y se devuelven deben coincidir con los tipos de datos correspondientes especificados para la UDF en la cláusula USING EXTERNAL FUNCTION.

## Ejemplos

Para ver ejemplos de consultas basadas en el código [AthenaUDFHandler.java](#) en GitHub, consulte la página [Conector UDF de Amazon Athena](#).

## Creación e implementación de una UDF mediante Lambda

Para crear una UDF personalizada, cree una nueva clase Java mediante la extensión de la clase `UserDefinedFunctionHandler`. El código fuente de [UserDefinedFunctionHandler.java](#) en el SDK está disponible en GitHub en el [repositorio](#) `aws-labs/aws-athena-query-federation/athena-federation-sdk`, junto con [implementaciones de UDF de ejemplo](#) que puede examinar y modificar para crear una UDF personalizada.

Los pasos de esta sección muestran cómo escribir y crear un archivo Jar de UDF personalizado mediante [Apache Maven](#) desde la línea de comandos y una implementación.

Pasos para crear una UDF personalizada para Athena mediante Maven

- [Clonación del SDK y preparación del entorno de desarrollo](#)
- [Creación del proyecto de Maven](#)
- [Adición de dependencias y complementos al proyecto de Maven](#)
- [Escritura de código Java para las UDF](#)
- [Creación del archivo JAR](#)
- [Implementación de JAR en AWS Lambda](#)

Clonación del SDK y preparación del entorno de desarrollo

Antes de comenzar, asegúrese de que git esté instalado en su sistema mediante `sudo yum install git -y`.

Para instalar el SDK de federación de consultas de AWS

- Escriba lo siguiente en la línea de comandos para clonar el repositorio de SDK. Este repositorio incluye el SDK, ejemplos y un conjunto de conectores de origen de datos. Para obtener más información sobre los conectores de origen de datos, consulte [Uso de consulta federada de Amazon Athena](#).

```
git clone https://github.com/aws-labs/aws-athena-query-federation.git
```

Para instalar los requisitos previos para este procedimiento, tenga en cuenta lo siguiente:

Si está trabajando en un equipo de desarrollo que ya tiene instalado Apache Maven, la AWS CLI y la herramienta de compilación de AWS Serverless Application Model, puede omitir este paso.

1. Desde la raíz del directorio `aws-athena-query-federation` que creó cuando realizó la clonación, ejecute el script [prepare\\_dev\\_env.sh](#) que prepara el entorno de desarrollo.
2. Actualice el shell para generar nuevas variables creadas por el proceso de instalación o reinicie la sesión de terminal.

```
source ~/.profile
```

**⚠ Important**

Si omite este paso, obtendrá errores más adelante acerca de que la herramienta de compilación de la AWS CLI o el AWS SAM no puede publicar su función de Lambda.

## Creación del proyecto de Maven

Ejecute el siguiente comando para crear el proyecto de Maven. Reemplace *groupId* por el ID único de su organización y sustituya *my-athena-udf* por el nombre de su aplicación. Para obtener más información, consulte [¿Cómo realizo mi primer proyecto de Maven?](#) en la documentación de Apache Maven.

```
mvn -B archetype:generate \  
-DarchetypeGroupId=org.apache.maven.archetypes \  
-DgroupId=groupId \  
-DartifactId=my-athena-udfs
```

## Adición de dependencias y complementos al proyecto de Maven

Agregue las siguientes configuraciones al archivo `pom.xml` del proyecto de Maven. Para ver un ejemplo, consulte el archivo [pom.xml](#) en GitHub.

```
<properties>  
  <aws-athena-federation-sdk.version>2022.47.1</aws-athena-federation-sdk.version>  
</properties>  
  
<dependencies>  
  <dependency>  
    <groupId>com.amazonaws</groupId>  
    <artifactId>aws-athena-federation-sdk</artifactId>  
    <version>${aws-athena-federation-sdk.version}</version>  
  </dependency>  
</dependencies>  
  
<build>  
  <plugins>  
    <plugin>  
      <groupId>org.apache.maven.plugins</groupId>  
      <artifactId>maven-shade-plugin</artifactId>  
      <version>3.2.1</version>
```

```
<configuration>
  <createDependencyReducedPom>>false</createDependencyReducedPom>
  <filters>
    <filter>
      <artifact>*:*</artifact>
      <excludes>
        <exclude>META-INF/*.SF</exclude>
        <exclude>META-INF/*.DSA</exclude>
        <exclude>META-INF/*.RSA</exclude>
      </excludes>
    </filter>
  </filters>
</configuration>
<executions>
  <execution>
    <phase>package</phase>
    <goals>
      <goal>shade</goal>
    </goals>
  </execution>
</executions>
</plugin>
</plugins>
</build>
```

## Escritura de código Java para las UDF

Extienda [UserDefinedFunctionHandler.java](#) para crear una nueva clase. Escriba sus UDF dentro de la clase.

En el siguiente ejemplo, se crean dos métodos Java para UDF `compress()` y `decompress()`, dentro de la clase `MyUserDefinedFunctions`.

```
*package *com.mycompany.athena.udfs;

public class MyUserDefinedFunctions
    extends UserDefinedFunctionHandler
{
    private static final String SOURCE_TYPE = "MyCompany";

    public MyUserDefinedFunctions()
    {
        super(SOURCE_TYPE);
    }
}
```

```
}

/**
 * Compresses a valid UTF-8 String using the zlib compression library.
 * Encodes bytes with Base64 encoding scheme.
 *
 * @param input the String to be compressed
 * @return the compressed String
 */
public String compress(String input)
{
    byte[] inputBytes = input.getBytes(StandardCharsets.UTF_8);

    // create compressor
    Deflater compressor = new Deflater();
    compressor.setInput(inputBytes);
    compressor.finish();

    // compress bytes to output stream
    byte[] buffer = new byte[4096];
    ByteArrayOutputStream byteArrayOutputStream = new
ByteArrayOutputStream(inputBytes.length);
    while (!compressor.finished()) {
        int bytes = compressor.deflate(buffer);
        byteArrayOutputStream.write(buffer, 0, bytes);
    }

    try {
        byteArrayOutputStream.close();
    }
    catch (IOException e) {
        throw new RuntimeException("Failed to close ByteArrayOutputStream", e);
    }

    // return encoded string
    byte[] compressedBytes = byteArrayOutputStream.toByteArray();
    return Base64.getEncoder().encodeToString(compressedBytes);
}

/**
 * Decompresses a valid String that has been compressed using the zlib compression
library.
 * Decodes bytes with Base64 decoding scheme.
 *

```

```
* @param input the String to be decompressed
* @return the decompressed String
*/
public String decompress(String input)
{
    byte[] inputBytes = Base64.getDecoder().decode((input));

    // create decompressor
    Inflater decompressor = new Inflater();
    decompressor.setInput(inputBytes, 0, inputBytes.length);

    // decompress bytes to output stream
    byte[] buffer = new byte[4096];
    ByteArrayOutputStream byteArrayOutputStream = new
ByteArrayOutputStream(inputBytes.length);
    try {
        while (!decompressor.finished()) {
            int bytes = decompressor.inflate(buffer);
            if (bytes == 0 && decompressor.needsInput()) {
                throw new DataFormatException("Input is truncated");
            }
            byteArrayOutputStream.write(buffer, 0, bytes);
        }
    }
    catch (DataFormatException e) {
        throw new RuntimeException("Failed to decompress string", e);
    }

    try {
        byteArrayOutputStream.close();
    }
    catch (IOException e) {
        throw new RuntimeException("Failed to close ByteArrayOutputStream", e);
    }

    // return decoded string
    byte[] decompressedBytes = byteArrayOutputStream.toByteArray();
    return new String(decompressedBytes, StandardCharsets.UTF_8);
}
}
```



## Creación del archivo JAR

Ejecute `mvn clean install` para crear su proyecto. Después de que se compile correctamente, se crea un archivo JAR en la carpeta `target` de su proyecto denominada `artifactId-version.jar`, donde `artifactId` es el nombre que proporcionó en el proyecto de Maven, por ejemplo, `my-athena-udfs`.

## Implementación de JAR en AWS Lambda

Tiene dos opciones para implementar su código en Lambda:

- Implementación mediante AWS Serverless Application Repository (recomendado)
- Creación de una función de Lambda desde el archivo JAR

### Opción 1: Implementación en AWS Serverless Application Repository

Al implementar el archivo JAR en AWS Serverless Application Repository, se crea un archivo YAML de plantilla AWS SAM que representa la arquitectura de la aplicación. A continuación, especifique este archivo YAML y un bucket de Amazon S3 donde los artefactos de su aplicación se carguen y estén disponibles para el AWS Serverless Application Repository. El siguiente procedimiento utiliza el script [publish.sh](#) ubicado en el directorio `athena-query-federation/tools` del SDK de Athena Query Federation que clonó anteriormente.

Para obtener más información y requisitos, consulte [Publicación de aplicaciones](#) en la Guía para desarrolladores de AWS Serverless Application Repository, [Conceptos de plantillas de AWS SAM](#) en la Guía para desarrolladores de AWS Serverless Application Model y [Publicación de aplicaciones sin servidor mediante la CLI de AWS SAM](#).

En el siguiente ejemplo se muestran los parámetros de un archivo YAML. Agregue parámetros similares a su archivo YAML y guárdelo en su directorio de proyecto. Consulte [athena-udf.yaml](#) en GitHub para ver un ejemplo completo.

```
Transform: 'AWS::Serverless-2016-10-31'  
Metadata:  
  'AWS::ServerlessRepo::Application':  
    Name: MyApplicationName  
    Description: 'The description I write for my application'  
    Author: 'Author Name'  
    Labels:  
      - athena-federation  
    SemanticVersion: 1.0.0
```

**Parameters:****LambdaFunctionName:**Description: *'The name of the Lambda function that will contain your UDFs.'*

Type: String

**LambdaTimeout:**

Description: 'Maximum Lambda invocation runtime in seconds. (min 1 - 900 max)'

Default: 900

Type: Number

**LambdaMemory:**

Description: 'Lambda memory in MB (min 128 - 3008 max).'

Default: 3008

Type: Number

**Resources:****ConnectorConfig:**

Type: 'AWS::Serverless::Function'

**Properties:**

FunctionName: !Ref LambdaFunctionName

Handler: *"full.path.to.your.handler. For example, com.amazonaws.athena.connectors.udfs.MyUDFHandler"*CodeUri: *"Relative path to your JAR file. For example, ./target/athena-udfs-1.0.jar"*Description: *"My description of the UDFs that this Lambda function enables."*

Runtime: java8

Timeout: !Ref LambdaTimeout

MemorySize: !Ref LambdaMemory

Copie el script `publish.sh` en el directorio del proyecto donde guardó el archivo YAML y ejecute el siguiente comando:

```
./publish.sh MyS3Location MyYamlFile
```

Por ejemplo, si la ubicación del bucket es `s3://DOC-EXAMPLE-BUCKET/mysarapps/athenaudf` y el archivo YAML se guardó como `my-athena-udfs.yaml`:

```
./publish.sh DOC-EXAMPLE-BUCKET/mysarapps/athenaudf my-athena-udfs
```

## Cómo crear una función de Lambda

1. Abra la consola de Lambda en <https://console.aws.amazon.com/lambda/>, elija **Create function** (Crear función), y luego elija **Browse serverless app repository** (Explorar el repositorio de aplicaciones sin servidor).

2. Elija Private applications (Aplicaciones privadas) busque su aplicación en la lista o búsquela con palabras clave y selecciónela.
3. Revise y proporcione los detalles de la aplicación y, a continuación, elija Deploy (Implementar).

Ahora puede utilizar los nombres de los métodos definidos en su archivo JAR de la función de Lambda como UDF en Athena.

## Opción 2: creación directa de una función de Lambda

También puede crear una función de Lambda directamente mediante la consola o la AWS CLI. En el siguiente ejemplo, se muestra el uso del comando de la CLI `create-function` de Lambda.

```
aws lambda create-function \  
  --function-name MyLambdaFunctionName \  
  --runtime java8 \  
  --role arn:aws:iam::1234567890123:role/my_lambda_role \  
  --handler com.mycompany.athena.udfs.MyUserDefinedFunctions \  
  --timeout 900 \  
  --zip-file fileb://./target/my-athena-udfs-1.0-SNAPSHOT.jar
```

## Consultas entre regiones

Athena admite la capacidad de consultar los datos de Amazon S3 en una Región de AWS que sea diferente de la región en la que se utiliza Athena. La consulta entre regiones puede ser una opción cuando transferir los datos no es práctico o no está permitido, o si se desea consultar datos en varias regiones. Incluso si Athena no está disponible en una región particular, los datos de esa región pueden consultarse desde otra región en la que Athena esté disponible.

Para consultar los datos que se encuentran en una región, su cuenta debe estar habilitada en esa región, incluso si los datos de Amazon S3 no pertenecen a su cuenta. En algunas regiones, como Este de EE. UU. (Ohio), el acceso a la región se habilita automáticamente cuando se crea la cuenta. Otras regiones requieren que su cuenta se “suscriba” a la región para poder utilizarla. Para obtener una lista de las regiones que requieren suscripción, consulte [Regiones disponibles](#) en la Guía del usuario de Amazon EC2 para instancias de Linux. Para obtener instrucciones específicas sobre cómo suscribirse a una región, consulte [Administración de regiones de AWS](#) en la Referencia general de Amazon Web Services.

## Consideraciones y limitaciones

- **Permisos de acceso a datos:** para consultar correctamente los datos de Amazon S3 desde Athena en las distintas regiones, su cuenta debe tener permisos para leer los datos. Si los datos que desea consultar pertenecen a otra cuenta, esta otra cuenta debe otorgarle acceso a la ubicación de Amazon S3 que contiene los datos.
- **Cargos por transferencia de datos:** se aplican cargos por transferencia de datos de Amazon S3 a las consultas entre regiones. La ejecución de una consulta puede provocar la transferencia de más datos que el tamaño del conjunto de datos. Le recomendamos que comience por probar las consultas en un subconjunto de datos y analizar los costos en [AWS Cost Explorer](#).
- **AWS Glue:** se puede utilizar AWS Glue entre regiones. Podrían aplicarse cargos adicionales por el tráfico de AWS Glue entre regiones. Para obtener más información, consulte [Create cross-account and cross-region AWS Glue connections](#) en el Blog de macrodatos de AWS.
- **Opciones de cifrado de Amazon S3:** las opciones de cifrado SSE-S3 y SSE-KMS son compatibles con las consultas entre regiones; CSE-KMS no lo es. Para obtener más información, consulte [Opciones de cifrado de Simple Storage Service \(Amazon S3\) compatibles](#).
- **Consultas federadas:** no se admite el uso de consultas federadas entre Regiones de AWS.
- **Regiones de China:** no se admiten consultas entre regiones en las siguientes regiones de China.

Siempre que se cumplan las condiciones anteriores, puede crear una tabla de Athena que apunte al valor LOCATION que especifique y consultar los datos de forma transparente. No se requiere ninguna sintaxis especial. Para obtener más información sobre cómo crear tablas en Athena, consulte [Creación de tablas en Athena](#).

## Consulta de AWS Glue Data Catalog

Debido a que muchos Servicios de AWS usan AWS Glue Data Catalog como repositorio central de metadatos, es posible que desee consultar metadatos del catálogo de datos. Para ello, puede utilizar consultas SQL en Athena. Puede utilizar Athena para consultar metadatos del catálogo de AWS Glue como bases de datos, tablas, particiones y columnas.

Para obtener metadatos del catálogo de AWS Glue, consulte la base de datos de `information_schema` en el backend de Athena. Las consultas de ejemplo de este tema muestran cómo utilizar Athena para consultar metadatos del catálogo de AWS Glue para casos de uso comunes.

### Temas

- [Consideraciones y limitaciones](#)
- [Enumeración de bases de datos y búsqueda de una base de datos especificada](#)
- [Enumeración de las tablas de una base de datos especificada y búsqueda de una tabla por nombre](#)
- [Enumeración de las particiones de una tabla específica](#)
- [Enumeración de todas las columnas de todas las tablas](#)
- [Enumeración de las columnas que tienen en común tablas específicas](#)
- [Enumeración o búsqueda de las columnas de una tabla o una vista especificadas](#)

## Consideraciones y limitaciones

- En lugar de consultar la base de datos de `information_schema`, es posible utilizar los [comandos DDL](#) de Apache Hive individuales para extraer información de metadatos de bases de datos, tablas, vistas, particiones y columnas específicas de Athena. Pero la salida está en un formato no tabular.
- Hacer una consulta a `information_schema` es más eficaz si tiene una cantidad de pequeña a moderada de metadatos de AWS Glue. Si tiene una gran cantidad de metadatos, pueden producirse errores.
- No puede utilizar `CREATE VIEW` para crear una consulta en la base de datos `information_schema`.

## Enumeración de bases de datos y búsqueda de una base de datos especificada

Los ejemplos de esta sección muestran cómo enumerar las bases de datos en metadatos por nombre de esquema.

### Example – Enumeración de las bases de datos

En la consulta de ejemplo siguiente se enumeran las bases de datos de la tabla `information_schema.schemata`.

```
SELECT schema_name
FROM   information_schema.schemata
LIMIT 10;
```

En la siguiente tabla se muestran los resultados de ejemplo.

6	alb-databas1
7	alb_original_cust
8	alblogsdatabase
9	athena_db_test
10	athena_ddl_db

### Example – Búsqueda de una base de datos especificada

En la siguiente consulta de ejemplo, `rdspostgresql` es una base de datos de ejemplo.

```
SELECT schema_name
FROM   information_schema.schemata
WHERE  schema_name = 'rdspostgresql'
```

En la siguiente tabla se muestran los resultados de ejemplo.

	schema_name
1	rdspostgresql

### Enumeración de las tablas de una base de datos especificada y búsqueda de una tabla por nombre

Para enumerar los metadatos de las tablas, puede consultar por esquema de tabla o por nombre de tabla.

#### Example – Enumeración de tablas por esquema

En la consulta siguiente se enumeran las tablas que utilizan el esquema de tabla `rdspostgresql`.

```
SELECT table_schema,
       table_name,
       table_type
```

```
FROM information_schema.tables
WHERE table_schema = 'rdspostgresql'
```

La siguiente imagen muestra un resultado de ejemplo.

	table_schema	table_name	table_type
1	rdspostgresql	rdspostgresqldb1_public_account	BASE TABLE

Example – Búsqueda de una tabla por nombre

La siguiente consulta obtiene información de metadatos para la tabla athena1.

```
SELECT table_schema,
       table_name,
       table_type
FROM information_schema.tables
WHERE table_name = 'athena1'
```

La siguiente imagen muestra un resultado de ejemplo.

	table_schema	table_name	table_type
1	predeterminada	athena1	BASE TABLE

## Enumeración de las particiones de una tabla específica

Puede utilizar `SHOW PARTITIONS table_name` para enumerar las particiones de una tabla especificada, como en el ejemplo siguiente.

```
SHOW PARTITIONS cloudtrail_logs_test2
```

También puede utilizar una consulta de metadatos de `$partitions` para enumerar los números de partición y los valores de partición de una tabla específica.

## Example – Consulta de las particiones de una tabla usando la sintaxis \$partitions

En la consulta de ejemplo siguiente, se enumeran las particiones de la tabla `cloudtrail_logs_test2` utilizando la sintaxis `$partitions`.

```
SELECT * FROM default."cloudtrail_logs_test2$partitions" ORDER BY partition_number
```

En la siguiente tabla se muestran los resultados de ejemplo.

	table_cat alog	table_sch ema	table_name	Año	Mes	Día
1	awsdataca talog	predeterm inado	cloudtrail_logs_te st2	2020	08	10
2	awsdataca talog	predeterm inado	cloudtrail_logs_te st2	2020	08	11
3	awsdataca talog	predeterm inado	cloudtrail_logs_te st2	2020	08	12

## Enumeración de todas las columnas de todas las tablas

Puede enumerar todas las columnas de todas las tablas de `AwsDataCatalog` o de todas las tablas de una base de datos específica de `AwsDataCatalog`.

- Para mostrar todas las columnas de todas las bases de datos en `AwsDataCatalog`, use la consulta `SELECT * FROM information_schema.columns`.
- Para restringir los resultados a una base de datos específica, use `table_schema='database_name'` en la cláusula `WHERE`.

## Example – Enumeración de todas las tablas en una base de datos específica

En la consulta de ejemplo siguiente se enumeran todas las columnas de todas las tablas de la base de datos `webdata`.

```
SELECT * FROM information_schema.columns WHERE table_schema = 'webdata'
```



## Enumeración de las columnas que tienen en común tablas específicas

Puede enumerar las columnas que tienen en común tablas específicas de una base de datos.

- Utilice la sintaxis `SELECT column_name FROM information_schema.columns`.
- Para la cláusula `WHERE`, utilice la sintaxis `WHERE table_name IN ('table1', 'table2')`.

Example – Enumeración de columnas en común de dos tablas de la misma base de datos

El siguiente ejemplo de consulta enumera las columnas que tienen en común las tablas `table1` y `table2`.

```
SELECT column_name
FROM information_schema.columns
WHERE table_name IN ('table1', 'table2')
GROUP BY column_name
HAVING COUNT(*) > 1;
```

## Enumeración o búsqueda de las columnas de una tabla o una vista especificadas

Puede enumerar todas las columnas de una tabla, todas las columnas de una vista o buscar una columna por nombre en una base de datos y tabla especificados.

Para enumerar las columnas, utilice una consulta `SELECT *`. En la cláusula `FROM`, especifique `information_schema.columns`. En el cláusula `WHERE`, utilice `table_schema = 'database_name'` para especificar la base de datos y `table_name = 'table_name'` para especificar la tabla o vista que tiene las columnas que desea enumerar.

Example – Enumeración de todas las columnas de una tabla especificada

En la consulta de ejemplo siguiente se enumeran todas las columnas de la tabla `rdspostgresqldb1_public_account`.

```
SELECT *
FROM information_schema.columns
WHERE table_schema = 'rdspostgresql'
      AND table_name = 'rdspostgresqldb1_public_account'
```

En la siguiente tabla se muestran los resultados de ejemplo.

	table_cat alog	table_scl ema	table_nam e	column_ me	ordinal_p osition	column_d fault	is_null le	data_t o	come o	extra_inf o
1	awsdataca talog	rdspostg esql	rdspostgr esqldb1_p ublic_acc ount	password	1		Sí	varchar		
2	awsdataca talog	rdspostg esql	rdspostgr esqldb1_p ublic_acc ount	user_id	2		Sí	intege		
3	awsdataca talog	rdspostg esql	rdspostgr esqldb1_p ublic_acc ount	created_ n	3		Sí	Marca de tiemp		
4	awsdataca talog	rdspostg esql	rdspostgr esqldb1_p ublic_acc ount	last_logi n	4		Sí	Marca de tiemp		
5	awsdataca talog	rdspostg esql	rdspostgr esqldb1_p ublic_acc ount	email	5		Sí	varchar		
6	awsdataca talog	rdspostg esql	rdspostgr esqldb1_p ublic_acc ount	usernam	6		Sí	varchar		

Example – Enumeración de las columnas de una vista especificada

En la consulta de ejemplo siguiente se enumeran todas las columnas de la base de datos default para la vista arrayview.

```
SELECT *
FROM   information_schema.columns
WHERE  table_schema = 'default'
       AND table_name = 'arrayview'
```

En la siguiente tabla se muestran los resultados de ejemplo.

	table_cat alog	table_sch ema	table_n e	column_n me	ordinal_p osition	column_d fault	is_null le	data_t yp	come o	extra_in fo
1	awsdataca talog	predeter inado	arrayvie e	searchdat e	1		Sí	varchar		
2	awsdataca talog	predeter inado	arrayvie e	sid	2		Sí	varchar		
3	awsdataca talog	predeter inado	arrayvie e	btid	3		Sí	varchar		
4	awsdataca talog	predeter inado	arrayvie e	p	4		Sí	varchar		
5	awsdataca talog	predeter inado	arrayvie e	infantpri ce	5		Sí	varchar		
6	awsdataca talog	predeter inado	arrayvie e	sump	6		Sí	varchar		
7	awsdataca talog	predeter inado	arrayvie e	journeyma pparray	7		Sí	array(va char)		

Example – Búsqueda de una columna por nombre en una base de datos y una tabla especificadas

La consulta de ejemplo siguiente busca metadatos para la columna sid en la vista arrayview de la base de datos default.

```
SELECT *
FROM   information_schema.columns
WHERE  table_schema = 'default'
```

```
AND table_name = 'arrayview'
AND column_name='sid'
```

La siguiente imagen muestra un resultado de ejemplo.

	table_cat alog	table_sch ema	table_nam e	column_r me	ordinal_p osition	column_de fault	is_nulla le	data_t o	come o	extra_inf o
1	awsdataca talog	predeterr inado	arrayview	sid	2		Sí	varcha		

## Consulta de registros de Servicio de AWS

En esta sección se incluyen varios procedimientos para utilizar Amazon Athena a fin de consultar conjuntos de datos habituales, como registros de AWS CloudTrail, de Amazon CloudFront, del Classic Load Balancer, del Application Load Balancer, registros de flujo de Amazon VPC y registros del Network Load Balancer.

Las tareas de esta sección utilizan la consola de Athena, pero también puede utilizar otras herramientas como el [controlador JDBC de Athena](#), la [AWS CLI](#) o la [Referencia de la API de Amazon Athena](#).

Para obtener información acerca del uso de AWS CloudFormation para crear automáticamente tablas de registro de Servicio de AWS, particiones y consultas de ejemplo en Athena, consulte [Automating Servicio de AWS logs table creation and querying them with Amazon Athena](#) (Automatización de la creación de tablas de registro y consulta con Amazon Athena) en el blog de macrodatos de AWS. Para obtener información sobre el uso de una biblioteca de Python para AWS Glue a fin de crear un marco común para el procesamiento de registros de Servicio de AWS y consultarlos en Athena, vea [Consultar registros de Servicio de AWS fácilmente con Amazon Athena](#).

Los temas de esta sección presuponen que ha configurado los permisos adecuados para acceder a Athena y al bucket de Amazon S3 en el que deben residir los datos que se van a consultar. Para obtener más información, consulte [Configuración](#) y [Introducción](#).

### Temas

- [Consulta de los registros de Application Load Balancer](#)
- [Consulta de los registros del Equilibrador de carga clásico](#)

- [Consultas de registros de Amazon CloudFront](#)
- [Consulta de registros de AWS CloudTrail](#)
- [Consultas de los registros de Amazon EMR](#)
- [Consulta de los registros de flujo de AWS Global Accelerator](#)
- [Consulta de los resultados de Amazon GuardDuty](#)
- [Consulta de registros de AWS Network Firewall](#)
- [Consulta de los registros del Network Load Balancer](#)
- [Consulta de los registros de consulta de Amazon Route 53 Resolver](#)
- [Consulta de registros de eventos de Amazon SES](#)
- [Consulta de los registros de flujo de Amazon VPC](#)
- [Consulta de registros de AWS WAF](#)

## Consulta de los registros de Application Load Balancer

Un Application Load Balancer aplicaciones es una opción de equilibrio de carga de Elastic Load Balancing que permite la distribución del tráfico en una implementación de microservicios con contenedores. La consulta de los registros del Equilibrador de carga de aplicación le permite ver el origen del tráfico, la latencia y los bytes transferidos a instancias de Elastic Load Balancing y desde dichas instancias y las aplicaciones de backend. Para obtener más información, consulte [Registros de acceso para el equilibrador de carga de aplicación](#) y [Registros de conexión para el equilibrador de carga de aplicación](#) en la Guía del usuario para equilibradores de carga de aplicación.

### Temas

- [Requisitos previos](#)
- [Creación de la tabla para los registros de acceso de ALB](#)
- [Creación de la tabla para los registros de acceso del ALB en Athena mediante la proyección de particiones](#)
- [Consultas de ejemplo para registros de acceso del ALB](#)
- [Creación de la tabla para los registros de conexión del ALB](#)
- [Creación de la tabla para los registros de conexión del ALB en Athena mediante la proyección de particiones](#)
- [Consultas de ejemplo para registros de conexión del ALB](#)
- [Recursos adicionales de](#)

## Requisitos previos

- Habilite el [registro de acceso](#) o el [registro de conexión](#) para que los registros del equilibrador de carga de aplicación se puedan guardar en el bucket de Amazon S3.
- Una base de datos en la que contener la tabla que creará para Athena. Para crear una base de datos, puede utilizar Athena o la consola de AWS Glue. Para obtener más información, consulte [Creación de bases de datos en Athena](#) en esta guía o [Trabajo con bases de datos en la consola de AWS Glue](#) en la Guía para desarrolladores de AWS Glue.

## Creación de la tabla para los registros de acceso de ALB

1. Copie y pegue la siguiente instrucción CREATE TABLE en el editor de consultas de la consola de Athena. Para obtener más información acerca de cómo empezar a usar la consola de Athena, consulte [Introducción](#). Sustituya la ruta de la cláusula LOCATION por la ubicación de la carpeta de registro de acceso de Amazon S3. Para obtener más información sobre la ubicación del archivo de registro de acceso, consulte [Archivos de registro de acceso](#) en la Guía del usuario para equilibradores de carga de aplicación. Para obtener información sobre cada campo del archivo de registro, consulte [Acceso a las entradas del registro](#).

### Note

La siguiente instrucción CREATE TABLE incluye las columnas añadidas recientemente `classification`, `classification_reason` y `traceability_id`. Para crear una tabla para los registros de acceso al equilibrador de carga de aplicaciones que no contenga estas entradas, elimine las columnas correspondientes de la instrucción CREATE TABLE y modifique la expresión regular en consecuencia.

```
CREATE EXTERNAL TABLE IF NOT EXISTS alb_access_logs (  
    type string,  
    time string,  
    elb string,  
    client_ip string,  
    client_port int,  
    target_ip string,  
    target_port int,  
    request_processing_time double,  
    target_processing_time double,
```

```

response_processing_time double,
elb_status_code int,
target_status_code string,
received_bytes bigint,
sent_bytes bigint,
request_verb string,
request_url string,
request_proto string,
user_agent string,
ssl_cipher string,
ssl_protocol string,
target_group_arn string,
trace_id string,
domain_name string,
chosen_cert_arn string,
matched_rule_priority string,
request_creation_time string,
actions_executed string,
redirect_url string,
lambda_error_reason string,
target_port_list string,
target_status_code_list string,
classification string,
classification_reason string,
traceability_id string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1',
  'input.regex' =
    '([^\ ]*) ([^\ ]*) ([^\ ]*) ([^\ ]*):([0-9]*) ([^\ ]*)[:-]([0-9]*) ([-\.0-9]*)
    ([-\.0-9]*) ([-\.0-9]*) (|[-0-9]*) (-|[-0-9]*) ([-0-9]*) ([-0-9]*) \"([^\ ]*) (.*) (-
    | [^\ ]*)\" \"([^\"]*)\" ([A-Z0-9-_\ ]+) ([A-Za-z0-9.-]*) ([^\ ]*) \"([^\"]*)\" \"([^\
    \"]*)\" \"([^\"]*)\" \"([^\ ]*)\" \"([^\ ]*)\" \"([^\ ]*)\" \"([^\ ]*)\" \"([^\
    \s]+?)\" \"([^\s]+)\s\" \"([^\ ]*)\" \"([^\ ]*)\" ?([^\ ]*)?( .*)?')
  LOCATION 's3://DOC-EXAMPLE-BUCKET/access-log-folder-path/'

```

2. Ejecute la consulta en la consola de Athena. Cuando la consulta finaliza, Athena registra la tabla `alb_access_logs`, dejando los datos que contiene listos para efectuar consultas.

## Creación de la tabla para los registros de acceso del ALB en Athena mediante la proyección de particiones

Dado que los registros de acceso del ALB tienen una estructura conocida, cuyo esquema de partición puede especificarse de antemano, es posible reducir el tiempo de ejecución de las consultas y automatizar la administración de particiones mediante la característica de proyección de particiones de Athena. La proyección de particiones agrega de forma automática nuevas particiones a medida que se agregan nuevos datos. Esto hace que no sea necesario agregar particiones manualmente mediante `ALTER TABLE ADD PARTITION`.

El siguiente ejemplo de instrucción `CREATE TABLE` utiliza de manera automática la proyección de particiones en registros de acceso del ALB desde una fecha especificada hasta el presente para una sola región de AWS. La instrucción se basa en el ejemplo de la sección anterior, pero agrega las cláusulas `PARTITIONED BY` y `TBLPROPERTIES` para habilitar la proyección de la partición. En las cláusulas `LOCATION` y `storage.location.template`, reemplace los marcadores de posición por valores que identifiquen la ubicación del bucket de Amazon S3 de los registros del ALB. Para obtener más información sobre la ubicación del archivo de registro de acceso, consulte [Archivos de registro de acceso](#) en la Guía del usuario para equilibradores de carga de aplicación. En `projection.day.range`, reemplace `2022/01/01` por la fecha de inicio que desee utilizar. Una vez ejecutada la consulta correctamente, puede consultar la tabla. No tiene que ejecutar `ALTER TABLE ADD PARTITION` para cargar las particiones. Para obtener información sobre cada campo del archivo de registro, consulte [Acceso a las entradas del registro](#).

```
CREATE EXTERNAL TABLE IF NOT EXISTS alb_access_logs (  
    type string,  
    time string,  
    elb string,  
    client_ip string,  
    client_port int,  
    target_ip string,  
    target_port int,  
    request_processing_time double,  
    target_processing_time double,  
    response_processing_time double,  
    elb_status_code int,  
    target_status_code string,  
    received_bytes bigint,  
    sent_bytes bigint,  
    request_verb string,  
    request_url string,
```



```

request_proto string,
user_agent string,
ssl_cipher string,
ssl_protocol string,
target_group_arn string,
trace_id string,
domain_name string,
chosen_cert_arn string,
matched_rule_priority string,
request_creation_time string,
actions_executed string,
redirect_url string,
lambda_error_reason string,
target_port_list string,
target_status_code_list string,
classification string,
classification_reason string,
traceability_id string
)
PARTITIONED BY
(
  day STRING
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1',
  'input.regex' =
    '([^\ ]*) ([^\ ]*) ([^\ ]*) ([^\ ]*):([0-9]*) ([^\ ]*)[:-]([0-9]*) ([-.\0-9]*)
  ([-.\0-9]*) ([-.\0-9]*) (|[-0-9]*) (-|[-0-9]*) ([-0-9]*) ([-0-9]*) \"([^\ ]*) (.*) (- |
  [^\ ]*)\" \"([^\"]*)\" ([A-Z0-9-_\ ]+) ([A-Za-z0-9-.\ ]*) ([^\ ]*) \"([^\"]*)\" \"([^\"]*)\"
  \"([^\"]*)\" ([-.\0-9]*) ([^\ ]*) \"([^\"]*)\" \"([^\"]*)\" \"([^\ ]*)\" \"([^\s]+?)\"
  \"([^\s]+)\s\" \"([^\ ]*)\" \"([^\ ]*)\" ?([^\ ]*)?(.*)?'
  LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/<ACCOUNT-NUMBER>/
elasticloadbalancing/<REGION>/'
TBLPROPERTIES
(
  "projection.enabled" = "true",
  "projection.day.type" = "date",
  "projection.day.range" = "2022/01/01,NOW",
  "projection.day.format" = "yyyy/MM/dd",
  "projection.day.interval" = "1",
  "projection.day.interval.unit" = "DAYS",
  "storage.location.template" = "s3://DOC-EXAMPLE-BUCKET/AWSLogs/<ACCOUNT-
NUMBER>/elasticloadbalancing/<REGION>/${day}"

```

```
)
```

Para obtener más información sobre la proyección de particiones, consulte [Proyección de particiones con Amazon Athena](#).

Consultas de ejemplo para registros de acceso del ALB

La siguiente consulta cuenta el número de solicitudes HTTP GET que el equilibrador de carga ha recibido, agrupadas en función de la dirección IP del cliente:

```
SELECT COUNT(request_verb) AS
count,
request_verb,
client_ip
FROM alb_logs
GROUP BY request_verb, client_ip
LIMIT 100;
```

Otra consulta muestra las direcciones URL visitadas por los usuarios del navegador Safari:

```
SELECT request_url
FROM alb_logs
WHERE user_agent LIKE '%Safari%'
LIMIT 10;
```

En la siguiente consulta, se muestran registros que tienen valores de código de estado de ELB superiores o iguales a 500.

```
SELECT * FROM alb_logs
WHERE elb_status_code >= 500
```

En el siguiente ejemplo, se muestra cómo analizar los registros por datetime:

```
SELECT client_ip, sum(received_bytes)
FROM alb_logs
WHERE parse_datetime(time, 'yyyy-MM-dd' 'T' 'HH:mm:ss.SSSSSS' 'Z')
BETWEEN parse_datetime('2018-05-30-12:00:00', 'yyyy-MM-dd-HH:mm:ss')
AND parse_datetime('2018-05-31-00:00:00', 'yyyy-MM-dd-HH:mm:ss')
GROUP BY client_ip;
```

La siguiente consulta busca la tabla que utiliza la proyección de la partición para todos los registros de ALB correspondientes al día especificado.

```
SELECT *
FROM alb_logs
WHERE day = '2022/02/12'
```

Creación de la tabla para los registros de conexión del ALB

1. Copie y pegue la siguiente instrucción CREATE TABLE en el editor de consultas de la consola de Athena. Para obtener más información acerca de cómo empezar a usar la consola de Athena, consulte [Introducción](#). Sustituya la ruta de la cláusula LOCATION por la ubicación de la carpeta del registro de conexiones de Amazon S3. Para obtener más información sobre la ubicación del archivo de registro de conexión, consulte [Archivos de registro de conexión](#) en la Guía del usuario para equilibradores de carga de aplicación. Para obtener información sobre cada campo del archivo de registro, consulte [Entradas del registro de conexión](#).

```
CREATE EXTERNAL TABLE IF NOT EXISTS alb_connection_logs (
    time string,
    client_ip string,
    client_port int,
    listener_port int,
    tls_protocol string,
    tls_cipher string,
    tls_handshake_latency double,
    leaf_client_cert_subject string,
    leaf_client_cert_validity string,
    leaf_client_cert_serial_number string,
    tls_verify_status string,
    traceability_id string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
    'serialization.format' = '1',
    'input.regex' =
    '^\"([^\"]*)\" ([^ ]*) ([^ ]*) ([0-9]*) ([0-9]*) ([A-Za-z0-9.-]*) ([^ ]*) ([-\.0-9]*)'
    LOCATION 's3://DOC-EXAMPLE-BUCKET/connection-log-folder-path/'
```

2. Ejecute la consulta en la consola de Athena. Cuando la consulta finaliza, Athena registra la tabla `alb_connection_logs`, dejando los datos que contiene listos para efectuar consultas.

## Creación de la tabla para los registros de conexión del ALB en Athena mediante la proyección de particiones

Dado que los registros de conexión del ALB tienen una estructura conocida, cuyo esquema de partición puede especificarse de antemano, es posible reducir el tiempo de ejecución de las consultas y automatizar la administración de particiones mediante la característica de proyección de particiones de Athena. La proyección de particiones agrega de forma automática nuevas particiones a medida que se agregan nuevos datos. Esto hace que no sea necesario agregar particiones manualmente mediante `ALTER TABLE ADD PARTITION`.

El siguiente ejemplo de instrucción `CREATE TABLE` utiliza de manera automática la proyección de particiones en registros de conexión del ALB desde una fecha especificada hasta el presente para una sola región de AWS. La instrucción se basa en el ejemplo de la sección anterior, pero agrega las cláusulas `PARTITIONED BY` y `TBLPROPERTIES` para habilitar la proyección de la partición. En las cláusulas `LOCATION` y `storage.location.template`, reemplace los marcadores de posición por valores que identifiquen la ubicación del bucket de Amazon S3 de los registros de conexión del ALB. Para obtener más información sobre la ubicación del archivo de registro de conexión, consulte [Archivos de registro de conexión](#) en la Guía del usuario para equilibradores de carga de aplicación. Para `projection.day.range`, reemplace `2023/01/01` por la fecha de inicio que quiere utilizar. Una vez ejecutada la consulta correctamente, puede consultar la tabla. No tiene que ejecutar `ALTER TABLE ADD PARTITION` para cargar las particiones. Para obtener información sobre cada campo del archivo de registro, consulte [Entradas del registro de conexión](#).

```
CREATE EXTERNAL TABLE IF NOT EXISTS alb_connection_logs (  
    time string,  
    client_ip string,  
    client_port int,  
    listener_port int,  
    tls_protocol string,  
    tls_cipher string,  
    tls_handshake_latency double,  
    leaf_client_cert_subject string,  
    leaf_client_cert_validity string,  
    leaf_client_cert_serial_number string,  
    tls_verify_status string,  
    traceability_id string  
)  
    PARTITIONED BY  
    (  
        day STRING
```

```

)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1',
  'input.regex' =
  '^([\ ]*)([\ ]*)([0-9]*)([0-9]*)([A-Za-z0-9.-]*)([\ ]*)([-.0-9]*)
\"([\ ]*)([\ ]*)([\ ]*)([\ ]*)?([\ ]*)?(.*)?'
  LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/<ACCOUNT-NUMBER>/
elasticloadbalancing/<REGION>/'
TBLPROPERTIES
(
  "projection.enabled" = "true",
  "projection.day.type" = "date",
  "projection.day.range" = "2023/01/01,NOW",
  "projection.day.format" = "yyyy/MM/dd",
  "projection.day.interval" = "1",
  "projection.day.interval.unit" = "DAYS",
  "storage.location.template" = "s3://DOC-EXAMPLE-BUCKET/AWSLogs/<ACCOUNT-
NUMBER>/elasticloadbalancing/<REGION>/${day}"
)

```

Para obtener más información sobre la proyección de particiones, consulte [Proyección de particiones con Amazon Athena](#).

### Consultas de ejemplo para registros de conexión del ALB

El siguiente recuento de consultas se produce cuando el valor de `tls_verify_status` no era 'Success', agrupadas por dirección IP del cliente:

```

SELECT DISTINCT client_ip, count() AS count FROM alb_connection_logs
WHERE tls_verify_status != 'Success'
GROUP BY client_ip
ORDER BY count() DESC;

```

La siguiente consulta busca los incidentes en los que el valor de `tls_handshake_latency` fuera superior a 2 segundos en el intervalo de tiempo especificado:

```

SELECT * FROM alb_connection_logs
WHERE
(
  parse_datetime(time, 'yyyy-MM-dd'T'HH:mm:ss.SSSSSS'Z')
  BETWEEN

```

```
parse_datetime('2024-01-01-00:00:00', 'yyyy-MM-dd-HH:mm:ss')
AND
parse_datetime('2024-03-20-00:00:00', 'yyyy-MM-dd-HH:mm:ss')
)
AND
(tls_handshake_latency >= 2.0);
```

## Recursos adicionales de

- [¿Cómo analizo los registros de acceso al Equilibrador de carga de aplicación mediante Amazon Athena](#) en el Centro de conocimientos de AWS.
- Para obtener información sobre los códigos de estado HTTP de Elastic Load Balancing, consulte [Solución de problemas de equilibradores de carga de aplicación](#) en la Guía del usuario para equilibradores de carga de aplicación.
- [Catalogue y analice los registros del Equilibrador de carga de aplicación de forma más eficiente con clasificadores AWS Glue personalizados y Amazon Athena](#) en el Blog de macrodatos de AWS.

## Consulta de los registros del Equilibrador de carga clásico

Utilice los registros del Equilibrador de carga clásico para analizar y conocer los patrones del tráfico desde y hacia instancias de Elastic Load Balancing y aplicaciones de backend. Puede ver el origen del tráfico, su latencia y los bytes que se han transferido.

Antes de analizar los registros del Elastic Load Balancing, configúrelos para que se guarden en el bucket de Amazon S3 de destino. Para obtener más información, consulte [Habilitación de los registros de acceso del equilibrador de carga clásico](#).

- [Creación de la tabla de registros de Elastic Load Balancing](#)
- [Consultas de ejemplo de Elastic Load Balancing](#)

## Cómo crear la tabla de registros de Elastic Load Balancing

1. Copie y pegue la siguiente instrucción DDL en la consola de Athena. Verifique la [sintaxis](#) de los registros log de Elastic Load Balancing. Es posible que tenga que actualizar la siguiente consulta para incluir las columnas y la sintaxis Regex de la versión más reciente del registro.

```
CREATE EXTERNAL TABLE IF NOT EXISTS elb_logs (
```

```

timestamp string,
elb_name string,
request_ip string,
request_port int,
backend_ip string,
backend_port int,
request_processing_time double,
backend_processing_time double,
client_response_time double,
elb_response_code string,
backend_response_code string,
received_bytes bigint,
sent_bytes bigint,
request_verb string,
url string,
protocol string,
user_agent string,
ssl_cipher string,
ssl_protocol string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1',
  'input.regex' = '([^\ ]*) ([^\ ]*) ([^\ ]*):([0-9]*) ([^\ ]*)[:-]([0-9]*) ([-\.0-9]*)
([-\.0-9]*) ([-\.0-9]*) (|[-0-9]*) (-|[-0-9]*) ([-0-9]*) ([-0-9]*) \\\\"([^\ ]*)
([^\ ]*) (- |[\ ]*)\\\\" (\\"[^\"]*"") ([A-Z0-9-]+) ([A-Za-z0-9.-]*)$'
)
LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/AWS_account_ID/elasticloadbalancing/';

```

2. Modifique el bucket de Amazon S3 LOCATION para especificar el destino de los registros de Elastic Load Balancing.
3. Ejecute la consulta en la consola de Athena. Cuando la consulta finaliza, Athena registra la tabla `elb_logs` y los datos que contiene quedan disponibles para efectuar consultas. Para obtener más información, consulte [Consultas de ejemplo de Elastic Load Balancing](#).

## Consultas de ejemplo de Elastic Load Balancing

Utilice una consulta similar a la del siguiente ejemplo. En ella se muestran los servidores de aplicaciones backend que han devuelto un código de respuesta de error 4XX o 5XX. Utilice el operador LIMIT para limitar el número de registros que devuelve la consulta.

```
SELECT
```

```
timestamp,  
elb_name,  
backend_ip,  
backend_response_code  
FROM elb_logs  
WHERE backend_response_code LIKE '4%' OR  
       backend_response_code LIKE '5%'  
LIMIT 100;
```

Ejecute una consulta adicional para sumar el tiempo de respuesta de todas las transacciones agrupado por la dirección IP y el nombre de la instancia de Elastic Load Balancing del backend.

```
SELECT sum(backend_processing_time) AS  
total_ms,  
elb_name,  
backend_ip  
FROM elb_logs WHERE backend_ip <> ''  
GROUP BY backend_ip, elb_name  
LIMIT 100;
```

Para obtener más información, consulte [Análisis de datos en S3 con Athena](#).

## Consultas de registros de Amazon CloudFront

Puede configurar Amazon CloudFront CDN para que exporte los registros de acceso a distribuciones web a Amazon Simple Storage Service. Con estos registros podrá explorar los patrones de navegación de los usuarios en función de las propiedades web ofrecidas por CloudFront.

Para poder consultar los registros, debe haber habilitado previamente el registro de acceso a distribuciones web en su distribución de CloudFront preferida. Para obtener más información, consulte [Registros de acceso](#) en la Guía para desarrolladores de Amazon CloudFront. Anote el bucket de Amazon S3 en el que guarda estos registros.

- [Creación de una tabla para los registros estándar de CloudFront](#)
- [Creación de una tabla para registros en tiempo real de CloudFront](#)
- [Ejemplos de consultas para registros estándar de CloudFront](#)



## Creación de una tabla para los registros estándar de CloudFront

### Note

Este procedimiento funciona para los registros de acceso a distribuciones web en CloudFront. No es aplicable a los registros de transmisión de distribuciones RTMP.

### Cómo crear una tabla para los campos de los archivos de registro estándar de CloudFront

1. Copie y pegue la siguiente instrucción DDL de ejemplo en el Editor de consultas de la consola de Athena. La instrucción de ejemplo utiliza los campos del archivo de registro documentados en la sección [Campos de archivos de registro estándar](#) de Guía para desarrolladores de Amazon CloudFront. Modifique el valor de LOCATION para indicar el bucket de Amazon S3 donde se almacenan los registros. Para obtener información sobre cómo usar el Editor de consultas, consulte [Introducción](#).

Esta consulta especifica ROW FORMAT DELIMITED y FIELDS TERMINATED BY '\t' para indicar que los campos están delimitados por caracteres de tabulación. Para ROW FORMAT DELIMITED, Athena utiliza [LazySimpleSerDe](#) de forma predeterminada. La columna date se ha incluido en una secuencia de escape con acentos graves (`) porque es una palabra reservada en Athena. Para obtener más información, consulte [Palabras clave reservadas](#).

```
CREATE EXTERNAL TABLE IF NOT EXISTS cloudfront_standard_logs (  
  `date` DATE,  
  time STRING,  
  x_edge_location STRING,  
  sc_bytes BIGINT,  
  c_ip STRING,  
  cs_method STRING,  
  cs_host STRING,  
  cs_uri_stem STRING,  
  sc_status INT,  
  cs_referrer STRING,  
  cs_user_agent STRING,  
  cs_uri_query STRING,  
  cs_cookie STRING,  
  x_edge_result_type STRING,  
  x_edge_request_id STRING,  
  x_host_header STRING,  
  cs_protocol STRING,
```

```
cs_bytes BIGINT,  
time_taken FLOAT,  
x_forwarded_for STRING,  
ssl_protocol STRING,  
ssl_cipher STRING,  
x_edge_response_result_type STRING,  
cs_protocol_version STRING,  
fle_status STRING,  
fle_encrypted_fields INT,  
c_port INT,  
time_to_first_byte FLOAT,  
x_edge_detailed_result_type STRING,  
sc_content_type STRING,  
sc_content_len BIGINT,  
sc_range_start BIGINT,  
sc_range_end BIGINT  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t'  
LOCATION 's3://DOC-EXAMPLE-BUCKET/'  
TBLPROPERTIES ( 'skip.header.line.count'='2' )
```

2. Ejecute la consulta en la consola de Athena. Cuando la consulta finaliza, Athena registra la tabla `cloudfront_standard_logs`, dejando los datos que contiene listos para efectuar consultas.

## Creación de una tabla para registros en tiempo real de CloudFront

### Cómo crear una tabla para los campos de los archivos de registro en tiempo real de CloudFront

1. Copie y pegue la siguiente instrucción DDL de ejemplo en el Editor de consultas de la consola de Athena. La instrucción de ejemplo utiliza los campos del archivo de registro documentados en la sección [Registros en tiempo real](#) de la Guía para desarrolladores de Amazon CloudFront. Modifique el valor de `LOCATION` para indicar el bucket de Amazon S3 donde se almacenan los registros. Para obtener información sobre cómo usar el Editor de consultas, consulte [Introducción](#).

Esta consulta especifica `ROW FORMAT DELIMITED` y `FIELDS TERMINATED BY '\t'` para indicar que los campos están delimitados por caracteres de tabulación. Para `ROW FORMAT DELIMITED`, Athena utiliza [LazySimpleSerDe](#) de forma predeterminada. La columna `timestamp` se ha incluido en una secuencia de escape con acentos graves (```) porque es una palabra reservada en Athena. Para obtener más información, consulte [Palabras clave reservadas](#).

El siguiente ejemplo contiene todos los campos disponibles. Puede añadir comentarios en los campos o eliminar los que no necesite.

```
CREATE EXTERNAL TABLE IF NOT EXISTS cloudfront_real_time_logs (  
  `timestamp` STRING,  
  c_ip STRING,  
  time_to_first_byte BIGINT,  
  sc_status BIGINT,  
  sc_bytes BIGINT,  
  cs_method STRING,  
  cs_protocol STRING,  
  cs_host STRING,  
  cs_uri_stem STRING,  
  cs_bytes BIGINT,  
  x_edge_location STRING,  
  x_edge_request_id STRING,  
  x_host_header STRING,  
  time_taken BIGINT,  
  cs_protocol_version STRING,  
  c_ip_version STRING,  
  cs_user_agent STRING,  
  cs_referer STRING,  
  cs_cookie STRING,  
  cs_uri_query STRING,  
  x_edge_response_result_type STRING,  
  x_forwarded_for STRING,  
  ssl_protocol STRING,  
  ssl_cipher STRING,  
  x_edge_result_type STRING,  
  fle_encrypted_fields STRING,  
  fle_status STRING,  
  sc_content_type STRING,  
  sc_content_len BIGINT,  
  sc_range_start STRING,  
  sc_range_end STRING,  
  c_port BIGINT,  
  x_edge_detailed_result_type STRING,  
  c_country STRING,  
  cs_accept_encoding STRING,  
  cs_accept STRING,  
  cache_behavior_path_pattern STRING,  
  cs_headers STRING,
```

```
cs_header_names STRING,  
cs_headers_count BIGINT,  
primary_distribution_id STRING,  
primary_distribution_dns_name STRING,  
origin_fbl STRING,  
origin_lbl STRING,  
asn STRING  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t'  
LOCATION 's3://DOC-EXAMPLE-BUCKET/'  
TBLPROPERTIES ( 'skip.header.line.count'='2' )
```

2. Ejecute la consulta en la consola de Athena. Cuando la consulta finaliza, Athena registra la tabla `cloudfront_real_time_logs`, dejando los datos que contiene listos para efectuar consultas.

### Ejemplos de consultas para registros estándar de CloudFront

La siguiente consulta suma el número de bytes enviados por CloudFront entre el 9 y el 11 de junio de 2018. El nombre de la columna de fecha se indica entre comillas dobles, ya que se trata de una palabra reservada.

```
SELECT SUM(bytes) AS total_bytes  
FROM cloudfront_standard_logs  
WHERE "date" BETWEEN DATE '2018-06-09' AND DATE '2018-06-11'  
LIMIT 100;
```

Para eliminar filas duplicadas (por ejemplo, filas vacías duplicadas) de los resultados de la consulta, puede utilizar la instrucción `SELECT DISTINCT`, como en el ejemplo siguiente.

```
SELECT DISTINCT *  
FROM cloudfront_standard_logs  
LIMIT 10;
```

### Recursos adicionales de

Para obtener más información sobre cómo utilizar Athena para realizar consultas en los registros de CloudFront, consulte las siguientes publicaciones del [Blog de macrodatos de AWS](#).

[Consulte los registros de Servicio de AWS fácilmente con Amazon Athena](#) (29 de mayo de 2019).

[Analyze your Amazon CloudFront access logs at scale](#) (21 de diciembre de 2018).

[Cree una arquitectura sin servidor para analizar los registros de acceso de Amazon CloudFront con AWS Lambda, Amazon Athena, y Amazon Managed Service para Apache Flink](#) (26 de mayo de 2017).

## Consulta de registros de AWS CloudTrail

AWS CloudTrail es un servicio que registra las llamadas a la API de AWS y los eventos para las cuentas de Amazon Web Services.

Los registros de CloudTrail contienen detalles sobre las llamadas a la API a los Servicios de AWS, incluida la consola. CloudTrail genera archivos de registro cifrados y los almacena en Amazon S3. Para obtener más información, consulte la [Guía del usuario de AWS CloudTrail](#).

### Note

Si se desea realizar consultas SQL sobre la información de eventos de CloudTrail en diversas cuentas, regiones y fechas, se puede utilizar CloudTrail Lake. CloudTrail Lake es una alternativa de AWS a la creación de trazas que agrega información de una empresa en un único almacén de datos de eventos con capacidad de búsqueda. En lugar de utilizar el almacenamiento de bucket de Amazon S3, almacena los eventos en un lago de datos, lo que permite realizar consultas más ricas y rápidas. Se puede utilizar para crear consultas SQL que buscan eventos en organizaciones, regiones y dentro de intervalos de tiempo personalizados. Dado que realiza consultas de CloudTrail Lake dentro de la consola de CloudTrail, usarlo no requiere Athena. Para obtener más información, consulte la documentación de [CloudTrail Lake](#).

El uso de Athena con los registros de CloudTrail supone un modo eficaz de mejorar el análisis de la actividad de un Servicio de AWS. Por ejemplo, puede ejecutar consultas que identifiquen tendencias y aislar la actividad por atributos, como el usuario o la dirección IP de origen.

Una aplicación habitual es usar los registros de CloudTrail para analizar la actividad operativa a fin de garantizar la seguridad y la conformidad. Para obtener información sobre un ejemplo detallado, consulte la publicación del Blog de macrodatos de AWS, [Analyze security, compliance, and operational activity using AWS CloudTrail and Amazon Athena](#).

Athena le permite consultar estos archivos de registro directamente desde Amazon S3 especificando su ubicación con LOCATION. Puede hacerlo de una de las dos formas siguientes:

- Creando tablas para los archivos de registro de CloudTrail directamente desde la consola de CloudTrail.
- Creando tablas manualmente para los archivos de registro de CloudTrail en la consola de Athena.

## Temas

- [Descripción de los registros de CloudTrail y las tablas Athena](#)
- [Uso de la consola de CloudTrail para crear una tabla de Athena para registros de CloudTrail](#)
- [Creación de una tabla de registros de CloudTrail en Athena mediante la partición manual](#)
- [Creación de una tabla para la ruta de toda una organización mediante particiones manuales](#)
- [Creación de la tabla para los registros de CloudTrail en Athena mediante la proyección de particiones](#)
- [Consulta de campos anidados](#)
- [Consulta de ejemplo](#)
- [Sugerencias para consultar registros de CloudTrail](#)

## Descripción de los registros de CloudTrail y las tablas Athena

Antes de comenzar a crear tablas, debe comprender mejor el funcionamiento de CloudTrail y la forma en la que almacena los datos. Esto puede ayudarlo a crear las tablas que necesita, tanto si lo hace con la consola de CloudTrail como si usa Athena.

CloudTrail guarda los registros como archivos de texto JSON en formato comprimido gzip (\*.json.gzip). La ubicación de los archivos de registro depende del modo en que se configuran los registros de seguimiento, la Región de AWS (o regiones) donde se lleva a cabo el registro y otros factores.

Para obtener más información sobre el lugar donde se almacenan los logs, la estructura de JSON y el contenido de los archivos de registro, consulte los siguientes temas en la [Guía del usuario de AWS CloudTrail](#):

- [Búsqueda de archivos de registros de CloudTrail](#)
- [Ejemplos de archivos de registros de CloudTrail](#)
- [Contenido de los registros de CloudTrail](#)
- [Referencia de eventos de CloudTrail](#)

Para recopilar los registros y guardarlos en Amazon S3, habilite CloudTrail desde la AWS Management Console. Para obtener más información, consulte [Crear un registro de seguimiento](#) en la Guía del usuario de AWS CloudTrail.

Anote el bucket de Amazon S3 de destino en el que guarda los registros. Sustituya la cláusula LOCATION por la ruta a la ubicación del registro de CloudTrail y el conjunto de objetos con los que va a trabajar. En el ejemplo se usa como valor de LOCATION los registros de una cuenta determinada, pero puede especificar tanto como requiera su aplicación.

Por ejemplo:

- Para analizar datos de varias cuentas, puede modificar el especificador LOCATION para que indique todos los AWSLogs con el valor LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/'.
- Para analizar los datos de una fecha, cuenta y región específicas, utilice LOCATION 's3://DOC-EXAMPLE-BUCKET/123456789012/CloudTrail/us-east-1/2016/03/14/'.

El uso del máximo nivel en la jerarquía de objetos le ofrece la mayor flexibilidad al ejecutar consultas con Athena.

Uso de la consola de CloudTrail para crear una tabla de Athena para registros de CloudTrail

Puede crear una tabla de Athena no particionada para consultar registros de CloudTrail directamente desde la consola de CloudTrail. La creación de una tabla de Athena desde la consola de CloudTrail requiere que inicie sesión con un rol que tenga permisos suficientes para crear tablas en Athena.

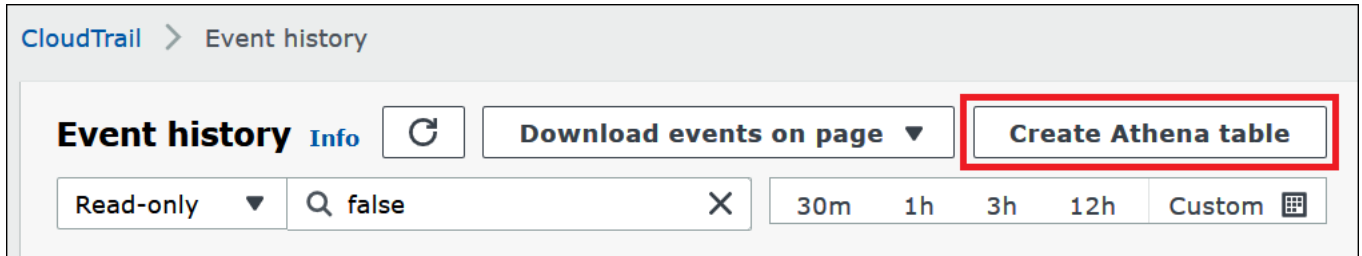
#### Note

No puede utilizar la consola de CloudTrail para crear una tabla de Athena para los registros de seguimiento de la organización. En su lugar, cree la tabla manualmente con la consola de Athena para que pueda especificar la ubicación de almacenamiento correcta. Para obtener información sobre los registros de seguimiento de una organización, consulte [Creación de registros de seguimiento de una organización](#) en la Guía del usuario de AWS CloudTrail.

- Para obtener información sobre cómo configurar permisos de Athena, consulte [Configuración](#).
- Para obtener información sobre la creación de una tabla con particiones, consulte [Creación de una tabla de registros de CloudTrail en Athena mediante la partición manual](#).

Para usar la consola de CloudTrail a fin de crear una tabla de Athena para registros de CloudTrail

1. Abra la consola de CloudTrail en <https://console.aws.amazon.com/cloudtrail/>.
2. En el panel de navegación, elija Historial de eventos.
3. Elija Creación de una tabla de Athena.



4. En Ubicación de almacenamiento, utilice la flecha hacia abajo para seleccionar el bucket de Amazon S3 en el que se almacenan los archivos de registro de seguimiento que desea consultar.

#### Note

Para buscar el nombre del bucket asociado a un registro de seguimiento, elija Registros de seguimiento en el panel de navegación de CloudTrail y vea la columna del bucket de S3 del registro de seguimiento. Para ver la ubicación del bucket en Amazon S3, elija el enlace del bucket en la columna del bucket de S3. Esto abre la consola de Amazon S3 en la ubicación del bucket de CloudTrail.

5. Elija Crear tabla. La tabla se crea con un nombre predeterminado que incluye el nombre del bucket de Amazon S3.

Creación de una tabla de registros de CloudTrail en Athena mediante la partición manual

Puede crear manualmente tablas para los archivos de registro de CloudTrail en la consola de Athena y, a continuación, ejecutar consultas en Athena.

Para crear una tabla de Athena para un registro de seguimiento de CloudTrail con la consola de Athena

1. Copie y pegue la siguiente instrucción DDL en el editor de consultas de la consola de Athena.

```
CREATE EXTERNAL TABLE cloudtrail_logs (  
  eventversion STRING,
```



```

useridentity STRUCT<
    type:STRING,
    principalid:STRING,
    arn:STRING,
    accountid:STRING,
    invokedby:STRING,
    accesskeyid:STRING,
    userName:STRING,
    sessioncontext:STRUCT<
        attributes:STRUCT<
            mfaauthenticated:STRING,
            creationdate:STRING>,
        sessionissuer:STRUCT<
            type:STRING,
            principalId:STRING,
            arn:STRING,
            accountId:STRING,
            userName:STRING>,
        ec2RoleDelivery:string,
        webIdFederationData: STRUCT<
            federatedProvider: STRING,
            attributes: map<string,string>
        >
    >
>,
eventtime STRING,
eventsource STRING,
eventname STRING,
awsregion STRING,
sourceipaddress STRING,
useragent STRING,
errorcode STRING,
errormessage STRING,
requestparameters STRING,
responseelements STRING,
additionaleventdata STRING,
requestid STRING,
eventid STRING,
resources ARRAY<STRUCT<
    arn:STRING,
    accountid:STRING,
    type:STRING>>,
eventtype STRING,
apiversion STRING,

```

```

readonly STRING,
recipientaccountid STRING,
serviceeventdetails STRING,
shareeventid STRING,
vpcendpointid STRING,
eventCategory STRING,
tlsDetails struct<
  tlsVersion:string,
  cipherSuite:string,
  clientProvidedHostHeader:string>
)
PARTITIONED BY (region string, year string, month string, day string)
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
STORED AS INPUTFORMAT 'com.amazon.emr.cloudtrail.CloudTrailInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/Account_ID/CloudTrail/';

```

### Note

Sugerimos usar `org.apache.hive.hcatalog.data.JsonSerDe` como se muestra en el ejemplo. Aunque existe `com.amazon.emr.hive.serde.CloudTrailSerde`, actualmente no gestiona algunos de los campos más recientes de CloudTrail.

2. (Opcional) Elimine los campos que no sean obligatorios para la tabla. Si solo necesita leer un determinado conjunto de columnas, la definición de la tabla puede excluir las demás columnas.
3. Modifique `s3://DOC-EXAMPLE-BUCKET/AWSLogs/Account_ID/CloudTrail/` de modo que apunte al bucket de Amazon S3 que contiene los datos de registro.
4. Compruebe que los campos se muestran correctamente. Para obtener más información sobre la lista completa de campos de un registro de CloudTrail, consulte [Contenido de los registros de CloudTrail](#).

El siguiente ejemplo utiliza [El SerDe JSON de Hive](#). En este ejemplo, los campos `requestparameters`, `responseelements` y `additional eventdata` se incluyen como tipo `STRING` en la consulta, pero son tipo de datos `STRUCT` utilizado en JSON. Por lo tanto, para obtener datos de estos campos puede usar funciones `JSON_EXTRACT`. Para obtener más información, consulte [the section called “Extracción de datos de JSON”](#). Para mejorar el rendimiento, este ejemplo particiona los datos por Región de AWS, año, mes y día.

5. Ejecutar la instrucción `CREATE TABLE` en la consola de Athena.

6. Utilice el comando [ALTER TABLE ADD PARTITION](#) para cargar las particiones de modo que pueda consultarlas, como en el ejemplo siguiente.

```
ALTER TABLE table_name ADD
  PARTITION (region='us-east-1',
            year='2019',
            month='02',
            day='01')
  LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/Account_ID/CloudTrail/us-
east-1/2019/02/01/'
```

Creación de una tabla para la ruta de toda una organización mediante particiones manuales

Para crear una tabla para los archivos de registro de CloudTrail de toda la organización en Athena, siga los pasos de [Creación de una tabla de registros de CloudTrail en Athena mediante la partición manual](#), pero realice las modificaciones que se indican en el siguiente procedimiento.

Para crear una tabla de Athena para los registros de CloudTrail de toda una organización

1. En la instrucción CREATE TABLE, modifique la cláusula LOCATION para incluir el ID de organización, como en el siguiente ejemplo:

```
LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/organization_id/Account_ID/CloudTrail/'
```

2. En la cláusula PARTITIONED BY, agregue una entrada para el ID de cuenta en forma de cadena, como en el siguiente ejemplo:

```
PARTITIONED BY (account string, region string, year string, month string, day
string)
```

En el siguiente ejemplo, se muestra el resultado combinado:

```
...

PARTITIONED BY (account string, region string, year string, month string, day
string)
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
STORED AS INPUTFORMAT 'com.amazon.emr.cloudtrail.CloudTrailInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
```

```
LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/organization_id/Account_ID/CloudTrail/'
```

3. En la cláusula ADD PARTITION de la instrucción ALTER TABLE, incluya el ID de cuenta, como en el siguiente ejemplo:

```
ALTER TABLE table_name ADD  
PARTITION (account='111122223333',  
region='us-east-1',  
year='2022',  
month='08',  
day='08')
```

4. En la cláusula LOCATION de la instrucción ALTER TABLE, incluya el ID de la organización, el ID de cuenta y la partición que desea agregar, como en el siguiente ejemplo:

```
LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/organization_id/Account_ID/CloudTrail/us-  
east-1/2022/08/08/'
```

En el siguiente ejemplo, la instrucción ALTER TABLE muestra el resultado combinado:

```
ALTER TABLE table_name ADD  
PARTITION (account='111122223333',  
region='us-east-1',  
year='2022',  
month='08',  
day='08')  
LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/organization_id/111122223333/CloudTrail/  
us-east-1/2022/08/08/'
```

## Creación de la tabla para los registros de CloudTrail en Athena mediante la proyección de particiones

Dado que los registros de CloudTrail tienen una estructura conocida cuyo esquema de partición puede especificar de antemano, puede reducir el tiempo de ejecución de las consultas y automatizar la administración de particiones mediante la característica de proyección de particiones de Athena. La proyección de particiones agrega de forma automática nuevas particiones a medida que se agregan nuevos datos. Esto hace que no sea necesario agregar particiones manualmente mediante ALTER TABLE ADD PARTITION.

En el siguiente ejemplo, la instrucción CREATE TABLE utiliza automáticamente la proyección de particiones en los registros de CloudTrail desde una fecha especificada hasta el presente

para una sola Región de AWS. En las cláusulas `LOCATION` y `storage.location.template`, reemplace los marcadores de posición *bucket*, *account-id* y *aws-region* por valores idénticos en consecuencia. En `projection.timestamp.range`, reemplace *2020/01/01* por la fecha de inicio que quiera utilizar. Una vez ejecutada la consulta correctamente, puede consultar la tabla. No tiene que ejecutar `ALTER TABLE ADD PARTITION` para cargar las particiones.

```
CREATE EXTERNAL TABLE cloudtrail_logs_pp(  
  eventVersion STRING,  
  userIdentity STRUCT<  
    type: STRING,  
    principalId: STRING,  
    arn: STRING,  
    accountId: STRING,  
    invokedBy: STRING,  
    accessKeyId: STRING,  
    userName: STRING,  
    sessionContext: STRUCT<  
      attributes: STRUCT<  
        mfaAuthenticated: STRING,  
        creationDate: STRING>,  
      sessionIssuer: STRUCT<  
        type: STRING,  
        principalId: STRING,  
        arn: STRING,  
        accountId: STRING,  
        userName: STRING>,  
      ec2RoleDelivery:string,  
      webIdFederationData: STRUCT<  
        federatedProvider: STRING,  
        attributes: map<string,string>  
      >  
    >  
  >,  
  eventTime STRING,  
  eventSource STRING,  
  eventName STRING,  
  awsRegion STRING,  
  sourceIpAddress STRING,  
  userAgent STRING,  
  errorCode STRING,  
  errorMessage STRING,  
  requestparameters STRING,  
  responseelements STRING,
```

```

additionalEventData STRING,
requestId STRING,
eventId STRING,
readOnly STRING,
resources ARRAY<STRUCT<
    arn: STRING,
    accountId: STRING,
    type: STRING>>,
eventType STRING,
apiVersion STRING,
recipientAccountId STRING,
serviceEventDetails STRING,
sharedEventID STRING,
vpcEndpointId STRING,
eventCategory STRING,
tlsDetails struct<
    tlsVersion:string,
    cipherSuite:string,
    clientProvidedHostHeader:string>
)
PARTITIONED BY (
    `timestamp` string)
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'
STORED AS INPUTFORMAT 'com.amazon.emr.cloudtrail.CloudTrailInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
    's3://DOC-EXAMPLE-BUCKET/AWSLogs/account-id/CloudTrail/aws-region'
TBLPROPERTIES (
    'projection.enabled'='true',
    'projection.timestamp.format'='yyyy/MM/dd',
    'projection.timestamp.interval'='1',
    'projection.timestamp.interval.unit'='DAYS',
    'projection.timestamp.range'='2020/01/01,NOW',
    'projection.timestamp.type'='date',
    'storage.location.template'='s3://DOC-EXAMPLE-BUCKET/AWSLogs/account-id/
CloudTrail/aws-region/${timestamp}')

```

Para obtener más información sobre la proyección de particiones, consulte [Proyección de particiones con Amazon Athena](#).

## Consulta de campos anidados

Ya que los campos `userIdentity` y `resources` son tipos de datos anidados, su consulta requiere un tratamiento especial.

El objeto `userIdentity` se compone de tipos `STRUCT` anidados. Estos se pueden consultar utilizando un punto para separar los campos, como en el ejemplo a continuación:

```
SELECT
  eventsource,
  eventname,
  useridentity.sessioncontext.attributes.creationdate,
  useridentity.sessioncontext.sessionissuer.arn
FROM cloudtrail_logs
WHERE useridentity.sessioncontext.sessionissuer.arn IS NOT NULL
ORDER BY eventsource, eventname
LIMIT 10
```

El campo `resources` es una matriz de objetos `STRUCT`. Para estas matrices, utilice `CROSS JOIN UNNEST` para desanidar la matriz, de modo que pueda consultar sus objetos.

En el ejemplo siguiente se devuelven todas las filas donde el ARN de recurso termina en `example/datafile.txt`. Por motivos de legibilidad, la función [replace](#) elimina la subcadena `arn:aws:s3:::` inicial del ARN.

```
SELECT
  awsregion,
  replace(unnested.resources_entry.ARN, 'arn:aws:s3:::') as s3_resource,
  eventname,
  eventtime,
  useragent
FROM cloudtrail_logs t
CROSS JOIN UNNEST(t.resources) unnested (resources_entry)
WHERE unnested.resources_entry.ARN LIKE '%example/datafile.txt'
ORDER BY eventtime
```

Las siguientes son consultas de ejemplo para eventos `DeleteBucket`. La consulta extrae el nombre del bucket y el ID de cuenta al que pertenece el bucket del objeto `resources`.

```
SELECT
  awsregion,
```

```
replace(unnested.resources_entry.ARN,'arn:aws:s3:::') as deleted_bucket,
eventtime AS time_deleted,
useridentity.username,
unnested.resources_entry.accountid as bucket_acct_id
FROM cloudtrail_logs t
CROSS JOIN UNNEST(t.resources) unnested (resources_entry)
WHERE eventname = 'DeleteBucket'
ORDER BY eventtime
```

Para obtener más información acerca de la desanidación, consulte [Filtro de matrices](#).

## Consulta de ejemplo

En el siguiente ejemplo, se muestra una parte de una consulta que devuelve todas las solicitudes anónimas (sin firmar) de la tabla creada para registros de eventos de CloudTrail. Esta consulta selecciona esas solicitudes donde `useridentity.accountid` es anónimo y no se ha especificado `useridentity.arn`:

```
SELECT *
FROM cloudtrail_logs
WHERE
    eventsource = 's3.amazonaws.com' AND
    eventname in ('GetObject') AND
    useridentity.accountid = 'anonymous' AND
    useridentity.arn IS NULL AND
    requestparameters LIKE '%[your bucket name ]%';
```

Para obtener más información, consulte la publicación del Blog de macrodatos de AWS [Análisis de la seguridad, la conformidad y la actividad operativa por medio de AWS CloudTrail y Amazon Athena](#).

## Sugerencias para consultar registros de CloudTrail

Para explorar los datos de los registros de CloudTrail siga estos consejos:

- Antes de consultar los registros, compruebe que la tabla de registros sea similar a la de [the section called “Creación de una tabla de registros de CloudTrail en Athena mediante la partición manual”](#). Si no es la primera tabla, elimine la tabla existente con el siguiente comando: `DROP TABLE cloudtrail_logs`.
- Después de eliminar la tabla existente, vuelva a crearla. Para obtener más información, consulte [Creación de una tabla de registros de CloudTrail en Athena mediante la partición manual](#).



Verifique que los campos de la consulta de Athena se muestren correctamente. Para obtener información sobre la lista completa de campos de un registro de CloudTrail, consulte [Contenido de los registros de CloudTrail](#).

Si la consulta incluye campos en formato JSON, como STRUCT, extraiga los datos de JSON. Para obtener más información, consulte [Extracción de datos de JSON](#).

Algunas sugerencias para realizar consultas en la tabla de CloudTrail:

- Para comenzar, observe qué usuarios de han invocado qué operaciones de la API y desde qué direcciones IP lo han hecho.
- Utilice la siguiente consulta SQL básica como plantilla. Pegue la consulta en la consola de Athena y ejecútela.

```
SELECT
  useridentity.arn,
  eventname,
  sourceipaddress,
  eventtime
FROM cloudtrail_logs
LIMIT 100;
```

- Modifique la consulta para explorar en profundidad los datos.
- Para mejorar el rendimiento, incluya la cláusula LIMIT para obtener un subconjunto de filas determinado.

## Consultas de los registros de Amazon EMR

Amazon EMR y las aplicaciones de macrodatos que se ejecutan en Amazon EMR producen archivos de registro. Los archivos de registro se escriben en el nodo maestro y también se puede configurar Amazon EMR para archivar los archivos de registro en Amazon S3 automáticamente. Puede utilizar Amazon Athena para consultar estos registros con el fin de identificar eventos y tendencias de las aplicaciones y los clústeres. Para obtener más información sobre los tipos de archivos de registro en Amazon EMR y guardarlos en Amazon S3, consulte [Ver archivos de registro](#) en la Guía de administración de Amazon EMR.

## Creación y consulta de una tabla básica basada en archivos de registro de Amazon EMR

El siguiente ejemplo crea una tabla básica, `myemrlogs`, basada en archivos de registro guardados en `s3://aws-logs-123456789012-us-west-2/elasticmapreduce/j-2ABCDE34F5GH6/elasticmapreduce/`. La ubicación de Amazon S3 utilizada en los siguientes ejemplos refleja el patrón de la ubicación de registro predeterminada para un clúster de EMR creado por la cuenta `123456789012` de Amazon Web Services en la región `us-west-2`. Si utiliza una ubicación personalizada, el patrón es `s3://DOC-EXAMPLE-BUCKET/ClusterID`.

Para obtener información sobre cómo crear una tabla con particiones para mejorar potencialmente el rendimiento de las consultas y reducir la transferencia de datos, consulte [Creación y consulta de una tabla particionada basada en los registros de Amazon EMR](#).

```
CREATE EXTERNAL TABLE `myemrlogs` (
  `data` string COMMENT 'from deserializer')
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
LINES TERMINATED BY '\n'
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://aws-logs-123456789012-us-west-2/elasticmapreduce/j-2ABCDE34F5GH6'
```

Las siguientes consultas de ejemplo se pueden ejecutar en la tabla `myemrlogs` creada por el ejemplo anterior.

Example – Consultar registros de pasos para incidentes de ERROR, WARN, INFO, EXCEPTION, FATAL o DEBUG

```
SELECT data,
       "$PATH"
FROM "default"."myemrlogs"
WHERE regexp_like("$PATH", 's-86URH188Z6B1')
       AND regexp_like(data, 'ERROR|WARN|INFO|EXCEPTION|FATAL|DEBUG') limit 100;
```

Example – Consultar un registro de instancia específico, `i-00b3c0a839ece0a9c`, para ERROR, WARN, INFO, EXCEPTION, FATAL o DEBUG

```
SELECT "data",
```

```

"$PATH" AS filepath
FROM "default"."myemrlogs"
WHERE regexp_like("$PATH", 'i-00b3c0a839ece0a9c')
      AND regexp_like("$PATH", 'state')
      AND regexp_like(data, 'ERROR|WARN|INFO|EXCEPTION|FATAL|DEBUG') limit 100;

```

Example – Consultar los registros de aplicación Presto para ERROR, WARN, INFO, EXCEPTION, FATAL o DEBUG

```

SELECT "data",
      "$PATH" AS filepath
FROM "default"."myemrlogs"
WHERE regexp_like("$PATH", 'presto')
      AND regexp_like(data, 'ERROR|WARN|INFO|EXCEPTION|FATAL|DEBUG') limit 100;

```

Example – Consultar los registros de aplicación Namenode para ERROR, WARN, INFO, EXCEPTION, FATAL o DEBUG

```

SELECT "data",
      "$PATH" AS filepath
FROM "default"."myemrlogs"
WHERE regexp_like("$PATH", 'namenode')
      AND regexp_like(data, 'ERROR|WARN|INFO|EXCEPTION|FATAL|DEBUG') limit 100;

```

Example – Consultar todos los registros por fecha y hora para ERROR, WARN, INFO, EXCEPTION, FATAL o DEBUG

```

SELECT distinct("$PATH") AS filepath
FROM "default"."myemrlogs"
WHERE regexp_like("$PATH", '2019-07-23-10')
      AND regexp_like(data, 'ERROR|WARN|INFO|EXCEPTION|FATAL|DEBUG') limit 100;

```

Creación y consulta de una tabla particionada basada en los registros de Amazon EMR

Estos ejemplos utilizan la misma ubicación de registro para crear una tabla de Athena, pero la tabla está particionada y, a continuación, se crea una partición para cada ubicación de registro. Para obtener más información, consulte [Particiones de datos en Athena](#).

La siguiente consulta crea la tabla particionada con el nombre mypartitionedemrlogs:

```
CREATE EXTERNAL TABLE `mypartitionedemrlogs`(  
  `data` string COMMENT 'from deserializer')  
  partitioned by (logtype string)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '|'   
LINES TERMINATED BY '\n'  
STORED AS INPUTFORMAT  
  'org.apache.hadoop.mapred.TextInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat'  
LOCATION 's3://aws-logs-123456789012-us-west-2/elasticmapreduce/j-2ABCDE34F5GH6'
```

Las siguientes instrucciones de consulta crean particiones de tabla basadas en subdirectorios para diferentes tipos de registro que Amazon EMR crea en Amazon S3:

```
ALTER TABLE mypartitionedemrlogs ADD  
  PARTITION (logtype='containers')  
  LOCATION 's3://aws-logs-123456789012-us-west-2/elasticmapreduce/j-2ABCDE34F5GH6/  
containers/'
```

```
ALTER TABLE mypartitionedemrlogs ADD  
  PARTITION (logtype='hadoop-mapreduce')  
  LOCATION 's3://aws-logs-123456789012-us-west-2/elasticmapreduce/j-2ABCDE34F5GH6/  
hadoop-mapreduce/'
```

```
ALTER TABLE mypartitionedemrlogs ADD  
  PARTITION (logtype='hadoop-state-pusher')  
  LOCATION 's3://aws-logs-123456789012-us-west-2/elasticmapreduce/j-2ABCDE34F5GH6/  
hadoop-state-pusher/'
```

```
ALTER TABLE mypartitionedemrlogs ADD  
  PARTITION (logtype='node')  
  LOCATION 's3://aws-logs-123456789012-us-west-2/elasticmapreduce/j-2ABCDE34F5GH6/  
node/'
```

```
ALTER TABLE mypartitionedemrlogs ADD  
  PARTITION (logtype='steps')  
  LOCATION 's3://aws-logs-123456789012-us-west-2/elasticmapreduce/j-2ABCDE34F5GH6/  
steps/'
```

Después de crear las particiones, puede ejecutar una consulta `SHOW PARTITIONS` en la tabla para confirmar:

```
SHOW PARTITIONS mypartitionedemrlogs;
```

Los siguientes ejemplos demuestran que las consultas de entradas de registro específicas utilizan la tabla y las particiones creadas por los ejemplos anteriores.

Example – Consulta de los registros de aplicación de `application_1561661818238_0002` en la partición de contenedores para `ERROR` o `WARN`

```
SELECT data,  
       "$PATH"  
FROM "default"."mypartitionedemrlogs"  
WHERE logtype='containers'  
       AND regexp_like("$PATH",'application_1561661818238_0002')  
       AND regexp_like(data, 'ERROR|WARN') limit 100;
```

Example – Consulta de la partición Hadoop-Mapreduce para el trabajo `job_1561661818238_0004` y las reducciones con error

```
SELECT data,  
       "$PATH"  
FROM "default"."mypartitionedemrlogs"  
WHERE logtype='hadoop-mapreduce'  
       AND regexp_like(data,'job_1561661818238_0004|Failed Reduces') limit 100;
```

Example – Consulta de los registros de Hive en la partición de nodo para el ID de consulta `056e0609-33e1-4611-956c-7a31b42d2663`

```
SELECT data,  
       "$PATH"  
FROM "default"."mypartitionedemrlogs"  
WHERE logtype='node'  
       AND regexp_like("$PATH",'hive')  
       AND regexp_like(data,'056e0609-33e1-4611-956c-7a31b42d2663') limit 100;
```

Example – Consulta de los registros Resource manager en la partición de nodo para la aplicación `1567660019320_0001_01_000001`

```
SELECT data,
```

```
"$PATH"  
FROM "default"."mypartitionedemrlogs"  
WHERE logtype='node'  
      AND regexp_like(data,'resourcemanager')  
      AND regexp_like(data,'1567660019320_0001_01_000001') limit 100
```

## Consulta de los registros de flujo de AWS Global Accelerator

Puede utilizar AWS Global Accelerator para crear aceleradores que dirijan el tráfico de red a los puntos de conexión óptimos a través de la red global de AWS. Para obtener más información sobre Global Accelerator, consulte [Qué es AWS Global Accelerator](#).

Los registros de flujo de Global Accelerator le permiten capturar información acerca del tráfico de direcciones IP entrante y saliente de las interfaces de red de los aceleradores. Los datos de registro de flujo se publican en Amazon S3, donde puede recuperar y ver los datos. Para obtener más información, consulte [Registros de flujo de AWS Global Accelerator](#).

Puede utilizar Athena para consultar los registros de flujo de Global Accelerator mediante la creación de una tabla que especifique su ubicación en Amazon S3.

Para crear la tabla de los registros de flujo de Global Accelerator

1. Copie y pegue la siguiente instrucción DDL en la consola de Athena. Esta consulta especifica ROW FORMAT DELIMITED (formato de columna delimitado) y omite la especificación de [SerDe](#), lo que significa que la consulta utiliza [LazySimpleSerDe](#). En esta consulta los campos terminan con un espacio.

```
CREATE EXTERNAL TABLE IF NOT EXISTS aga_flow_logs (  
  version string,  
  account string,  
  acceleratorid string,  
  clientip string,  
  clientport int,  
  gip string,  
  gipport int,  
  endpointip string,  
  endpointport int,  
  protocol string,  
  ipaddresstype string,  
  numpackets bigint,  
  numbytes int,  
  starttime int,
```

```

endtime int,
action string,
logstatus string,
agasourceip string,
agasourceport int,
endpointregion string,
agaregion string,
direction string
)
PARTITIONED BY (dt string)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ' '
LOCATION 's3://DOC-EXAMPLE-BUCKET/prefix/AWSLogs/account_id/globalaccelerator/
region/'
TBLPROPERTIES ("skip.header.line.count"="1");

```

2. Modifique el valor LOCATION para que apunte al bucket de Amazon S3 que contiene los datos de registro.

```
's3://DOC-EXAMPLE-BUCKET/prefix/AWSLogs/account_id/globalaccelerator/region_code/'
```

3. Ejecute la consulta en la consola de Athena. Cuando la consulta finaliza, Athena registra la tabla `aga_flow_logs` y los datos que contiene quedan disponibles para efectuar consultas.
4. Cree particiones para leer los datos, tal y como se indica en la siguiente consulta de ejemplo. La consulta crea una sola partición para la fecha especificada. Sustituya los marcadores de posición de fecha y ubicación.

```

ALTER TABLE aga_flow_logs
ADD PARTITION (dt='YYYY-MM-dd')
LOCATION 's3://DOC-EXAMPLE-BUCKET/prefix/AWSLogs/account_id/
globalaccelerator/region_code/YYYY/MM/dd';

```

## Ejemplos de consulta de registros de flujo de AWS Global Accelerator

Example – Enumera las solicitudes que pasan a través de una ubicación de borde específica

En la siguiente consulta de ejemplo se enumeran las solicitudes que pasaron a través de la ubicación periférica LHR. Utilice el operador LIMIT para limitar el número de registros que devuelve la consulta a la vez.

```
SELECT
```

```
clientip,  
agaregion,  
protocol,  
action  
FROM  
  aga_flow_logs  
WHERE  
  agaregion LIKE 'LHR%'  
LIMIT  
  100;
```

**Example – Enumera las direcciones IP del punto de conexión que reciben la mayor cantidad de solicitudes HTTPS**

Para ver qué direcciones IP de punto de conexión reciben el mayor número de solicitudes HTTPS, utilice la siguiente consulta. Esta consulta cuenta el número de paquetes recibidos en el puerto HTTPS 443, los agrupa según la dirección IP de destino, y devuelve las 10 direcciones IP principales.

```
SELECT  
  SUM(numpackets) AS packetcount,  
  endpointip  
FROM  
  aga_flow_logs  
WHERE  
  endpointport = 443  
GROUP BY  
  endpointip  
ORDER BY  
  packetcount DESC  
LIMIT  
  10;
```

## Consulta de los resultados de Amazon GuardDuty

[Amazon GuardDuty](#) es un servicio de monitoreo de seguridad que ayuda a identificar actividades inesperadas y potencialmente no autorizadas o malintencionadas en su entorno de AWS. Cuando detecta alguna actividad inesperada y potencialmente malintencionada, GuardDuty genera [resultados](#) de seguridad que puede exportar a Amazon S3 para su almacenamiento y análisis. Después de exportar los resultados a Amazon S3, puede utilizar Athena para consultarlos. En



este artículo se muestra cómo crear una tabla en Athena para obtener resultados en GuardDuty y consultarlos.

Para obtener más información sobre Amazon GuardDuty, consulte la [Guía del usuario de Amazon GuardDuty](#).

### Requisitos previos

- Habilite la característica GuardDuty para exportar los resultados a Amazon S3. Para ver los pasos, consulte [Exportación de resultados](#) en la Guía del usuario de Amazon GuardDuty.

### Creación de una tabla en Athena para obtener resultados en GuardDuty

Para consultar los resultados de GuardDuty desde Athena, debe crear una tabla para ellos.

Para crear una tabla en Athena para los resultados de GuardDuty

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Pegue la siguiente instrucción DDL en la consola de Athena. Modifique los valores de LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/*account-id*/GuardDuty/' para que apunten a los resultados de GuardDuty en Amazon S3.

```
CREATE EXTERNAL TABLE `gd_logs` (  
  `schemaversion` string,  
  `accountid` string,  
  `region` string,  
  `partition` string,  
  `id` string,  
  `arn` string,  
  `type` string,  
  `resource` string,  
  `service` string,  
  `severity` string,  
  `createdat` string,  
  `updatedat` string,  
  `title` string,  
  `description` string)  
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'  
LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/account-id/GuardDuty/'  
TBLPROPERTIES ('has_encrypted_data'='true')
```

**Note**

El SerDe espera que cada documento JSON esté en una sola línea de texto sin caracteres de terminación de línea que separen los campos del registro. Si el texto JSON está en formato de impresión, puede recibir un mensaje de error como `HIVE_CURSOR_ERROR: la fila no es un objeto JSON válido` o `HIVE_CURSOR_ERROR: JsonParseException: fin de entrada inesperado: marcador de cierre esperado para OBJECT cuando intenta consultar la tabla después de crearla`. Para obtener más información, consulte los [Archivos de datos JSON](#) en la documentación de OpenX SerDE en GitHub.

3. Ejecute la consulta en la consola de Athena para registrar la tabla `gd_logs`. Cuando se complete la consulta, los resultados estarán listos para su consulta desde Athena.

### Consultas de ejemplo

En los siguientes ejemplos se muestra cómo consultar resultados de GuardDuty desde Athena.

#### Example – Exfiltración de datos DNS

La siguiente consulta devuelve información acerca de las instancias de Amazon EC2 que podrían estar filtrando datos a través de consultas de DNS.

```
SELECT
  title,
  severity,
  type,
  id AS FindingID,
  accountid,
  region,
  createdat,
  updatedat,
  json_extract_scalar(service, '$.count') AS Count,
  json_extract_scalar(resource, '$.instancedetails.instanceid') AS InstanceID,
  json_extract_scalar(service, '$.action.actiontype') AS DNS_ActionType,
  json_extract_scalar(service, '$.action.dnsrequestaction.domain') AS DomainName,
  json_extract_scalar(service, '$.action.dnsrequestaction.protocol') AS protocol,
  json_extract_scalar(service, '$.action.dnsrequestaction.blocked') AS blocked
FROM gd_logs
WHERE type = 'Trojan:EC2/DNSDataExfiltration'
```

```
ORDER BY severity DESC
```

## Example – Acceso no autorizado de un usuario de IAM

La siguiente consulta devuelve todos los tipos de resultados `UnauthorizedAccess:IAMUser` de una entidad principal de IAM en todas las regiones.

```
SELECT title,
       severity,
       type,
       id,
       accountid,
       region,
       createdat,
       updatedat,
       json_extract_scalar(service, '$.count') AS Count,
       json_extract_scalar(resource, '$.accesskeydetails.username') AS IAMPrincipal,
       json_extract_scalar(service, '$.action.awsapicallaction.api') AS
APIActionCalled
FROM gd_logs
WHERE type LIKE '%UnauthorizedAccess:IAMUser%'
ORDER BY severity desc;
```

## Consejos para consultar los resultados de GuardDuty

Cuando cree una consulta, tenga en cuenta lo siguiente.

- Para extraer datos de campos JSON anidados, utilice las funciones `json_extract` o `json_extract_scalar` de Presto. Para obtener más información, consulte [Extracción de datos de JSON](#).
- Asegúrese de que todos los caracteres de los campos JSON estén en minúsculas.
- Para obtener información sobre la descarga de resultados de consulta, consulte [Descarga de archivos de resultados de consultas mediante la consola de Athena](#).

## Consulta de registros de AWS Network Firewall

AWS Network Firewall es un servicio administrado que puede utilizar para implementar protecciones de red esenciales para las instancias de Amazon Virtual Private Cloud. AWS Network Firewall funciona junto con AWS Firewall Manager para que pueda crear políticas basadas en reglas de AWS Network Firewall y, a continuación, aplicar esas directivas de forma centralizada en las VPC y las

cuentas. Para obtener más información acerca de AWS Network Firewall, consulte [AWS Network Firewall](#).

Puede configurar el registro de AWS Network Firewall para el tráfico que reenvía al motor de reglas con estado de su firewall. El registro proporciona información detallada sobre el tráfico de red, incluida la hora en la que el motor con estado recibió un paquete, información detallada sobre el paquete y cualquier acción de regla de estado realizada con el paquete. Los registros se publican en el destino de registro que ha configurado, donde puede recuperarlos y verlos. Para obtener más información, consulte [Registro del tráfico de red desde AWS Network Firewall](#) en la Guía para desarrolladores de AWS Network Firewall.

### Creación de una tabla para los registros de alertas

1. Modifique el siguiente ejemplo de instrucción DDL para adaptarlo a la estructura de su registro de alertas. Es posible que tenga que actualizar la instrucción para incluir las columnas de la versión más reciente de los registros. Para obtener más información, consulte [Contenido de un registro de firewall](#) en la Guía para desarrolladores de AWS Network Firewall.

```
CREATE EXTERNAL TABLE network_firewall_alert_logs (  
  firewall_name string,  
  availability_zone string,  
  event_timestamp string,  
  event struct<  
    timestamp:string,  
    flow_id:bigint,  
    event_type:string,  
    src_ip:string,  
    src_port:int,  
    dest_ip:string,  
    dest_port:int,  
    proto:string,  
    app_proto:string,  
    tls_inspected:boolean,  
    alert:struct<  
      alert_id:string,  
      alert_type:string,  
      action:string,  
      signature_id:int,  
      rev:int,  
      signature:string,  
      category:string,  
      severity:int,
```

```

    rule_name:string,
    alert_name:string,
    alert_severity:string,
    alert_description:string,
    file_name:string,
    file_hash:string,
    packet_capture:string,
    reference_links:array<string>
  >,
  src_country:string,
  dest_country:string,
  src_hostname:string,
  dest_hostname:string,
  user_agent:string,
  url:string
)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION 's3://DOC-EXAMPLE-BUCKET/path_to_alert_logs_folder/';

```

2. Modifique la cláusula LOCATION para especificar la carpeta de sus registros en Amazon S3.
3. Ejecute la consulta CREATE TABLE en el editor de consultas de Athena. Cuando la consulta finaliza, Athena registra la tabla network\_firewall\_alert\_logs, lo que hace que los datos a los que apunta estén listos para las consultas.

### Consulta de ejemplo de registro de alerta

El ejemplo de consulta del registro de alertas de esta sección filtra los eventos en los que se realizó una inspección de TLS que tienen alertas con un nivel de gravedad igual o superior a 2.

La consulta utiliza alias para crear encabezados de columna de salida que muestran a qué struct pertenece la columna. Por ejemplo, el encabezado de la columna del campo event.alert.category es event\_alert\_category en lugar de simplemente category. Para personalizar aún más los nombres de las columnas, puede modificar los alias según sus preferencias. Por ejemplo, puede usar guiones bajos u otros separadores para delimitar los nombres de struct y los nombres de los campos.

Recuerde modificar los nombres y las referencias a struct de las columnas en función de la definición de la tabla y de los campos que desee incluir en el resultado de la consulta.

```
SELECT
```

```
firewall_name,  
availability_zone,  
event_timestamp,  
event.timestamp AS event_timestamp,  
event.flow_id AS event_flow_id,  
event.event_type AS event_type,  
event.src_ip AS event_src_ip,  
event.src_port AS event_src_port,  
event.dest_ip AS event_dest_ip,  
event.dest_port AS event_dest_port,  
event.proto AS event_protol,  
event.app_proto AS event_app_proto,  
event.tls_inspected AS event_tls_inspected,  
event.alert.alert_id AS event_alert_alert_id,  
event.alert.alert_type AS event_alert_alert_type,  
event.alert.action AS event_alert_action,  
event.alert.signature_id AS event_alert_signature_id,  
event.alert.rev AS event_alert_rev,  
event.alert.signature AS event_alert_signature,  
event.alert.category AS event_alert_category,  
event.alert.severity AS event_alert_severity,  
event.alert.rule_name AS event_alert_rule_name,  
event.alert.alert_name AS event_alert_alert_name,  
event.alert.alert_severity AS event_alert_alert_severity,  
event.alert.alert_description AS event_alert_alert_description,  
event.alert.file_name AS event_alert_file_name,  
event.alert.file_hash AS event_alert_file_hash,  
event.alert.packet_capture AS event_alert_packet_capture,  
event.alert.reference_links AS event_alert_reference_links,  
event.src_country AS event_src_country,  
event.dest_country AS event_dest_country,  
event.src_hostname AS event_src_hostname,  
event.dest_hostname AS event_dest_hostname,  
event.user_agent AS event_user_agent,  
event.url AS event_url  
FROM  
  network_firewall_alert_logs  
WHERE  
  event.alert.severity >= 2  
  AND event.tls_inspected = true  
LIMIT 10;
```

## Creación de una tabla para los registros de netflow

1. Modifique el siguiente ejemplo de instrucción DDL para adaptarlo a la estructura de su registro de alertas. Es posible que tenga que actualizar la instrucción para incluir las columnas de la versión más reciente de los registros. Para obtener más información, consulte [Contenido de un registro de firewall](#) en la Guía para desarrolladores de AWS Network Firewall.

```
CREATE EXTERNAL TABLE network_firewall_netflow_logs (  
  firewall_name string,  
  availability_zone string,  
  event_timestamp string,  
  event struct<  
    timestamp:string,  
    flow_id:bigint,  
    event_type:string,  
    src_ip:string,  
    src_port:int,  
    dest_ip:string,  
    dest_port:int,  
    proto:string,  
    app_proto:string,  
    netflow:struct<  
      pkts:int,  
      bytes:bigint,  
      start:string,  
      `end`:string,  
      age:int,  
      min_ttl:int,  
      max_ttl:int,  
      tcp_flags:struct<  
        syn:boolean,  
        fin:boolean,  
        rst:boolean,  
        psh:boolean,  
        ack:boolean,  
        urg:boolean  
      >,  
      tls_inspected:boolean  
    >  
  >  
)  
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
```

```
LOCATION 's3://DOC-EXAMPLE-BUCKET/path_to_netflow_logs_folder';
```

2. Modifique la cláusula LOCATION para especificar la carpeta de sus registros en Amazon S3.
3. Ejecute la consulta CREATE TABLE en el editor de consultas de Athena. Cuando la consulta finaliza, Athena registra la tabla `network_firewall_netflow_logs`, lo que hace que los datos a los que apunta estén listos para las consultas.

## Consulta de ejemplo de registro de Netflow

El ejemplo de consulta de registro de netflow de esta sección filtra los eventos en los que se realizó una inspección de TLS.

La consulta utiliza alias para crear encabezados de columna de salida que muestran a qué struct pertenece la columna. Por ejemplo, el encabezado de la columna del campo `event.netflow.bytes` es `event_netflow_bytes` en lugar de simplemente `bytes`. Para personalizar aún más los nombres de las columnas, puede modificar los alias según sus preferencias. Por ejemplo, puede usar guiones bajos u otros separadores para delimitar los nombres de struct y los nombres de los campos.

Recuerde modificar los nombres y las referencias a struct de las columnas en función de la definición de la tabla y de los campos que desee incluir en el resultado de la consulta.

```
SELECT
  event.src_ip AS event_src_ip,
  event.dest_ip AS event_dest_ip,
  event.proto AS event_proto,
  event.app_proto AS event_app_proto,
  event.netflow.pkts AS event_netflow_pkts,
  event.netflow.bytes AS event_netflow_bytes,
  event.netflow.tcp_flags.syn AS event_netflow_tcp_flags_syn,
  event.netflow.tls_inspected AS event_netflow_tls_inspected
FROM network_firewall_netflow_logs
WHERE event.netflow.tls_inspected = true
```

## Consulta de los registros del Network Load Balancer

Utilice Athena para analizar y procesar los registros del Network Load Balancer. Estos registros reciben información detallada de las solicitudes de seguridad de la capa de transporte (TLS) enviadas al Network Load Balancer. Puede utilizar estos registros de acceso para analizar los patrones de tráfico y solucionar problemas.



Antes de analizar los registros de acceso del Network Load Balancer, habilítelos y configúrelos para que se guarden en el bucket de Amazon S3 de destino. Para obtener más información sobre cada entrada del registro de acceso al Equilibrador de carga de red, consulte [Registros de acceso al Equilibrador de carga de red](#).

- [Cree la tabla para los registros del Network Load Balancer](#)
- [Ejemplos de consulta del Network Load Balancer](#)

Para crear la tabla para los registros del Network Load Balancer

1. Copie y pegue la instrucción DDL siguiente en la consola de Athena. Verifique la [sintaxis](#) de los registros log del Network Load Balancer. Es posible que tenga que actualizar la siguiente consulta para incluir las columnas y la sintaxis Regex de la versión más reciente del registro.

```
CREATE EXTERNAL TABLE IF NOT EXISTS nlb_tls_logs (  
    type string,  
    version string,  
    time string,  
    elb string,  
    listener_id string,  
    client_ip string,  
    client_port int,  
    target_ip string,  
    target_port int,  
    tcp_connection_time_ms double,  
    tls_handshake_time_ms double,  
    received_bytes bigint,  
    sent_bytes bigint,  
    incoming_tls_alert int,  
    cert_arn string,  
    certificate_serial string,  
    tls_cipher_suite string,  
    tls_protocol_version string,  
    tls_named_group string,  
    domain_name string,  
    alpn_fe_protocol string,  
    alpn_be_protocol string,  
    alpn_client_preference_list string,  
    tls_connection_creation_time string  
)  
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
```

```

WITH SERDEPROPERTIES (
  'serialization.format' = '1',
  'input.regex' =
    '([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*):([0-9]*) ([ ]*):([0-9]*)
  ([-.\0-9]*) ([-.\0-9]*) ([-0-9]*) ([-0-9]*) ([-0-9]*) ([ ]*) ([ ]*) ([ ]*)
  ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*) ([ ]*)$')
  LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/AWS_account_ID/
  elasticloadbalancing/region';

```

2. Modifique el bucket de Amazon S3 LOCATION para especificar el destino de los registros del Network Load Balancer.
3. Ejecute la consulta en la consola de Athena. Cuando la consulta finaliza, Athena registra la tabla nlb\_tls\_logs y los datos que contiene quedan disponibles para efectuar consultas.

### Ejemplos de consulta del Network Load Balancer

Para saber cuántas veces se utiliza un certificado, utilice una consulta similar a la de este ejemplo:

```

SELECT count(*) AS
      ct,
      cert_arn
FROM "nlb_tls_logs"
GROUP BY cert_arn;

```

En la siguiente consulta, se muestra la cantidad de usuarios que utilizan una versión de TLS anterior a la versión 1.3:

```

SELECT tls_protocol_version,
      COUNT(tls_protocol_version) AS
      num_connections,
      client_ip
FROM "nlb_tls_logs"
WHERE tls_protocol_version < 'tlsv13'
GROUP BY tls_protocol_version, client_ip;

```

Utilice la siguiente consulta para identificar las conexiones que tardan mucho tiempo en completar el protocolo TLS:

```

SELECT *
FROM "nlb_tls_logs"

```

```
ORDER BY  tls_handshake_time_ms DESC
LIMIT 10;
```

Utilice la siguiente consulta para identificar y contar qué versiones del protocolo TLS y suites de cifrado se negociaron en los últimos 30 días.

```
SELECT  tls_cipher_suite,
        tls_protocol_version,
        COUNT(*) AS ct
FROM    "nlb_tls_logs"
WHERE   from_iso8601_timestamp(time) > current_timestamp - interval '30' day
        AND NOT tls_protocol_version = '-'
GROUP BY  tls_cipher_suite, tls_protocol_version
ORDER BY  ct DESC;
```

## Consulta de los registros de consulta de Amazon Route 53 Resolver

Puede crear tablas de Athena para sus registros de consulta de Amazon Route 53 Resolver y consultarlas desde Athena.

El registro de consultas de Route 53 Resolver se utiliza para el registro de consultas de DNS realizadas por recursos dentro de una VPC, recursos en las instalaciones que utilizan un punto de conexión de Resolver saliente para resolución DNS recursiva y consultas que utilizan reglas de firewall DNS de Route 53 Resolver para bloquear, permitir o monitorear una lista de dominios. Para obtener más información acerca del registro de consultas de Resolver, consulte [Registro de consultas de Resolver](#) en la Guía para desarrolladores de Amazon Route 53. Para obtener información sobre cada uno de los campos de los registros, consulte [Valores que aparecen en los registros de consulta de Resolver](#) en la Guía para desarrolladores de Amazon Route 53.

### Creación de la tabla para los registros de consultas de Resolver

Puede utilizar el Editor de consultas de la consola de Athena para crear y consultar una tabla para los registros de consulta de Route 53 Resolver.

Para crear y consultar una tabla de Athena para los registros de consulta de Route 53 Resolver

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. En el Editor de consultas de Athena, ingrese la siguiente instrucción CREATE TABLE. Reemplace los valores de la cláusula LOCATION con los correspondientes a la ubicación de los registros de Resolver de Amazon S3.

```

CREATE EXTERNAL TABLE r53_rlogs (
  version string,
  account_id string,
  region string,
  vpc_id string,
  query_timestamp string,
  query_name string,
  query_type string,
  query_class
    string,
  rcode string,
  answers array<
    struct<
      Rdata: string,
      Type: string,
      Class: string>
    >,
  srcaddr string,
  srcport int,
  transport string,
  srcids struct<
    instance: string,
    resolver_endpoint: string
  >,
  firewall_rule_action string,
  firewall_rule_group_id string,
  firewall_domain_list_id string
)

ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION 's3://DOC-EXAMPLE-BUCKET/AWSLogs/aws_account_id/vpcdnsquerylogs/{vpc-id}/'

```

Dado que los datos del registro de consulta de Resolver están en formato JSON, la instrucción CREATE TABLE utiliza una [biblioteca de SerDe JSON](#) para analizar los datos.

#### Note

El SerDe espera que cada documento JSON esté en una sola línea de texto sin caracteres de terminación de línea que separen los campos del registro. Si el texto JSON está en formato de impresión, puede recibir un mensaje de error como HIVE\_CURSOR\_ERROR: la fila no es un objeto JSON válido o

HIVE\_CURSOR\_ERROR: JsonParseException: fin de entrada inesperado: marcador de cierre esperado para OBJECT cuando intenta consultar la tabla después de crearla. Para obtener más información, consulte los [Archivos de datos JSON](#) en la documentación de OpenX SerDE en GitHub.

3. Elija Ejecutar consulta. La instrucción crea una tabla de Athena llamada `r53_rlogs` cuyas columnas representan cada uno de los campos de los datos de registro de Resolver.
4. En el Editor de consultas de la consola de Athena, ejecute la siguiente consulta para verificar que se ha creado la tabla.

```
SELECT * FROM "r53_rlogs" LIMIT 10
```

### Ejemplo de partición

En el siguiente ejemplo, se muestra una instrucción `CREATE TABLE` para los registros de consultas de Resolver que utiliza la proyección de particiones y se encuentra particionada por VPC y por fecha. Para obtener más información sobre la proyección de particiones, consulte [Proyección de particiones con Amazon Athena](#).

```
CREATE EXTERNAL TABLE r53_rlogs (  
  version string,  
  account_id string,  
  region string,  
  vpc_id string,  
  query_timestamp string,  
  query_name string,  
  query_type string,  
  query_class string,  
  rcode string,  
  answers array<  
    struct<  
      Rdata: string,  
      Type: string,  
      Class: string>  
  >,  
  srcaddr string,  
  srcport int,  
  transport string,  
  srcids struct<  
    instance: string,
```

```

    resolver_endpoint: string
  >,
  firewall_rule_action string,
  firewall_rule_group_id string,
  firewall_domain_list_id string
)
PARTITIONED BY (
  `date` string,
  `vpc` string
)
ROW FORMAT SERDE      'org.openx.data.jsonserde.JsonSerDe'
STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT          'org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat'
LOCATION                's3://DOC-EXAMPLE-BUCKET/route53-query-logging/
AWSLogs/aws_account_id/vpcdnsquerylogs/'
TBLPROPERTIES(
  'projection.enabled' = 'true',
  'projection.vpc.type' = 'enum',
  'projection.vpc.values' = 'vpc-6446ae02',
  'projection.date.type' = 'date',
  'projection.date.range' = '2023/06/26,NOW',
  'projection.date.format' = 'yyyy/MM/dd',
  'projection.date.interval' = '1',
  'projection.date.interval.unit' = 'DAYS',
  'storage.location.template' = 's3://DOC-EXAMPLE-BUCKET/route53-query-logging/
AWSLogs/aws_account_id/vpcdnsquerylogs/${vpc}/${date}/'
)

```

## Consultas de ejemplo

Los siguientes ejemplos muestran algunas consultas que puede efectuar desde Athena en los registros de consultas de Resolver.

### Ejemplo 1: registros de consulta en orden de `query_timestamp` descendente

La siguiente consulta muestra los resultados del registro en orden `query_timestamp` descendente.

```

SELECT * FROM "r53_rlogs"
ORDER BY query_timestamp DESC

```

## Ejemplo 2: registros de consulta dentro de las horas de inicio y finalización especificadas

La consulta a continuación consulta los registros entre las medianoche y las 8 h del 24 de septiembre de 2020. Sustituye las horas de inicio y finalización de acuerdo con sus propios requisitos.

```
SELECT query_timestamp, srcids.instance, srcaddr, srcport, query_name, rcode
FROM "r53_rlogs"
WHERE (parse_datetime(query_timestamp, 'yyyy-MM-dd' 'T' 'HH:mm:ss' 'Z')
      BETWEEN parse_datetime('2020-09-24-00:00:00', 'yyyy-MM-dd-HH:mm:ss')
      AND parse_datetime('2020-09-24-00:08:00', 'yyyy-MM-dd-HH:mm:ss'))
ORDER BY query_timestamp DESC
```

## Ejemplo 3: registros de consulta basados en un patrón de nombre de consulta de DNS especificado

La siguiente consulta selecciona registros cuyo nombre de consulta incluye la cadena "example.com".

```
SELECT query_timestamp, srcids.instance, srcaddr, srcport, query_name, rcode, answers
FROM "r53_rlogs"
WHERE query_name LIKE '%example.com%'
ORDER BY query_timestamp DESC
```

## Ejemplo 4: solicitudes de registro de consultas sin respuesta

La siguiente consulta selecciona entradas de registro en las que la solicitud no recibió respuesta.

```
SELECT query_timestamp, srcids.instance, srcaddr, srcport, query_name, rcode, answers
FROM "r53_rlogs"
WHERE cardinality(answers) = 0
```

## Ejemplo 5: registros de consultas con una respuesta específica

La siguiente consulta muestra los registros en los que el valor `answer.Rdata` tiene la dirección IP especificada.

```
SELECT query_timestamp, srcids.instance, srcaddr, srcport, query_name, rcode,
       answer.Rdata
FROM "r53_rlogs"
CROSS JOIN UNNEST(r53_rlogs.answers) as st(answer)
WHERE answer.Rdata='203.0.113.16';
```

## Consulta de registros de eventos de Amazon SES

Puede utilizar Amazon Athena para consultar registros de eventos de [Amazon Simple Email Service](#) (Amazon SES).

Amazon SES es una plataforma de correo electrónico que ofrece un método sencillo y rentable de envío y recepción de correo electrónico a través de sus propios dominios y direcciones de correo electrónico. Puede supervisar la actividad de envío de Amazon SES a un nivel detallado mediante eventos, métricas y estadísticas.

Según las características que defina, puede publicar eventos de Amazon SES en [Amazon CloudWatch](#), [Amazon Data Firehose](#) o [Amazon Simple Notification Service](#). Después de almacenar la información en Amazon S3, puede consultarla desde Amazon Athena.

Para obtener información sobre cómo analizar los eventos de correo electrónico de Amazon SES con Firehose, Amazon Athena y Amazon QuickSight, consulte [Analizar datos de eventos de Amazon SES con los servicios de análisis de AWS](#) en el Blog de mensajería y segmentación de AWS.

## Consulta de los registros de flujo de Amazon VPC

Los registros de flujo de Amazon Virtual Private Cloud capturan información acerca del tráfico IP entrante y saliente por las interfaces de red de una VPC. Utilice los registros para investigar los patrones de tráfico de red e identificar las amenazas y los riesgos en la red de VPC.

Tiene dos opciones para consultar los registros de flujo de Amazon VPC:

- **Consola de Amazon VPC:** utilice la característica de integración de Athena en la consola de Amazon VPC para generar una plantilla de AWS CloudFormation que crea una base de datos, un grupo de trabajo y una tabla de registros de flujo de Athena con particiones por usted. La plantilla también crea un conjunto de [consultas de registro de flujo predefinidas](#) que puede utilizar para obtener información sobre el tráfico que fluye a través de la VPC.

Para obtener más información sobre este enfoque, consulte [Realizar consultas en los registros de flujo mediante Amazon Athena](#) en la Guía del usuario de Amazon VPC.

- **Consola Amazon Athena:** cree tablas y consultas directamente en la consola de Athena. Para obtener más información, continúe leyendo esta página.



## Creación y consulta de tablas para los registros de flujo de VPC personalizados

Antes de comenzar la consulta de los registros en Athena, [habilite los registros de flujo de VPC](#) y configúrelos para que se guarden en el bucket de Amazon S3. Después de crear los registros, deje que se ejecuten durante unos minutos para recopilar algunos datos. Los registros se crean con un formato de compresión GZIP que Athena le permite consultar directamente.

Al crear un registro de flujo de VPC, puede utilizar un formato personalizado cuando quiera especificar los campos que se devolverán en el registro de flujo y el orden en que aparecerán. Para obtener más información sobre los registros de flujo, consulte [Registros de flujo](#) en la Guía del usuario de Amazon VPC.

### Consideraciones comunes

Cuando cree tablas en los registros de flujo de Athena para Amazon VPC, recuerde los puntos a continuación:

- De forma predeterminada, en Athena, Parquet accede a las columnas por su nombre. Para obtener más información, consulte [Gestión de las actualizaciones de los esquemas](#).
- Utilice los nombres de los registros del registro de flujo en los nombres de las columnas de Athena. Los nombres de las columnas del esquema de Athena deben coincidir exactamente con los nombres de campo de los registros de flujo de Amazon VPC, con las siguientes diferencias:
  - Sustituya los guiones de los nombres de los campos de registro de Amazon VPC por guiones bajos en los nombres de columna de Athena. En Athena, los únicos caracteres aceptables para los nombres de base de datos, los nombres de tablas y los nombres de columnas son las letras minúsculas, los números y los guiones bajos. Para obtener más información, consulte [Nombres de base de datos, tablas y columnas](#).
  - Escape los nombres de registros del registro de flujo que sean [palabras clave reservadas](#) en Athena encerrándolos con acentos graves.
- Los registros de flujo de VPC son específicos de la Cuenta de AWS. Cuando publica sus archivos de registro en Amazon S3, la ruta que Amazon VPC crea en Amazon S3 incluye el ID de la Cuenta de AWS que se usó para crear el registro de flujo. Para obtener más información, consulte [Publicar registros de flujo en Amazon S3](#) en la Guía del usuario de Amazon VPC.

### Instrucción CREATE TABLE para los registros de flujo de Amazon VPC

En el siguiente procedimiento, se crea una tabla de Amazon VPC para los registros de flujo de Amazon VPC. Cuando crea un registro de flujo con un formato personalizado, crea una tabla con

campos que coinciden con los campos especificados al crear el registro de flujo, en el mismo orden en que los especificó.

Para crear una tabla de Athena para los registros de flujo de Amazon VPC

1. Ingrese una instrucción DDL como la siguiente en el editor de consultas de la consola de Athena y siga las directrices de la sección [Consideraciones comunes](#). Con la instrucción de muestra, se crea una tabla que tiene las columnas para los registros de flujo de Amazon VPC (versiones 2 a 5), como se documenta en [Registros de flujo](#). Si utiliza un conjunto o un orden de columnas diferente, modifique la instrucción según corresponda.

```
CREATE EXTERNAL TABLE IF NOT EXISTS `vpc_flow_logs` (  
  version int,  
  account_id string,  
  interface_id string,  
  srcaddr string,  
  dstaddr string,  
  srcport int,  
  dstport int,  
  protocol bigint,  
  packets bigint,  
  bytes bigint,  
  start bigint,  
  `end` bigint,  
  action string,  
  log_status string,  
  vpc_id string,  
  subnet_id string,  
  instance_id string,  
  tcp_flags int,  
  type string,  
  pkt_srcaddr string,  
  pkt_dstaddr string,  
  region string,  
  az_id string,  
  sublocation_type string,  
  sublocation_id string,  
  pkt_src_aws_service string,  
  pkt_dst_aws_service string,  
  flow_direction string,  
  traffic_path int  
)
```

```
PARTITIONED BY (`date` date)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ' '
LOCATION 's3://DOC-EXAMPLE-BUCKET/prefix/AWSLogs/{account_id}/
vpcflowlogs/{region_code}/'
TBLPROPERTIES ("skip.header.line.count"="1");
```

Tenga en cuenta los siguientes puntos:

- La consulta especifica ROW FORMAT DELIMITED y omite especificar un elemento SerDe. Esto significa que la consulta utiliza [LazySimpleSerDe para CSV, TSV y archivos con delimitación personalizada](#). En esta consulta los campos terminan con un espacio.
- La cláusula PARTITIONED BY utiliza el tipo date. Esto permite utilizar operadores matemáticos en las consultas para seleccionar los elementos más antiguos o más recientes respecto a una fecha determinada.

#### Note

Dado que date es una palabra clave reservada en las instrucciones DDL, se incluye en una secuencia de escape con caracteres de comilla simple. Para obtener más información, consulte [Palabras clave reservadas](#).

- En el caso de un registro de flujo de VPC con un formato personalizado diferente, modifique los campos para que coincidan con los campos especificados al crear el registro de flujo.
2. Modifique LOCATION 's3://DOC-EXAMPLE-BUCKET/*prefix*/AWSLogs/{*account\_id*}/vpcflowlogs/{*region\_code*}/' de modo que apunte al bucket de Amazon S3 que contiene los datos de registro.
  3. Ejecute la consulta en la consola de Athena. Cuando la consulta finaliza, Athena registra la tabla vpc\_flow\_logs, dejando los datos que contiene listos para efectuar consultas.
  4. Cree particiones para poder leer los datos, tal y como se indica en la siguiente consulta de ejemplo. Esta consulta de ejemplo crea una sola partición para la fecha especificada. Sustituya los marcadores de posición para la fecha y la ubicación según sea necesario.

#### Note

Esta consulta crea solo una partición única para la fecha que especifique. Para automatizar el proceso, utilice un script que ejecute esta consulta y que cree las

particiones de este modo para `year/month/day`, o bien utilice una instrucción `CREATE TABLE` en la que se especifique la [proyección de particiones](#).

```
ALTER TABLE vpc_flow_logs
ADD PARTITION (`date`='YYYY-MM-dd')
LOCATION 's3://DOC-EXAMPLE-BUCKET/prefix/AWSLogs/{account_id}/
vpcflowlogs/{region_code}/YYYY/MM/dd';
```

## Consultas de ejemplo para la tabla `vpc_flow_logs`

Utilice el editor de consultas de la consola de Athena para ejecutar instrucciones SQL en la tabla que cree. Puede guardar las consultas, ver consultas anteriores o descargar los resultados de las consultas en formato CSV. En los siguientes ejemplos, sustituya `vpc_flow_logs` por el nombre de la tabla. Modifique los valores de columna y otras variables según sus requisitos.

La siguiente consulta de ejemplo muestra un máximo de 100 registros de flujo para la fecha especificada.

```
SELECT *
FROM vpc_flow_logs
WHERE date = DATE('2020-05-04')
LIMIT 100;
```

La siguiente consulta obtiene una lista de todas las conexiones TCP rechazadas y utiliza la columna de partición de fecha que acaba de crear, `date`, para extraer de ella el día de la semana en el que se produjeron esos eventos.

```
SELECT day_of_week(date) AS
    day,
    date,
    interface_id,
    srcaddr,
    action,
    protocol
FROM vpc_flow_logs
WHERE action = 'REJECT' AND protocol = 6
LIMIT 100;
```

Para saber qué servidor está recibiendo el mayor número de solicitudes HTTPS, utilice la siguiente consulta. Cuenta el número de paquetes recibidos en el puerto 443 de HTTPS, los agrupa según la dirección IP de destino y devuelve los 10 principales de la última semana.

```
SELECT SUM(packets) AS
  packetcount,
  dstaddr
FROM vpc_flow_logs
WHERE dstport = 443 AND date > current_date - interval '7' day
GROUP BY dstaddr
ORDER BY packetcount DESC
LIMIT 10;
```

### Creación de tablas para los registros de flujo en formato Apache Parquet

En el siguiente procedimiento, se crea una tabla de Amazon VPC para los registros de flujo de Amazon VPC en formato Apache Parquet.

Para crear una tabla de Athena para los registros de flujo de Amazon VPC en formato Parquet

1. Ingrese una instrucción DDL como la siguiente en el editor de consultas de la consola de Athena y siga las directrices de la sección [Consideraciones comunes](#). Con la instrucción de muestra, se crea una tabla que tiene las columnas para los registros de flujo de Amazon VPC (versiones 2 a 5), como se documenta en [Registros de flujo](#) en formato Parquet, particionada en Hive por hora. Si no tiene particiones por hora, elimine `hour` de la cláusula `PARTITIONED BY`.

```
CREATE EXTERNAL TABLE IF NOT EXISTS vpc_flow_logs_parquet (
  version int,
  account_id string,
  interface_id string,
  srcaddr string,
  dstaddr string,
  srcport int,
  dstport int,
  protocol bigint,
  packets bigint,
  bytes bigint,
  start bigint,
  `end` bigint,
  action string,
  log_status string,
  vpc_id string,
```

```
    subnet_id string,  
    instance_id string,  
    tcp_flags int,  
    type string,  
    pkt_srcaddr string,  
    pkt_dstaddr string,  
    region string,  
    az_id string,  
    sublocation_type string,  
    sublocation_id string,  
    pkt_src_aws_service string,  
    pkt_dst_aws_service string,  
    flow_direction string,  
    traffic_path int  
  )  
PARTITIONED BY (  
  `aws-account-id` string,  
  `aws-service` string,  
  `aws-region` string,  
  `year` string,  
  `month` string,  
  `day` string,  
  `hour` string  
)  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'  
STORED AS INPUTFORMAT  
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'  
LOCATION  
  's3://DOC-EXAMPLE-BUCKET/prefix/AWSLogs/'  
TBLPROPERTIES (  
  'EXTERNAL'='true',  
  'skip.header.line.count'='1'  
)
```

2. Modifique la muestra LOCATION 's3://DOC-EXAMPLE-BUCKET/*prefix*/AWSLogs/' para que apunte a la ruta de Amazon S3 que contiene los datos de registro.
3. Ejecute la consulta en la consola de Athena.

4. Si sus datos están en formato compatible con HIVE, ejecute el siguiente comando en la consola de Athena para actualizar y cargar las particiones de Hive en el almacén de metadatos. Una vez completada la consulta, puede consultar los datos en la tabla `vpc_flow_logs_parquet`.

```
MSCK REPAIR TABLE vpc_flow_logs_parquet
```

Si no utiliza datos compatibles con Hive, ejecute [ALTER TABLE ADD PARTITION](#) para cargar las particiones.

Para obtener más información sobre el uso de Athena para consultar los registros de flujo de Amazon VPC en formato Parquet, consulte la publicación [Optimice el rendimiento y reduzca los costos de los análisis de red con los registros de flujo de VPC en formato Apache Parquet](#) en el Blog de macrodatos de AWS.

### Creación y consulta de una tabla para los registros de flujo de Amazon VPC por medio de la proyección de particiones

Use la instrucción `CREATE TABLE` como la siguiente para crear una tabla, particionar la tabla y rellenar las particiones automáticamente mediante la [proyección de particiones](#). Sustituya el nombre de la tabla `test_table_vpclogs` en el ejemplo con el nombre de su tabla. Edite la cláusula `LOCATION` para especificar el bucket de Amazon S3 que contiene los datos de registro de Amazon VPC.

La siguiente instrucción `CREATE TABLE` es para los registros de flujo de VPC entregados en formato de partición que no es de estilo Hive. El ejemplo permite agregar varias cuentas. Si va a centralizar los registros de flujo de VPC de varias cuentas en un bucket de Amazon S3, el ID de la cuenta debe introducirse en la ruta de Amazon S3.

```
CREATE EXTERNAL TABLE IF NOT EXISTS test_table_vpclogs (  
  version int,  
  account_id string,  
  interface_id string,  
  srcaddr string,  
  dstaddr string,  
  srcport int,  
  dstport int,  
  protocol bigint,  
  packets bigint,  
  bytes bigint,
```

```

start bigint,
`end` bigint,
action string,
log_status string,
vpc_id string,
subnet_id string,
instance_id string,
tcp_flags int,
type string,
pkt_srcaddr string,
pkt_dstaddr string,
az_id string,
sublocation_type string,
sublocation_id string,
pkt_src_aws_service string,
pkt_dst_aws_service string,
flow_direction string,
traffic_path int
)
PARTITIONED BY (accid string, region string, day string)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ' '
LOCATION '$LOCATION_OF_LOGS'
TBLPROPERTIES
(
"skip.header.line.count"="1",
"projection.enabled" = "true",
"projection.accid.type" = "enum",
"projection.accid.values" = "$ACCID_1,$ACCID_2",
"projection.region.type" = "enum",
"projection.region.values" = "$REGION_1,$REGION_2,$REGION_3",
"projection.day.type" = "date",
"projection.day.range" = "$START_RANGE,NOW",
"projection.day.format" = "yyyy/MM/dd",
"storage.location.template" = "s3://DOC-EXAMPLE-BUCKET/AWSLogs/${accid}/vpcflowlogs/
${region}/${day}"
)

```

## Consultas de ejemplo para test\_table\_vpclogs

Las consultas de ejemplo a continuación consultan los test\_table\_vpclogs creados por la instrucción CREATE TABLE anterior. Reemplace test\_table\_vpclogs en las consultas por el nombre de su propia tabla. Modifique los valores de columna y otras variables según sus requisitos.



Para devolver las 100 primeras entradas de registro de acceso en orden cronológico durante un período especificado, ejecute una consulta como la siguiente.

```
SELECT *
FROM test_table_vpclogs
WHERE day >= '2021/02/01' AND day < '2021/02/28'
ORDER BY day ASC
LIMIT 100
```

Para ver qué servidor recibe el mayor número de paquetes HTTP durante un período especificado, ejecute una consulta como la siguiente. La consulta cuenta el número de paquetes recibidos en el puerto 443 HTTPS, los agrupa según la dirección IP de destino y devuelve las 10 principales entradas de la semana anterior.

```
SELECT SUM(packets) AS packetcount,
       dstaddr
FROM test_table_vpclogs
WHERE dstport = 443
      AND day >= '2021/03/01'
      AND day < '2021/03/31'
GROUP BY dstaddr
ORDER BY packetcount DESC
LIMIT 10
```

Para devolver los registros creados durante un período especificado, ejecute una consulta como la siguiente.

```
SELECT interface_id,
       srcaddr,
       action,
       protocol,
       to_iso8601(from_unixtime(start)) AS start_time,
       to_iso8601(from_unixtime("end")) AS end_time
FROM test_table_vpclogs
WHERE DAY >= '2021/04/01'
      AND DAY < '2021/04/30'
```

Para devolver los registros de acceso de una dirección IP de origen entre periodos especificados, ejecute una consulta como la siguiente.

```
SELECT *
```

```
FROM test_table_vpclogs
WHERE srcaddr = '10.117.1.22'
    AND day >= '2021/02/01'
    AND day < '2021/02/28'
```

Para enumerar las conexiones TCP rechazadas, ejecute una consulta como la siguiente.

```
SELECT day,
       interface_id,
       srcaddr,
       action,
       protocol
FROM test_table_vpclogs
WHERE action = 'REJECT' AND protocol = 6 AND day >= '2021/02/01' AND day < '2021/02/28'
LIMIT 10
```

Para devolver los registros de acceso del intervalo de direcciones IP que comienza por 10.117, ejecute una consulta como la siguiente.

```
SELECT *
FROM test_table_vpclogs
WHERE split_part(srcaddr, '.', 1)='10'
    AND split_part(srcaddr, '.', 2) = '117'
```

Para devolver los registros de acceso de una dirección IP de destino entre un intervalo de tiempo determinado, ejecute una consulta como la siguiente.

```
SELECT *
FROM test_table_vpclogs
WHERE dstaddr = '10.0.1.14'
    AND day >= '2021/01/01'
    AND day < '2021/01/31'
```

### Creación de tablas para los registros de flujo en formato Apache Parquet mediante la proyección de particiones

La siguiente instrucción CREATE TABLE de proyección de particiones para los registros de flujo de VPC se encuentra en formato Apache Parquet, no es compatible con Hive, y se divide por hora y fecha en lugar de por día. Sustituya el nombre de la tabla test\_table\_vpclogs\_parquet en el ejemplo con el nombre de su tabla. Edite la cláusula LOCATION para especificar el bucket de Amazon S3 que contiene los datos de registro de Amazon VPC.

```
CREATE EXTERNAL TABLE IF NOT EXISTS test_table_vpclogs_parquet (  
  version int,  
  account_id string,  
  interface_id string,  
  srcaddr string,  
  dstaddr string,  
  srcport int,  
  dstport int,  
  protocol bigint,  
  packets bigint,  
  bytes bigint,  
  start bigint,  
  `end` bigint,  
  action string,  
  log_status string,  
  vpc_id string,  
  subnet_id string,  
  instance_id string,  
  tcp_flags int,  
  type string,  
  pkt_srcaddr string,  
  pkt_dstaddr string,  
  az_id string,  
  sublocation_type string,  
  sublocation_id string,  
  pkt_src_aws_service string,  
  pkt_dst_aws_service string,  
  flow_direction string,  
  traffic_path int  
)  
PARTITIONED BY (region string, date string, hour string)  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'  
STORED AS INPUTFORMAT  
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'  
LOCATION 's3://DOC-EXAMPLE-BUCKET/prefix/AWSLogs/{account_id}/vpcflowlogs/'  
TBLPROPERTIES (  
  "EXTERNAL"="true",  
  "skip.header.line.count" = "1",  
  "projection.enabled" = "true",  
  "projection.region.type" = "enum",
```

```
"projection.region.values" = "us-east-1,us-west-2,ap-south-1,eu-west-1",
"projection.date.type" = "date",
"projection.date.range" = "2021/01/01,NOW",
"projection.date.format" = "yyyy/MM/dd",
"projection.hour.type" = "integer",
"projection.hour.range" = "00,23",
"projection.hour.digits" = "2",
"storage.location.template" = "s3://DOC-EXAMPLE-BUCKET/prefix/AWSLogs/{account_id}/
vpcflowlogs/{region}/{date}/{hour}"
)
```

## Recursos adicionales de

Para obtener más información acerca de cómo utilizar Athena para consultar registros, consulte las siguientes entradas del Blog de macrodatos de AWS:

- [Analyze VPC Flow Logs with point-and-click Amazon Athena integration](#) (Análisis de los registros de flujo de VPC con una integración de Amazon Athena de apuntar y hacer clic)
- [Analyzing VPC flow logs using Amazon Athena and Amazon QuickSight](#) (Análisis de los registros de flujo de VPC con Amazon Athena y Amazon QuickSight)
- [Optimize performance and reduce costs for network analytics with VPC Flow Logs in Apache Parquet format](#) (Optimización del rendimiento y reducción de los costos de análisis de red con registros de flujo de VPC en formato Apache Parquet)

## Consulta de registros de AWS WAF

AWS WAF es un firewall de aplicaciones web que permite supervisar y controlar las solicitudes HTTP y HTTPS que reciben sus aplicaciones web protegidas de los clientes. Usted define cómo gestionar las solicitudes web configurando las reglas dentro de una lista de control de acceso (ACL) web AWS WAF. A continuación, se protege una aplicación web asociándole una ACL web. Entre los ejemplos de recursos de aplicaciones web que puede proteger con AWS WAF incluyen las distribuciones de Amazon CloudFront, las API de REST de Amazon API Gateway y los equilibradores de carga de aplicación. Para obtener más información acerca de las AWS WAF, consulte [AWS WAF](#) en la Guía para desarrolladores de AWS WAF.

Los registros de AWS WAF incluyen información sobre el tráfico que analiza su ACL web, como la hora en que AWS WAF recibió la solicitud de su recurso de AWS, información detallada sobre la solicitud y la acción de la regla con la que coincide cada solicitud.

Puede configurar una web ACL AWS WAF para publicar los registros en uno de varios destinos, donde podrá consultarlos y verlos. Para obtener más información sobre cómo configurar el registro de ACL web y el contenido de los registros AWS WAF, consulte [Registrar el tráfico de ACL AWS WAF web](#) en la Guía para desarrolladores de AWS WAF.

Para ver un ejemplo sobre cómo agregar registros AWS WAF en un repositorio de lago de datos central y consultarlos con Athena, consulte la publicación del Blog de macrodatos de AWS [Análisis de registros AWS WAF con OpenSearch Service, Amazon Athena y Amazon QuickSight](#).

Este tema ofrece dos ejemplos de instrucciones CREATE TABLE: una que usa particiones y otra que no.

#### Note

Las instrucciones CREATE TABLE de este tema se pueden usar tanto para registros AWS WAF v1 como v2. En v1, el campo `webaclid` contiene un ID. En v2, el campo `webaclid` contiene un ARN completo. Las instrucciones CREATE TABLE aquí tratan este contenido de forma agnóstica mediante el uso de tipos de datos `string`.

## Temas

- [Creación de una tabla para los registros S3 de AWS WAF en Athena por medio de la proyección de particiones](#)
- [Creación de una tabla para registros de AWS WAF sin particionar](#)
- [Ejemplos de consultas de registros de AWS WAF](#)

## Creación de una tabla para los registros S3 de AWS WAF en Athena por medio de la proyección de particiones

Dado que los registros de AWS WAF tienen una estructura conocida cuyo esquema de partición puede especificar de antemano, puede reducir el tiempo de ejecución de las consultas y automatizar la administración de particiones mediante la característica de [proyección de particiones](#) de Athena. La proyección de particiones agrega de forma automática nuevas particiones a medida que se agregan nuevos datos. Esto hace que no sea necesario agregar particiones manualmente mediante `ALTER TABLE ADD PARTITION`.

La siguiente instrucción de ejemplo CREATE TABLE utiliza automáticamente la proyección de particiones en registros AWS WAF desde una fecha especificada hasta el presente para

cuatro regiones diferentes de AWS. La cláusula `PARTITION BY` de este ejemplo particiona por región y por fecha, pero puede modificarla según sus requisitos. Modifique los campos según sea necesario para que coincidan con el resultado del registro. En las cláusulas `LOCATION` y `storage.location.template`, reemplace los marcadores de posición de *bucket* y *accountID* por valores que identifiquen la ubicación del bucket de Amazon S3 de los registros de AWS WAF. Para `projection.day.range`, reemplace *2021/01/01* por la fecha de inicio que quiere utilizar. Una vez ejecutada la consulta correctamente, puede consultar la tabla. No tiene que ejecutar `ALTER TABLE ADD PARTITION` para cargar las particiones.

```
CREATE EXTERNAL TABLE `waf_logs`(
  `timestamp` bigint,
  `formatversion` int,
  `webaclid` string,
  `terminatingruleid` string,
  `terminatingruletype` string,
  `action` string,
  `terminatingrulematchdetails` array <
    struct <
      conditiontype: string,
      sensitivitylevel: string,
      location: string,
      matcheddata: array < string >
    >
  >,
  `httpsourcename` string,
  `httpsourceid` string,
  `rulegrouplist` array <
    struct <
      rulegroupid: string,
      terminatingrule: struct <
        ruleid: string,
        action: string,
        rulematchdetails: array <
          struct <
            conditiontype:
string,
            sensitivitylevel: string,
            location:
string,
            matcheddata:
array < string >
```

```

    >
    >
    >,
    nonterminatingmatchingrules: array <
        struct <
            ruleid: string,
            action: string,
            overriddenaction:
                string,
            rulematchdetails:
                array <
                    struct <
                        conditiontype: string,
                        sensitivitylevel: string,
                        location: string,
                        matcheddata: array < string >
                    >
                >
            challengerresponse:
                struct <
                    responsecode: string,
                    solvetimestamp: string
                >
            captcharesponse:
                struct <
                    responsecode: string,
                    solvetimestamp: string
                >
        >
    >,
    excludedrules: string
    >
    >,
    `ratebasedrulelist` array <

```

```

        struct <
            ratebasedruleid: string,
            limitkey: string,
            maxrateallowed: int
        >
    >,
`nonterminatingmatchingrules` array <
    struct <
        ruleid: string,
        action: string,
        rulematchdetails: array <
            struct <
                conditiontype: string,
                sensitivitylevel:
string,

                location: string,
                matcheddata: array <
string >
            >
        >,
        challengerresponse: struct <
            responsecode: string,
            solvetimestamp: string
        >,
        captcharesponse: struct <
            responsecode: string,
            solvetimestamp: string
        >
    >
>,
`requestheadersinserted` array <
    struct <
        name: string,
        value: string
    >
>,
`responsecodesent` string,
`httprequest` struct <
    clientip: string,
    country: string,
    headers: array <
        struct <
            name: string,
            value: string
    >
    >

```



```

        >
        >,
        uri: string,
        args: string,
        httpversion: string,
        httpmethod: string,
        requestid: string
    >,
`labels` array <
    struct <
        name: string
    >
>,
`captcharesponse` struct <
    responsecode: string,
    solvetimestamp: string,
    failureReason: string
    >,
`challengeresponse` struct <
    responsecode: string,
    solvetimestamp: string,
    failureReason: string
    >,
`ja3Fingerprint` string,
`oversizefields` string,
`requestbodysize` int,
`requestbodysizeinspectedbywaf` int
)
PARTITIONED BY (
`region` string,
`date` string)
ROW FORMAT SERDE
    'org.openx.data.jsonserde.JsonSerDe'
STORED AS INPUTFORMAT
    'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
    'org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
    's3://DOC-EXAMPLE-BUCKET/AWSLogs/accountID/WAFLogs/region/DOC-EXAMPLE-WEBACL/'
TBLPROPERTIES(
'projection.enabled' = 'true',
'projection.region.type' = 'enum',
'projection.region.values' = 'us-east-1,us-west-2,eu-central-1,eu-west-1',
'projection.date.type' = 'date',

```

```
'projection.date.range' = '2021/01/01,NOW',  
'projection.date.format' = 'yyyy/MM/dd',  
'projection.date.interval' = '1',  
'projection.date.interval.unit' = 'DAYS',  
'storage.location.template' = 's3://DOC-EXAMPLE-BUCKET/AWSLogs/accountID/WAFLogs/  
${region}/DOC-EXAMPLE-WEBACL/${date}/')
```

### Note

El formato de la ruta de la cláusula LOCATION del ejemplo es estándar, pero puede variar en función de la configuración de AWS WAF que haya implementado. Por ejemplo, el siguiente ejemplo de ruta de registros de AWS WAF es para una distribución de CloudFront:

```
s3://DOC-EXAMPLE-BUCKET/AWSLogs/12345678910/WAFLogs/cloudfront/  
cloudfrontyt/2022/08/08/17/55/
```

Si tiene problemas al crear o consultar la tabla de registros de AWS WAF, confirme la ubicación de los datos de registro o [póngase en contacto con AWS Support](#).

Para obtener más información sobre la proyección de particiones, consulte [Proyección de particiones con Amazon Athena](#).

## Creación de una tabla para registros de AWS WAF sin particionar

En esta sección se describe cómo crear una tabla para registros AWS WAF sin particiones ni proyección de particiones.

### Note

Por motivos de rendimiento y costo, no le recomendamos utilizar un esquema no particionado para las consultas. Para obtener información, consulte [Los 10 principales consejos de ajuste de rendimiento de Amazon Athena](#) en el Blog de macrodatos de AWS.

## Para crear la tabla de AWS WAF

1. Copie y pegue la siguiente instrucción DDL en la consola de Athena. Modifique los campos según sea necesario para que coincidan con el resultado del registro. Modifique el valor de LOCATION

para indicar el bucket de Amazon S3 que se corresponda con el bucket donde se almacenan los registros.

Esta consulta utiliza [El SerDe JSON de OpenX](#).

### Note

El SerDe espera que cada documento JSON esté en una sola línea de texto sin caracteres de terminación de línea que separen los campos del registro. Si el texto JSON está en formato de impresión, puede recibir un mensaje de error como `HIVE_CURSOR_ERROR: la fila no es un objeto JSON válido` o `HIVE_CURSOR_ERROR: JsonParseException: fin de entrada inesperado: marcador de cierre esperado para OBJECT cuando intenta consultar la tabla después de crearla`. Para obtener más información, consulte los [Archivos de datos JSON](#) en la documentación de OpenX SerDE en GitHub.

```
CREATE EXTERNAL TABLE `waf_logs` (
  `timestamp` bigint,
  `formatversion` int,
  `webaclid` string,
  `terminatingruleid` string,
  `terminatingruletype` string,
  `action` string,
  `terminatingrulematchdetails` array <
    struct <
      conditiontype: string,
      sensitivitylevel: string,
      location: string,
      matcheddata: array < string >
    >
  >,
  `httpsourcename` string,
  `httpsourceid` string,
  `rulegrouplist` array <
    struct <
      rulegroupid: string,
      terminatingrule: struct <
        ruleid: string,
        action: string,
        rulematchdetails: array <
          struct <
```

```

string,
sensitivitylevel: string,
string,
array < string >
conditiontype:
location:
matcheddata:
>
>
>,
nonterminatingmatchingrules: array <
struct <
ruleid: string,
action: string,
overriddenaction:
string,
rulematchdetails:
array <
struct <
conditiontype: string,
sensitivitylevel: string,
location: string,
matcheddata: array < string >
>
>,
challengeresponse:
struct <
responsecode: string,
solvetimestamp: string
>,
captcharesponse:
struct <
responsecode: string,

```

```

solvetimestamp: string
                                >
                                >
                                >,
                                excludedrules: string
                                >
                                >,
`ratebasedrulelist` array <
    struct <
        ratebasedruleid: string,
        limitkey: string,
        maxrateallowed: int
    >
    >,
`nonterminatingmatchingrules` array <
    struct <
        ruleid: string,
        action: string,
        rulematchdetails: array <
            struct <
                conditiontype:
                sensitivitylevel:
                location: string,
                matcheddata: array <
                    string >
                    string >
                    string >
            >
        >,
        challengerresponse: struct <
            responsecode: string,
            solvetimestamp: string
        >,
        captcharesponse: struct <
            responsecode: string,
            solvetimestamp: string
        >
    >
    >,
`requestheadersinserted` array <
    struct <
        name: string,
        value: string
    >

```

```

        >
        >,
`responsecodesent` string,
`httprequest` struct <
    clientip: string,
    country: string,
    headers: array <
        struct <
            name: string,
            value: string
        >
    >,
    uri: string,
    args: string,
    httpversion: string,
    httpmethod: string,
    requestid: string
    >,
`labels` array <
    struct <
        name: string
    >
    >,
`captcharesponse` struct <
    responsecode: string,
    solvetimestamp: string,
    failureReason: string
    >,
`challengeresponse` struct <
    responsecode: string,
    solvetimestamp: string,
    failureReason: string
    >,
`ja3Fingerprint` string,
`oversizefields` string,
`requestbodysize` int,
`requestbodysizeinspectedbywaf` int
)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://DOC-EXAMPLE-BUCKET/prefix'

```

2. Ejecute la instrucción `CREATE EXTERNAL TABLE` en el editor de consultas de la consola de Athena. Esto registra la tabla `waf_logs` y hace que los datos que contiene estén disponibles para las consultas de Athena.

## Ejemplos de consultas de registros de AWS WAF

En muchos de los siguientes ejemplos de consultas se utiliza la tabla de proyección de particiones creada anteriormente en este documento. Modifique el nombre de la tabla, los valores de columna y otras variables de los ejemplos según sus requisitos. Para mejorar el rendimiento de las consultas y reducir el costo, agregue la columna de partición en la condición de filtro.

- [Count the number of referrers that contain a specified term](#)
- [Count all matched IP addresses in the last 10 days that have matched excluded rules](#)
- [Group all counted managed rules by the number of times matched](#)
- [Group all counted custom rules by number of times matched](#)

## Uso de fecha y hora

- [Return the timestamp field in human-readable ISO 8601 format](#)
- [Return records from the last 24 hours](#)
- [Return records for a specified date range and IP address](#)
- [For a specified date range, count the number of IP addresses in five minute intervals](#)
- [Count the number of X-Forwarded-For IP in the last 10 days](#)

## Trabajar con solicitudes y direcciones bloqueadas

- [Extract the top 100 IP addresses blocked by a specified rule type](#)
- [Count the number of times a request from a specified country has been blocked](#)
- [Count the number of times a request has been blocked, grouping by specific attributes](#)
- [Count the number of times a specific terminating rule ID has been matched](#)
- [Retrieve the top 100 IP addresses blocked during a specified date range](#)

Example : contar el número de referencias que contienen un término especificado

La siguiente consulta cuenta el número de referencias que contienen el término “amazon” para el intervalo de fechas especificado.

```
WITH test_dataset AS
  (SELECT header FROM waf_logs
   CROSS JOIN UNNEST(httprequest.headers) AS t(header) WHERE "date" >= '2021/03/01'
   AND "date" < '2021/03/31')
SELECT COUNT(*) referer_count
FROM test_dataset
WHERE LOWER(header.name)='referer' AND header.value LIKE '%amazon%'
```

Example : contar todas las direcciones IP coincidentes en los últimos 10 días que coincidieron con las reglas excluidas

En la siguiente consulta se cuenta el número de veces que, en los últimos 10 días, la dirección IP coincidió con la regla excluida del grupo de reglas.

```
WITH test_dataset AS
  (SELECT * FROM waf_logs
   CROSS JOIN UNNEST(rulegrouplist) AS t(allrulegroups))
SELECT
  COUNT(*) AS count,
  "httprequest"."clientip",
  "allrulegroups"."excludedrules",
  "allrulegroups"."ruleGroupId"
FROM test_dataset
WHERE allrulegroups.excludedrules IS NOT NULL AND from_unixtime(timestamp/1000) > now()
  - interval '10' day
GROUP BY "httprequest"."clientip", "allrulegroups"."ruleGroupId",
  "allrulegroups"."excludedrules"
ORDER BY count DESC
```

Example : agrupar todas las reglas administradas contadas por el número de veces que coinciden

Si estableció las acciones de reglas del grupo de reglas en Contar en su configuración de ACL web antes del 27 de octubre de 2022, AWS WAF guardó las anulaciones en el archivo JSON de ACL web como `excludedRules`. Ahora, la configuración JSON para anular una regla en



Contar se encuentra en la configuración `ruleActionOverrides`. Para obtener más información, consulte [Anulación de acciones en grupos de reglas](#) en la Guía para desarrolladores de AWS WAF. Para extraer las reglas administradas en el modo Contar de la estructura de registro nueva, consulte `nonTerminatingMatchingRules` en la sección `ruleGroupList` en lugar del campo `excludedRules`, como en el siguiente ejemplo.

```
SELECT
  count(*) AS count,
  httpsourceid,
  httprequest.clientip,
  t.rulegroupid,
  t.nonTerminatingMatchingRules
FROM "waf_logs"
CROSS JOIN UNNEST(rulegroupList) AS t(t)
WHERE action <> 'BLOCK' AND cardinality(t.nonTerminatingMatchingRules) > 0
GROUP BY t.nonTerminatingMatchingRules, action, httpsourceid, httprequest.clientip,
  t.rulegroupid
ORDER BY "count" DESC
Limit 50
```

Example : agrupar todas las reglas personalizadas contadas por el número de veces que coinciden

En la siguiente consulta se agrupan todas las reglas personalizadas contadas según el número de veces que coinciden.

```
SELECT
  count(*) AS count,
  httpsourceid,
  httprequest.clientip,
  t.ruleid,
  t.action
FROM "waf_logs"
CROSS JOIN UNNEST(nonterminatingmatchingrules) AS t(t)
WHERE action <> 'BLOCK' AND cardinality(nonTerminatingMatchingRules) > 0
GROUP BY t.ruleid, t.action, httpsourceid, httprequest.clientip
ORDER BY "count" DESC
Limit 50
```

Para obtener información sobre las ubicaciones de registro de las reglas personalizadas y los grupos de reglas administrados, consulte [Supervisión y ajuste](#) en la Guía para desarrolladores de AWS WAF.

## Uso de fecha y hora

Example : devolver el campo de marca de tiempo en formato ISO 8601 legible por humanos

La siguiente consulta utiliza las funciones `from_unixtime` y `to_iso8601` para devolver el campo `timestamp` en formato ISO 8601 legible por humanos (por ejemplo, `2019-12-13T23:40:12.000Z` en lugar de `1576280412771`). La consulta devuelve también el nombre de origen HTTP, el ID de origen y la solicitud.

```
SELECT to_iso8601(from_unixtime(timestamp / 1000)) as time_ISO_8601,  
       httpsourcename,  
       httpsourceid,  
       httprequest  
FROM waf_logs  
LIMIT 10;
```

Example : devolver registros de las últimas 24 horas

La siguiente consulta utiliza un filtro en el cláusula `WHERE` para devolver el nombre de origen HTTP, el ID de origen HTTP y los campos de solicitud HTTP para los registros de las últimas 24 horas.

```
SELECT to_iso8601(from_unixtime(timestamp/1000)) AS time_ISO_8601,  
       httpsourcename,  
       httpsourceid,  
       httprequest  
FROM waf_logs  
WHERE from_unixtime(timestamp/1000) > now() - interval '1' day  
LIMIT 10;
```

Example : devolver registros para un intervalo de fechas y una dirección IP especificados

En la siguiente consulta, se enumeran los registros de un intervalo de fechas especificado para una dirección IP de cliente especificada.

```
SELECT *  
FROM waf_logs  
WHERE httprequest.clientip='53.21.198.66' AND "date" >= '2021/03/01' AND "date" <  
      '2021/03/31'
```

Example : contar el número de direcciones IP en intervalos de cinco minutos para un intervalo de fechas especificado

La siguiente consulta cuenta el número de direcciones IP en intervalos de cinco minutos para un intervalo de fechas determinado.

```
WITH test_dataset AS
  (SELECT
    format_datetime(from_unixtime((timestamp/1000) -
      ((minute(from_unixtime(timestamp / 1000))%5) * 60)), 'yyyy-MM-dd HH:mm') AS
    five_minutes_ts,
    "httprequest"."clientip"
  FROM waf_logs
  WHERE "date" >= '2021/03/01' AND "date" < '2021/03/31')
SELECT five_minutes_ts,"clientip",count(*) ip_count
FROM test_dataset
GROUP BY five_minutes_ts,"clientip"
```

Example : contar el número de IP X-Forwarded-For en los últimos 10 días

En la siguiente consulta se filtran los encabezados de la solicitud y se cuenta el número de IP de X-Forwarded-For de los últimos 10 días.

```
WITH test_dataset AS
  (SELECT header
  FROM waf_logs
  CROSS JOIN UNNEST (httprequest.headers) AS t(header)
  WHERE from_unixtime("timestamp"/1000) > now() - interval '10' DAY)
SELECT header.value AS ip,
  count(*) AS COUNT
FROM test_dataset
WHERE header.name='X-Forwarded-For'
GROUP BY header.value
ORDER BY COUNT DESC
```

Para obtener más información sobre las funciones de fecha y hora, consulte [Funciones y operadores de fecha y hora](#) en la documentación de Trino.

## Trabajar con solicitudes y direcciones bloqueadas

Example : extraer las 100 direcciones IP principales bloqueadas por un tipo de regla especificado

La siguiente consulta extrae y cuenta las 100 direcciones IP principales que han sido bloqueadas por la regla de finalización RATE\_BASED durante el intervalo de tiempo especificado.

```
SELECT COUNT(httpRequest.clientIp) as count,
httpRequest.clientIp
FROM waf_logs
WHERE terminatingruletype='RATE_BASED' AND action='BLOCK' and "date" >= '2021/03/01'
AND "date" < '2021/03/31'
GROUP BY httpRequest.clientIp
ORDER BY count DESC
LIMIT 100
```

Example : contar el número de veces que se bloqueó una solicitud de un país especificado

La siguiente consulta cuenta el número de veces que la solicitud ha llegado de una dirección IP que pertenece a Irlanda (IE) y ha sido bloqueada por la regla de terminación RATE\_BASED.

```
SELECT
COUNT(httpRequest.country) as count,
httpRequest.country
FROM waf_logs
WHERE
terminatingruletype='RATE_BASED' AND
httpRequest.country='IE'
GROUP BY httpRequest.country
ORDER BY count
LIMIT 100;
```

Example : contar el número de veces que se bloqueó una solicitud, agrupando por atributos específicos

La siguiente consulta cuenta el número de veces que la solicitud se ha bloqueado, con los resultados agrupados por WebACL, RuleId, ClientIP y URI de la solicitud HTTP.

```
SELECT
COUNT(*) AS count,
webaclid,
```

```

terminatingruleid,
httprequest.clientip,
httprequest.uri
FROM waf_logs
WHERE action='BLOCK'
GROUP BY webaclid, terminatingruleid, httprequest.clientip, httprequest.uri
ORDER BY count DESC
LIMIT 100;

```

Example : contar el número de veces que se encontraron coincidencias con un ID de regla de terminación específico

La siguiente consulta cuenta el número de veces que se han encontrado coincidencias con un ID de regla de terminación específico (WHERE terminatingruleid= 'e9dd190d-7a43-4c06-bcea-409613d9506e '). La consulta agrupa después los resultados por WebACL, Action, ClientIP y URI de solicitud HTTP.

```

SELECT
COUNT(*) AS count,
webaclid,
action,
httprequest.clientip,
httprequest.uri
FROM waf_logs
WHERE terminatingruleid='e9dd190d-7a43-4c06-bcea-409613d9506e'
GROUP BY webaclid, action, httprequest.clientip, httprequest.uri
ORDER BY count DESC
LIMIT 100;

```

Example : recuperar las 100 direcciones IP principales bloqueadas durante un intervalo de fechas especificado

La siguiente consulta extrae las 100 direcciones IP principales que se han bloqueado durante un intervalo de tiempo especificado. La consulta muestra también el número de veces que se han bloqueado las direcciones IP.

```

SELECT "httprequest"."clientip", "count"(*) "ipcount", "httprequest"."country"
FROM waf_logs
WHERE "action" = 'BLOCK' and "date" >= '2021/03/01'
AND "date" < '2021/03/31'
GROUP BY "httprequest"."clientip", "httprequest"."country"

```

```
ORDER BY "ipcount" DESC limit 100
```

Para obtener más información sobre la consulta de registros de Amazon S3, consulte los siguientes temas:

- [¿Cómo analizo los registros de acceso al servidor de Amazon S3 mediante Athena?](#) en el Centro de conocimientos de AWS
- [Consulta de los registros de acceso de Amazon S3 para solicitudes mediante Amazon Athena](#) en la Guía del usuario de Amazon Simple Storage Service
- [Uso de AWS CloudTrail para identificar solicitudes de Amazon S3](#) en la Guía del usuario de Amazon Simple Storage Service

## Consulta de registros de servidor web almacenados en Simple Storage Service (Amazon S3)

Puede utilizar Athena para consultar los registros del servidor web almacenados en Amazon S3. En los temas de esta sección se muestra cómo crear tablas en Athena para consultar registros de servidor web en distintos formatos.

### Temas

- [Consulta de registros de Apache almacenados en Amazon S3](#)
- [Consulta de los registros de Internet Information Services \(IIS\) almacenados en Simple Storage Service \(Amazon S3\)](#)

## Consulta de registros de Apache almacenados en Amazon S3

Puede utilizar Amazon Athena para consultar [archivos de registro del servidor HTTP Apache](#) almacenados en su cuenta de Amazon S3. En este tema, se muestra cómo crear esquemas de tabla para consultar archivos de [registro de acceso](#) de Apache en el formato de registro común.

Los campos en el formato de registro común incluyen la dirección IP del cliente, el ID del cliente, el ID de usuario, la marca de hora de la solicitud recibida, el texto de la solicitud del cliente, el código de estado del servidor y el tamaño del objeto devuelto al cliente.

En el siguiente ejemplo de datos se muestra el formato de registro común de Apache.

```
198.51.100.7 - Li [10/Oct/2019:13:55:36 -0700] "GET /logo.gif HTTP/1.0" 200 232
```

```

198.51.100.14 - Jorge [24/Nov/2019:10:49:52 -0700] "GET /index.html HTTP/1.1" 200 2165
198.51.100.22 - Mateo [27/Dec/2019:11:38:12 -0700] "GET /about.html HTTP/1.1" 200 1287
198.51.100.9 - Nikki [11/Jan/2020:11:40:11 -0700] "GET /image.png HTTP/1.1" 404 230
198.51.100.2 - Ana [15/Feb/2019:10:12:22 -0700] "GET /favicon.ico HTTP/1.1" 404 30
198.51.100.13 - Saanvi [14/Mar/2019:11:40:33 -0700] "GET /intro.html HTTP/1.1" 200 1608
198.51.100.11 - Xiulan [22/Apr/2019:10:51:34 -0700] "GET /group/index.html HTTP/1.1"
200 1344

```

## Creación de una tabla en Athena para los registros de Apache

Antes de poder consultar los registros de Apache almacenados en Amazon S3, debe crear un esquema de tabla para Athena de modo que pueda leer los datos de registro. Para crear una tabla de Athena para los registros de Apache, puede usar [El SerDe de Grok](#). Para obtener más información sobre el uso del SerDe de Grok, consulte [Escritura de clasificadores personalizados de Grok](#) en la Guía para desarrolladores de AWS Glue.

### Cómo crear una tabla en Athena para los registros del servidor web de Apache

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Pegue la siguiente instrucción DDL en el Editor de consultas de Athena. Modifique los valores de LOCATION 's3://DOC-EXAMPLE-BUCKET/*apache-log-folder*/' para que apunten a los registros de Apache en Amazon S3.

```

CREATE EXTERNAL TABLE apache_logs (
  client_ip string,
  client_id string,
  user_id string,
  request_received_time string,
  client_request string,
  server_status string,
  returned_obj_size string
)
ROW FORMAT SERDE
  'com.amazonaws.glue.serde.GrokSerDe'
WITH SERDEPROPERTIES (
  'input.format'='^[IPV4:client_ip] %{DATA:client_id} %{USERNAME:user_id}
  %{GREEDYDATA:request_received_time} %{QUOTEDSTRING:client_request}
  %{DATA:server_status} %{DATA: returned_obj_size}$'
)
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT

```

```
'org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
's3://DOC-EXAMPLE-BUCKET/apache-log-folder';
```

- Ejecute la consulta en la consola de Athena para registrar la tabla `apache_logs`. Cuando se complete la consulta, los registros estarán listos para su consulta desde Athena.

## Ejemplo de consultas Select para registros de Apache

### Example – Filtrado de errores 404

En la siguiente consulta de ejemplo, se selecciona la hora recibida de la solicitud, el texto de la solicitud del cliente y el código de estado del servidor en la tabla `apache_logs`. La cláusula `WHERE` filtra el código de estado HTTP 404 (página no encontrada).

```
SELECT request_received_time, client_request, server_status
FROM apache_logs
WHERE server_status = '404'
```

En la siguiente imagen se muestran los resultados de la consulta en el Editor de consultas de Athena.

Results		
<b>request_received_time</b>	<b>client_request</b>	<b>server_status</b>
1 [11/Jan/2020:11:40:11 -0700]	GET /image.png HTTP/1.1	404
2 [15/Feb/2019:10:12:22 -0700]	GET /favicon.ico HTTP/1.1	404

### Example – Filtrado de solicitudes correctas

En la siguiente consulta de ejemplo, se selecciona el ID de usuario, la hora recibida de la solicitud, el texto de la solicitud del cliente y el código de estado del servidor en la tabla `apache_logs`. La cláusula `WHERE` filtra el código de estado HTTP 200 (satisfactorio).

```
SELECT user_id, request_received_time, client_request, server_status
FROM apache_logs
WHERE server_status = '200'
```



En la siguiente imagen se muestran los resultados de la consulta en el Editor de consultas de Athena.

Results				
	user_id	request_received_time	client_request	server_status
1	Li	[10/Oct/2019:13:55:36 -0700]	GET /logo.gif HTTP/1.0	200
2	Jorge	[24/Nov/2019:10:49:52 -0700]	GET /index.html HTTP/1.1	200
3	Mateo	[27/Dec/2019:11:38:12 -0700]	GET /about.html HTTP/1.1	200
4	Saanvi	[14/Mar/2019:11:40:33 -0700]	GET /intro.html HTTP/1.1	200
5	Xiulan	[22/Apr/2019:10:51:34 -0700]	GET /group/index.html HTTP/1.1	200

### Example – Filtrado por marca de tiempo

En el siguiente ejemplo, se realiza una consulta de registros cuya hora de recepción de la solicitud sea posterior a la marca de tiempo especificada.

```
SELECT * FROM apache_logs WHERE request_received_time > 10/Oct/2023:00:00:00
```

## Consulta de los registros de Internet Information Services (IIS) almacenados en Simple Storage Service (Amazon S3)

Puede utilizar Amazon Athena para consultar los registros del servidor web de Microsoft Internet Information Services (IIS) almacenados en su cuenta de Amazon S3. Aunque IIS utiliza [diferentes](#) formatos de archivo de registro, en este tema se muestra cómo crear esquemas de tabla para consultar los registros de formato de archivo de registro extendido de W3C y IIS desde Athena.

Dado que los formatos de archivo de registro extendido de W3C y IIS utilizan delimitadores de caracteres únicos (espacios y comas, respectivamente) y no tienen valores entre comillas, puede utilizar [LazySimpleSerDe](#) para crear tablas de Athena para ellos.

### Formato de archivo de registro extendido de W3C

El formato de datos de archivo de registro [extendido de W3C](#) tiene campos separados por espacios. Un administrador del servidor web, que elige los campos de registro que desea incluir, determina los campos que aparecen en los registros extendidos de W3C. En el siguiente ejemplo, los datos de registro tienen los campos `date`, `time`, `c-ip`, `s-ip`, `cs-method`, `cs-uri-stem`, `sc-status`, `sc-bytes`, `cs-bytes`, `time-taken` y `cs-version`.

```
2020-01-19 22:48:39 203.0.113.5 198.51.100.2 GET /default.html 200 540 524 157 HTTP/1.0
2020-01-19 22:49:40 203.0.113.10 198.51.100.12 GET /index.html 200 420 324 164 HTTP/1.0
2020-01-19 22:50:12 203.0.113.12 198.51.100.4 GET /image.gif 200 324 320 358 HTTP/1.0
2020-01-19 22:51:44 203.0.113.15 198.51.100.16 GET /faq.html 200 330 324 288 HTTP/1.0
```

## Creación de una tabla en Athena para los registros extendidos de W3C

Antes de poder consultar los registros extendidos de W3C, debe crear un esquema de tabla de modo que Athena pueda leer los datos de registro.

Para crear una tabla en Athena para registros extendidos de W3C

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Pegue una instrucción DDL como la siguiente en la consola de Athena, teniendo en cuenta los puntos a continuación:
  - a. Agregue o elimine las columnas del ejemplo para que se correspondan con los campos de los registros que desea consultar.
  - b. Los nombres de columna en el formato de archivo de registro extendido de W3C contienen guiones (-). Sin embargo, de acuerdo con las [convenciones de nomenclatura de Athena](#), la instrucción CREATE TABLE de ejemplo los reemplaza con guiones bajos (\_).
  - c. Para especificar el delimitador de espacio, utilice FIELDS TERMINATED BY ' '.
  - d. Modifique los valores de LOCATION 's3://DOC-EXAMPLE-BUCKET/*w3c-log-folder*/' para que apunten a los registros extendidos de W3C en Amazon S3.

```
CREATE EXTERNAL TABLE `iis_w3c_logs`(  
  date_col string,  
  time_col string,  
  c_ip string,  
  s_ip string,  
  cs_method string,  
  cs_uri_stem string,  
  sc_status string,  
  sc_bytes string,  
  cs_bytes string,  
  time_taken string,  
  cs_version string  
)  
ROW FORMAT DELIMITED
```

```

FIELDS TERMINATED BY ' '
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://DOC-EXAMPLE-BUCKET/w3c-log-folder/'

```

- Ejecute la consulta en la consola de Athena para registrar la tabla `iis_w3c_logs`. Cuando se complete la consulta, los registros estarán listos para su consulta desde Athena.

### Ejemplo de consulta Select de registro extendido de W3C

En la siguiente consulta de ejemplo se selecciona la fecha, la hora, el destino de la solicitud y el tiempo que demora la solicitud de la tabla `iis_w3c_logs`. La cláusula `WHERE` filtra los casos en los que el método HTTP es `GET` y el código de estado HTTP es `200` (satisfactorio).

```

SELECT date_col, time_col, cs_uri_stem, time_taken
FROM iis_w3c_logs
WHERE cs_method = 'GET' AND sc_status = '200'

```

En la siguiente imagen se muestran los resultados de la consulta en el Editor de consultas de Athena.

Results				
	▲ date_col ▼	time_col ▼	cs_uri_stem ▼	time_taken ▼
1	2020-01-19	22:48:39	/default.html	157
2	2020-01-19	22:49:40	/index.html	164
3	2020-01-19	22:50:12	/image.gif	358
4	2020-01-19	22:51:44	/faq.html	288

### Combinación de los campos de fecha y hora

Los campos delimitados por espacios `date` y `time` son entradas separadas en los datos de origen del registro, pero puede combinarlas en una marca temporal, si lo desea. Utilice las funciones [concat\(\)](#) y [date\\_parse\(\)](#) en una consulta [SELECT](#) o [CREATE TABLE AS SELECT](#) para concatenar

y convertir las columnas de fecha y hora en formato de marca temporal. En el siguiente ejemplo se utiliza una consulta CTAS para crear una nueva tabla con una columna `derived_timestamp`.

```
CREATE TABLE iis_w3c_logs_w_timestamp AS
SELECT
  date_parse(concat(date_col, ' ', time_col), '%Y-%m-%d %H:%i:%s') as derived_timestamp,
  c_ip,
  s_ip,
  cs_method,
  cs_uri_stem,
  sc_status,
  sc_bytes,
  cs_bytes,
  time_taken,
  cs_version
FROM iis_w3c_logs
```

Una vez creada la tabla, puede consultar la nueva columna de marca temporal directamente, como en el siguiente ejemplo.

```
SELECT derived_timestamp, cs_uri_stem, time_taken
FROM iis_w3c_logs_w_timestamp
WHERE cs_method = 'GET' AND sc_status = '200'
```

En la siguiente imagen, se muestran los resultados de la consulta.

Results			
	▲ <code>derived_timestamp</code> ▼	<code>cs_uri_stem</code> ▼	<code>time_taken</code> ▼
1	2020-01-19 22:48:39.000	/default.html	157
2	2020-01-19 22:49:40.000	/index.html	164
3	2020-01-19 22:50:12.000	/image.gif	358
4	2020-01-19 22:51:44.000	/faq.html	288

## Formato de archivo de registro de IIS

A diferencia del formato extendido de W3C, el [formato de archivo de registro de IIS](#) tiene un conjunto fijo de campos e incluye una coma como delimitador. El LazySimpleSerDe trata la coma como el delimitador y el espacio después de la coma como el inicio del siguiente campo.

En el siguiente ejemplo, se muestran datos de ejemplo en el formato de archivo de registro de IIS.

```
203.0.113.15, -, 2020-02-24, 22:48:38, W3SVC2, SERVER5, 198.51.100.4, 254, 501, 488,
200, 0, GET, /index.htm, -,
203.0.113.4, -, 2020-02-24, 22:48:39, W3SVC2, SERVER6, 198.51.100.6, 147, 411, 388,
200, 0, GET, /about.html, -,
203.0.113.11, -, 2020-02-24, 22:48:40, W3SVC2, SERVER7, 198.51.100.18, 170, 531, 468,
200, 0, GET, /image.png, -,
203.0.113.8, -, 2020-02-24, 22:48:41, W3SVC2, SERVER8, 198.51.100.14, 125, 711, 868,
200, 0, GET, /intro.htm, -,
```

## Creación de una tabla en Athena para los archivos de registro de IIS

Para consultar los registros de formato de archivo de registro de IIS almacenados en Amazon S3, debe crear un esquema de tabla de modo que Athena pueda leer los datos de registro.

Para crear una tabla en Athena para registros de formato de archivo de registro de IIS

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Pegue la siguiente instrucción DDL en la consola de Athena, teniendo en cuenta los puntos a continuación:
  - a. Para especificar el delimitador de coma, utilice `FIELDS TERMINATED BY ','`.
  - b. Modifique los valores de `LOCATION 's3://DOC-EXAMPLE-BUCKET/iis-log-file-folder'` de forma que se dirijan a los archivos de registro del formato de registro de IIS en Amazon S3.

```
CREATE EXTERNAL TABLE `iis_format_logs`(  
  client_ip_address string,  
  user_name string,  
  request_date string,  
  request_time string,  
  service_and_instance string,  
  server_name string,  
  server_ip_address string,  
  time_taken_millisec string,  
  client_bytes_sent string,  
  server_bytes_sent string,  
  service_status_code string,  
  windows_status_code string,
```

```

request_type string,
target_of_operation string,
script_parameters string
)
ROW FORMAT DELIMITED
  FIELDS TERMINATED BY ','
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET/iis-log-file-folder/'

```

3. Ejecute la consulta en la consola de Athena para registrar la tabla `iis_format_logs`. Cuando se complete la consulta, los registros estarán listos para su consulta desde Athena.

### Ejemplo de consulta Select de formato de registro de IIS

En la siguiente consulta de ejemplo se selecciona la fecha, la hora, el destino de la solicitud y el tiempo que demora en milisegundos de la tabla `iis_format_logs`. La cláusula `WHERE` filtra los casos en los que el tipo de solicitud es `GET` y el código de estado HTTP es `200` (satisfactorio). En la consulta, tenga en cuenta que los espacios iniciales en `' GET'` y `' 200'` son necesarios para que la consulta sea correcta.

```

SELECT request_date, request_time, target_of_operation, time_taken_millisec
FROM iis_format_logs
WHERE request_type = ' GET' AND service_status_code = ' 200'

```

En la siguiente imagen se muestran resultados de la consulta de los datos de muestra.

Results				
	request_date ▼	request_time ▼	target_of_operation ▼	time_taken_millisec ▼
1	2020-02-24	22:48:38	/index.htm	254
2	2020-02-24	22:48:39	/about.html	147
3	2020-02-24	22:48:40	/image.png	170
4	2020-02-24	22:48:41	/intro.htm	125

## Formato del archivo de registro NCSA

IIS también utiliza el formato de [registro NCSA](#), que tiene un número fijo de campos en formato de texto ASCII separados por espacios. La estructura es similar al formato de registro común utilizado para los registros de acceso de Apache. Los campos en el formato de datos de registro común de NCSA incluyen la dirección IP del cliente, el ID del cliente (no se utiliza normalmente), el dominio\ ID de usuario, la marca temporal de la solicitud recibida, el texto de la solicitud del cliente, el código de estado del servidor y el tamaño del objeto devuelto al cliente.

En el siguiente ejemplo, se muestran los datos en el formato de registro común de NCSA documentado para IIS.

```
198.51.100.7 - ExampleCorp\Li [10/Oct/2019:13:55:36 -0700] "GET /logo.gif HTTP/1.0" 200
 232
198.51.100.14 - AnyCompany\Jorge [24/Nov/2019:10:49:52 -0700] "GET /index.html
HTTP/1.1" 200 2165
198.51.100.22 - ExampleCorp\Mateo [27/Dec/2019:11:38:12 -0700] "GET /about.html
HTTP/1.1" 200 1287
198.51.100.9 - AnyCompany\Nikki [11/Jan/2020:11:40:11 -0700] "GET /image.png HTTP/1.1"
404 230
198.51.100.2 - ExampleCorp\Ana [15/Feb/2019:10:12:22 -0700] "GET /favicon.ico HTTP/1.1"
404 30
198.51.100.13 - AnyCompany\Saanvi [14/Mar/2019:11:40:33 -0700] "GET /intro.html
HTTP/1.1" 200 1608
198.51.100.11 - ExampleCorp\Xiulan [22/Apr/2019:10:51:34 -0700] "GET /group/index.html
HTTP/1.1" 200 1344
```

## Creación de una tabla en Athena para los registros NCSA de IIS

Para la instrucción `CREATE TABLE`, puede utilizar [El SerDe de Grok](#) y un patrón grok similar al de los [registros del servidor web de Apache](#). A diferencia de los registros de Apache, el patrón grok utiliza `%{DATA:user_id}` para el tercer campo en lugar de `%{USERNAME:user_id}` para dar cuenta de la presencia de la barra invertida en `domain\user_id`. Para obtener más información sobre el uso del SerDe de Grok, consulte [Escritura de clasificadores personalizados de Grok](#) en la Guía para desarrolladores de AWS Glue.

Para crear una tabla en Athena para los registros del servidor web NCSA de IIS

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.

2. Pegue la siguiente instrucción DDL en el Editor de consultas de Athena. Modifique los valores de LOCATION 's3://DOC-EXAMPLE-BUCKET/*iis-ncsa-logs*/' para apuntar a los registros NCSA de IIS en Amazon S3.

```
CREATE EXTERNAL TABLE iis_ncsa_logs(  
  client_ip string,  
  client_id string,  
  user_id string,  
  request_received_time string,  
  client_request string,  
  server_status string,  
  returned_obj_size string  
)  
ROW FORMAT SERDE  
  'com.amazonaws.glue.serde.GrokSerDe'  
WITH SERDEPROPERTIES (  
  'input.format'='^%{IPV4:client_ip} %{DATA:client_id} %{DATA:user_id}  
  %{GREEDYDATA:request_received_time} %{QUOTEDSTRING:client_request}  
  %{DATA:server_status} %{DATA: returned_obj_size}$'  
)  
STORED AS INPUTFORMAT  
  'org.apache.hadoop.mapred.TextInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat'  
LOCATION  
  's3://DOC-EXAMPLE-BUCKET/iis-ncsa-logs/';
```

3. Ejecute la consulta en la consola de Athena para registrar la tabla `iis_ncsa_logs`. Cuando se complete la consulta, los registros estarán listos para su consulta desde Athena.

## Ejemplo de consultas Select para registros NCSA de IIS

### Example – Filtrado de errores 404

En la siguiente consulta de ejemplo, se selecciona la hora recibida de la solicitud, el texto de la solicitud del cliente y el código de estado del servidor en la tabla `iis_ncsa_logs`. La cláusula WHERE filtra el código de estado HTTP 404 (página no encontrada).

```
SELECT request_received_time, client_request, server_status  
FROM iis_ncsa_logs  
WHERE server_status = '404'
```



En la siguiente imagen se muestran los resultados de la consulta en el Editor de consultas de Athena.



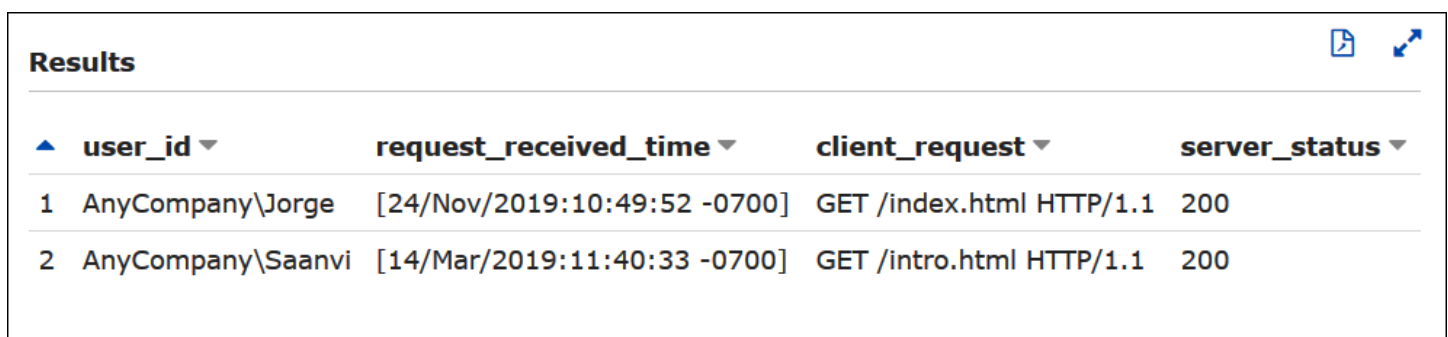
	<b>request_received_time</b>	<b>client_request</b>	<b>server_status</b>
1	[11/Jan/2020:11:40:11 -0700]	GET /image.png HTTP/1.1	404
2	[15/Feb/2019:10:12:22 -0700]	GET /favicon.ico HTTP/1.1	404

Example – Filtrar solicitudes correctas de un dominio en particular

En la siguiente consulta de ejemplo, se selecciona el ID de usuario, la hora recibida de la solicitud, el texto de la solicitud del cliente y el código de estado del servidor en la tabla `iis_ncsa_logs`. La cláusula `WHERE` filtra solicitudes con código de estado HTTP `200` (satisfactorio) de los usuarios en el dominio `AnyCompany`.

```
SELECT user_id, request_received_time, client_request, server_status
FROM iis_ncsa_logs
WHERE server_status = '200' AND user_id LIKE 'AnyCompany%'
```

En la imagen siguiente se muestran los resultados de la consulta en el Editor de consultas de Athena.



	<b>user_id</b>	<b>request_received_time</b>	<b>client_request</b>	<b>server_status</b>
1	AnyCompany\Jorge	[24/Nov/2019:10:49:52 -0700]	GET /index.html HTTP/1.1	200
2	AnyCompany\Saanvi	[14/Mar/2019:11:40:33 -0700]	GET /intro.html HTTP/1.1	200

## Uso de las transacciones ACID de Athena

El término “transacciones ACID” se refiere a un conjunto de propiedades ([atomicidad](#), [consistencia](#), [aislamiento](#) y [durabilidad](#)) que garantizan la integridad de los datos en las transacciones de bases

de datos. Las transacciones ACID permiten a varios usuarios agregar y eliminar objetos de Amazon S3 de forma atómica, simultánea y confiable, al mismo tiempo que aíslan las consultas existentes y mantienen la coherencia de lectura de las consultas en el lago de datos. Las transacciones ACID de Athena agregan compatibilidad con una única tabla para las operaciones de inserción, eliminación, actualización y viaje en el tiempo al lenguaje de manipulación de datos (DML) SQL de Athena. Las transacciones ACID de Athena permiten que, junto con varios usuarios simultáneos, pueda hacer modificaciones confiables por fila en los datos de Amazon S3. Las transacciones de Athena administran automáticamente la semántica de bloqueo y la coordinación y no requieren ninguna solución de bloqueo de registros personalizada.

Las transacciones ACID de Athena y la conocida sintaxis SQL simplifican las actualizaciones de los datos empresariales y normativos. Por ejemplo, para responder a una solicitud de borrado de datos, puede llevar a cabo una operación DELETE de SQL. Para hacer correcciones manuales de registros, puede utilizar una sola instrucción UPDATE. Para recuperar los datos eliminados recientemente, puede emitir consultas de viaje en el tiempo mediante una instrucción SELECT.

Al estar creadas en formatos de tabla compartida, las transacciones ACID de Athena son compatibles con otros servicios y motores, como [Amazon EMR](#) y [Apache Spark](#), que también admiten los formatos de tabla compartida.

Las transacciones de Athena están disponibles a través de la consola de Athena, las operaciones de la API y los controladores ODBC y JDBC.

## Temas

- [Consulta de tablas de Linux Foundation Delta Lake](#)
- [Uso de Athena para consultar conjuntos de datos de Apache Hudi](#)
- [Uso de tablas de Apache Iceberg](#)

## Consulta de tablas de Linux Foundation Delta Lake

Linux Foundation [Delta Lake](#) es un formato de tabla para el análisis de macrodatos. Puede usar Amazon Athena para leer directamente las tablas de Delta Lake almacenadas en Amazon S3 sin tener que generar archivos de manifiesto ni ejecutar la instrucción MSCK REPAIR.

El formato de Delta Lake almacena los valores mínimo y máximo por columna de cada archivo de datos. La implementación de Athena utiliza esta información para permitir omitir archivos en los predicados a fin de evitar que se tengan en cuenta los archivos no deseados.

## Consideraciones y limitaciones

La compatibilidad de Delta Lake Athena tiene las siguientes consideraciones y limitaciones:

- Solo tablas con catálogo de AWS Glue: la compatibilidad nativa de Delta Lake solo se admite a través de tablas registradas con AWS Glue. Si tiene una tabla de Delta Lake registrada en otro metaalmacén, puede conservarla y tratarla como su metaalmacén principal. Como los metadatos de Delta Lake se almacenan en el sistema de archivos (por ejemplo, en Amazon S3) y no en el metaalmacén, Athena solo requiere la propiedad de ubicación de AWS Glue para leer las tablas de Delta Lake.
- Solo motor V3: las consultas de Delta Lake solo se admiten en la versión 3 del motor Athena. Debe asegurarse de que el grupo de trabajo que cree esté configurado para utilizar la versión 3 del motor Athena.
- Versión del lector de Delta Lake: se admite hasta la versión 3 del protocolo del lector de Delta Lake.
- Asignación de columnas y marca de tiempo sin zona horaria (Ntz): se admiten la [asignación de columnas de Delta](#), que permite que las columnas de la tabla Delta y las columnas del archivo Parquet subyacente utilicen nombres diferentes, y la marca de tiempo sin zona horaria ([Ntz](#)).
- No hay compatibilidad con viajes en el tiempo: no hay compatibilidad con consultas que utilicen las capacidades de viaje en el tiempo de Delta Lake.
- Solo lectura: las instrucciones DML de escritura como UPDATE, INSERT o DELETE no se admiten.
- Soporte de Lake Formation: la integración de Lake Formation se encuentra disponible para las tablas de Delta Lake con su esquema sincronizado con AWS Glue. A fin de obtener más información, consulte [Uso de AWS Lake Formation con Amazon Athena](#) y [Configuración de permisos para una tabla de Delta Lake](#) en la Guía para desarrolladores de AWS Lake Formation.
- Compatibilidad con DDL limitada: se admiten las siguientes instrucciones DDL: CREATE EXTERNAL TABLE, SHOW COLUMNS, SHOW TBLPROPERTIES, SHOW PARTITIONS, SHOW CREATE TABLE y DESCRIBE. Para obtener información sobre el uso de la instrucción CREATE EXTERNAL TABLE, consulte la sección [Introducción](#).
- No se admite la omisión de objetos de S3 Glacier: si los objetos de la tabla de Linux Foundation Delta Lake se encuentran en una clase de almacenamiento de Amazon S3 Glacier, establecer la propiedad de la tabla `read_restored_glacier_objects` en `false` no tendrá ningún efecto.

Suponga, por ejemplo, que ejecuta el siguiente comando:

```
ALTER TABLE table_name SET TBLPROPERTIES ('read_restored_glacier_objects' = 'false')
```

En el caso de las tablas de Iceberg y Delta Lake, el comando produce el error Clave de propiedad de tabla no compatible: `read_restored_glacier_objects`. En el caso de las tablas de Hudi, el comando `ALTER TABLE` no produce ningún error, pero los objetos de Amazon S3 Glacier siguen sin omitirse. Al ejecutar consultas `SELECT` después del comando `ALTER TABLE`, se siguen devolviendo todos los objetos.

## Tipos de datos de columnas sin particiones compatibles

En el caso de las columnas sin particiones, se admiten todos los tipos de datos que admite Athena excepto `CHAR` (`CHAR` no se admite en el propio protocolo de Delta Lake). Los tipos de datos admitidos incluyen:

```
boolean
tinyint
smallint
integer
bigint
double
float
decimal
varchar
string
binary
date
timestamp
array
map
struct
```

## Tipos de datos de columnas de particiones compatibles

Para las columnas de particiones, Athena admite tablas con los siguientes tipos de datos:

```
boolean
integer
smallint
tinyint
```

```
bigint
decimal
float
double
date
timestamp
varchar
```

Para más información sobre los tipos de datos en Athena, consulte [Tipos de datos en Amazon Athena](#).

## Introducción

Para poder consultarla, su tabla de Delta Lake debe existir en AWS Glue. Si la tabla está en Amazon S3, pero no en AWS Glue, ejecute una instrucción CREATE EXTERNAL TABLE con la siguiente sintaxis. Si la tabla ya existe en AWS Glue (por ejemplo, porque está usando Apache Spark u otro motor con AWS Glue), puede omitir este paso.

```
CREATE EXTERNAL TABLE
  [db_name.]table_name
  LOCATION 's3://DOC-EXAMPLE-BUCKET/your-folder/'
  TBLPROPERTIES ('table_type' = 'DELTA')
```

Tenga en cuenta la omisión de las definiciones de columnas, la biblioteca SerDe y otras propiedades de tablas. A diferencia de las tablas de Hive tradicionales, los metadatos de las tablas de Delta Lake se deducen del registro de transacciones de Delta Lake y se sincronizan directamente con AWS Glue.

### Note

En el caso de las tablas de Delta Lake, no se permiten instrucciones CREATE TABLE que incluyan más que las propiedades LOCATION y table\_type.

## Lectura de las tablas de Delta Lake

Para consultar una tabla de Delta Lake, utilice la sintaxis SELECT SQL estándar:

```
[ WITH with_query [, ...] ]SELECT [ ALL | DISTINCT ] select_expression [, ...]
```

```
[ FROM from_item [, ...] ]
[ WHERE condition ]
[ GROUP BY [ ALL | DISTINCT ] grouping_element [, ...] ]
[ HAVING condition ]
[ { UNION | INTERSECT | EXCEPT } [ ALL | DISTINCT ] select ]
[ ORDER BY expression [ ASC | DESC ] [ NULLS FIRST | NULLS LAST] [, ...] ]
[ OFFSET count [ ROW | ROWS ] ]
[ LIMIT [ count | ALL ] ]
```

Para obtener más información acerca de la sintaxis SELECT, consulte [SELECT](#) en la documentación de Athena.

El formato de Delta Lake almacena los valores mínimo y máximo por columna de cada archivo de datos. Athena utiliza esta información para permitir omitir archivos en los predicados a fin de evitar que se tengan en cuenta los archivos innecesarios.

## Sincronización de metadatos de Delta Lake

Athena sincroniza los metadatos de la tabla, incluidos el esquema, las columnas de partición y las propiedades de la tabla, con AWS Glue si usa Athena para crear su tabla de Delta Lake. A medida que pasa el tiempo, estos metadatos pueden perder su sincronización con los metadatos de la tabla subyacente en el registro de transacciones. Para mantener su tabla actualizada, puede elegir una de las siguientes opciones:

- Utilizar el rastreador de AWS Glue para tablas de Delta Lake. Para obtener más información, consulte [Introducing native Delta Lake table support with AWS Glue crawlers](#) en el Blog de macrodatos de AWS y [Programación de un rastreador de AWS Glue](#) en la Guía para desarrolladores de AWS Glue.
- Eliminar y recrear la tabla en Athena.
- Utilizar el SDK, la CLI o la consola de AWS Glue para actualizar de forma manual el esquema en AWS Glue.

Tenga en cuenta que las siguientes características requieren que su esquema de AWS Glue tenga siempre el mismo esquema que el registro de transacciones:

- Lake Formation
- Vistas
- Filtros de filas y columnas

Si su flujo de trabajo no requiere ninguna de estas funciones y prefiere no mantener esta compatibilidad, puede utilizar el DDL `CREATE TABLE` en Athena y, a continuación, agregar la ruta de Amazon S3 como parámetro `Serde` en AWS Glue.

Para crear una tabla de Delta Lake con las consolas de Athena y AWS Glue

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. En el editor de consultas de Athena, utilice el siguiente DDL para crear la tabla de Delta Lake. Tenga en cuenta que cuando utilice este método, el valor de `TBLPROPERTIES` debe ser `'spark.sql.sources.provider' = 'delta'` y no `'table_type' = 'delta'`.

Tenga en cuenta que este mismo esquema (con una sola columna con el nombre `col` del tipo `array<string>`) se inserta cuando utiliza Apache Spark (Athena para Apache Spark) o la mayoría de los demás motores a fin de crear la tabla.

```
CREATE EXTERNAL TABLE
  [db_name.]table_name(col array<string>)
  LOCATION 's3://DOC-EXAMPLE-BUCKET/your-folder/'
  TBLPROPERTIES ('spark.sql.sources.provider' = 'delta')
```

3. Abra la consola de AWS Glue en <https://console.aws.amazon.com/glue/>.
4. En el panel de navegación, elija Catálogo de datos, Tablas.
5. En la lista de tablas, elija el enlace de su tabla.
6. En la página de la tabla, seleccione Acciones, Editar tabla.
7. En la sección de Parámetros Serde, agregue la clave `path` con el valor `s3://DOC-EXAMPLE-BUCKET/your-folder/`.
8. Seleccione Guardar.

## Recursos adicionales de

Para obtener información sobre el uso de las tablas de Delta Lake con AWS Glue y cómo consultarlas con Athena, consulte [Handle UPSERT data operations using open-source Delta Lake and AWS Glue](#) en el Blog de macrodatos de AWS.

## Uso de Athena para consultar conjuntos de datos de Apache Hudi

[Apache Hudi](#) es un marco de administración de datos de código abierto que simplifica el procesamiento incremental de datos. Las acciones de inserción, actualización, upsert y eliminación

a nivel de registro se procesan de manera mucho más pormenorizada, lo que reduce la sobrecarga. Upsert se refiere a la capacidad de insertar registros en un conjunto de datos si aún no existen o de actualizarlos en caso de que ya estén presentes.

Hudi gestiona los eventos de inserción y actualización de datos sin crear muchos archivos pequeños que pueden causar problemas de rendimiento para el análisis. Apache Hudi realiza un seguimiento automático de los cambios y fusiona los archivos, para que permanezcan en un tamaño óptimo. Esto evita la necesidad de crear soluciones personalizadas que monitoreen y reescriban muchos archivos pequeños en menos archivos grandes.

Los conjuntos de datos de Hudi son adecuados para los siguientes casos de uso:

- Cumplir con las regulaciones de privacidad, como el [Reglamento General de Protección de Datos \(RGPD\)](#) y la [Ley de Privacidad del Consumidor de California \(CCPA\)](#) que velan por el derecho de las personas a eliminar su información personal o cambiar la forma en que se utilizan sus datos.
- Trabajo con datos de transmisión desde sensores y otros dispositivos del Internet de las cosas (IoT) que requieren eventos específicos de inserción y actualización de datos.
- Implementar un [sistema de captura de datos modificados \(CDC, change data capture\)](#).

Los conjuntos de datos administrados por Hudi se almacenan en Amazon S3 mediante formatos de almacenamiento abierto. Actualmente, Athena puede leer conjuntos de datos Hudi compactados, pero no puede escribir datos Hudi. Athena admite hasta la versión 0.8.0 de Hudi con la versión 2 del motor de Athena, y hasta la versión 0.14.0 de Hudi con la versión 3 del motor de Athena. Esto está sujeto a cambios. Athena no puede garantizar la compatibilidad de lectura con tablas que se hayan creado con versiones posteriores de Hudi. Para obtener más información acerca del control de versiones del motor Athena, consulte [Control de versiones del motor Athena](#). Para obtener más información sobre las características y el control de versiones de Hudi, consulte la [documentación de Hudi](#) en el sitio web de Apache.

## Tipos de tablas de los conjuntos de datos de Hudi

Un conjunto de datos de Hudi puede tratarse de uno de los siguientes tipos:

- Copiar al escribir (CoW, Copy on Write): los datos se almacenan en un formato de columnas (Parquet) y cada actualización crea una nueva versión de los archivos durante una escritura.
- Fusionar al leer (MoR, Merge on Read): los datos se almacenan utilizando un formato que combina columnas (Parquet) y filas (Avro). Las actualizaciones se registran en archivos de `delta` basados



en filas y se compactan según sea necesario para crear nuevas versiones de los archivos en columnas.

Con los datasets de tipo CoW, cada vez que se produce una actualización de un registro, el archivo que contiene el registro se vuelve a escribir con los valores actualizados. Con un conjunto de datos de tipo MoR, cada vez que hay una actualización, Hudi escribe solo la fila correspondiente al registro modificado. MoR es más adecuado para cargas de trabajo con gran cantidad de escrituras o cambios y menos lecturas. CoW es más adecuado para cargas de trabajo con gran cantidad de lecturas con datos que cambian con menos frecuencia.

Hudi ofrece tres tipos de consulta para acceder a los datos:

- Consultas de instantáneas: consultas que ven la última instantánea de la tabla a partir de una acción de confirmación o compactación determinada. Para las tablas MoR, las consultas de instantáneas exponen el estado más reciente de la tabla mediante la combinación de los archivos base y delta del segmento de archivos más reciente en el momento de la consulta.
- Consultas progresivas: consultas que solo ven los nuevos datos escritos en la tabla, desde una confirmación o compactación determinada. Esto proporciona flujos de cambio de manera efectiva para habilitar canalizaciones de datos incrementales.
- Consultas optimizadas para lectura: para las tablas de MoR, las consultas ven compactados los datos más recientes. Para las tablas CoW, las consultas ven los últimos datos confirmados.

En la siguiente tabla, se muestran los tipos de consulta Hudi posibles para cada tipo de tabla.

Tipo de tabla	Tipos de consultas posibles de Hudi
Copiar al escribir	Instantánea, incremental
Fusionar al leer	instantánea, incremental, lectura optimizada

Actualmente, Athena admite consultas de instantáneas y consultas optimizadas de lectura, pero no consultas incrementales. En las tablas MoR, todos los datos expuestos a consultas optimizadas de lectura están compactados. Esto proporciona un buen rendimiento, pero no incluye las últimas

confirmaciones delta. Las consultas instantáneas contienen los datos más recientes, pero generan alguna sobrecarga informática, lo que hace que estas consultas sean menos eficientes.

Para obtener más información acerca de las ventajas e inconvenientes de los tipos de tablas y consultas, visite [Tipos de tablas y de consultas](#) en la documentación de Apache Hudi.

Cambio de la terminología de Hudi: las vistas ahora son consultas

A partir de la versión 0.5.1, Apache Hudi cambió parte de su terminología. Lo que antes eran vistas se denominan consultas en versiones posteriores. En la siguiente tabla, se resumen los cambios entre los términos anteriores y los nuevos.

Término antiguo	Término nuevo
CoW: vista optimizada de lectura	Consultas de instantáneas
MoR: vista en tiempo real	
Vista incremental	Consulta incremental
MoR: vista optimizada de lectura	Consulta optimizada de lectura

### Tablas de operación de Bootstrap

A partir de la versión 0.6.0 de Apache Hudi, la característica de operación de arranque proporciona un mejor rendimiento con los conjuntos de datos de Parquet existentes. En lugar de volver a escribir el conjunto de datos, una operación de arranque solo puede generar metadatos, lo que deja el conjunto de datos como está.

Athena se puede utilizar para consultar tablas desde una operación de arranque al igual que otras tablas basadas en datos de Amazon S3. En la instrucción `CREATE TABLE`, especifique la ruta de la tabla Hudi en la cláusula `LOCATION`.

Para obtener más información sobre cómo crear tablas Hudi mediante la operación de arranque en Amazon EMR, consulte el artículo [Nuevas características de Apache Hudi disponibles en Amazon EMR](#) en el Blog de macrodatos de AWS.

## Lista de metadatos de Hudi

Apache Hudi tiene una [tabla de metadatos](#) que contiene características de indexación para mejorar el rendimiento, como la lista de archivos, la omisión de datos mediante estadísticas de columnas y un índice basado en un filtro de Bloom.

En la actualidad, de estas características, Athena solo admite el índice de listas de archivos. El índice de listas de archivos elimina las llamadas al sistema de archivos, como aquellas destinadas a “enumerar archivos”, ya que obtiene la información de un índice que mantiene una partición para asignar los archivos. Esto elimina la necesidad de enumerar de forma recursiva todas y cada una de las particiones en la ruta de la tabla para obtener una vista del sistema de archivos. Al trabajar con conjuntos de datos de gran tamaño, esta indexación reduce de forma drástica la latencia que, de otro modo, se produciría al obtener la lista de archivos durante las escrituras y consultas. También evita cuellos de botella, como la limitación de los límites de solicitudes en las llamadas LIST de Amazon S3.

### Note

Athena no admite la omisión de datos ni la indexación del filtro de Bloom en este momento.

## Habilitación de la tabla de metadatos de Hudi

La lista de archivos basada en tablas de metadatos se encuentra deshabilitada de forma predeterminada. Para habilitar la tabla de metadatos de Hudi y la funcionalidad de lista de archivos relacionada, defina la propiedad de la tabla `hudi.metadata-listing-enabled` en TRUE.

## Ejemplo

En el siguiente ejemplo de `ALTER TABLE SET TBLPROPERTIES`, se habilita la tabla de metadatos de la tabla `partition_cow` de ejemplo.

```
ALTER TABLE partition_cow SET TBLPROPERTIES('hudi.metadata-listing-enabled'='TRUE')
```

## Consideraciones y limitaciones

- Athena no admite consultas incrementales.
- Athena no es compatible con [CTAS](#) o [INSERT INTO](#) sobre los datos de Hudi. Si desea soporte de Athena para escribir conjuntos de datos de Hudi, envíe sus comentarios a <athena-feedback@amazon.com>.

Para obtener más información sobre cómo escribir datos de Hudi, consulte los siguientes recursos:

- [Uso del conjunto de datos de Hudi](#) en la [Guía de publicación de Amazon EMR](#)
- [Escritura de datos](#) en la documentación de Apache Hudi.
- El uso de MSCK REPAIR TABLE en tablas Hudi en Athena no es compatible. Si necesita cargar una tabla Hudi no creada en AWS Glue, utilice [ALTER TABLE ADD PARTITION](#).
- No se admite la omisión de objetos de S3 Glacier: si los objetos de la tabla de Apache Hudi se encuentran en una clase de almacenamiento de Amazon S3 Glacier, establecer la propiedad de la tabla `read_restored_glacier_objects` en `false` no tendrá ningún efecto.

Suponga, por ejemplo, que ejecuta el siguiente comando:

```
ALTER TABLE table_name SET TBLPROPERTIES ('read_restored_glacier_objects' = 'false')
```

En el caso de las tablas de Iceberg y Delta Lake, el comando produce el error Clave de propiedad de tabla no compatible: `read_restored_glacier_objects`. En el caso de las tablas de Hudi, el comando `ALTER TABLE` no produce ningún error, pero los objetos de Amazon S3 Glacier siguen sin omitirse. Al ejecutar consultas `SELECT` después del comando `ALTER TABLE`, se siguen devolviendo todos los objetos.

## Video

En el siguiente video se muestra cómo puede utilizar Amazon Athena para consultar un conjunto de datos de Apache Hudi optimizado para lectura en su lago de datos basado en Amazon S3.

[Query Apache Hudi datasets using Amazon Athena](#) (Consulta de los conjuntos de datos de Apache Hudi con Amazon Athena)

## Creación de tablas de Hudi

En esta sección se proporcionan ejemplos de instrucciones CREATE TABLE en Athena para tablas particionadas y no particionadas de datos Hudi.

Si ya tiene tablas Hudi creadas en AWS Glue, puede consultarlas directamente en Athena. Cuando se crean tablas de Hudi particionadas en Athena, se debe ejecutar ALTER TABLE ADD PARTITION para cargar los datos de Hudi antes de poder consultarlos.

### Crear tablas de ejemplo de copiar al escribir (CoW)

#### Tabla CoW no particionada

En el siguiente ejemplo, se crea una tabla CoW sin particiones en Athena.

```
CREATE EXTERNAL TABLE `non_partition_cow`(  
  `_hoodie_commit_time` string,  
  `_hoodie_commit_seqno` string,  
  `_hoodie_record_key` string,  
  `_hoodie_partition_path` string,  
  `_hoodie_file_name` string,  
  `event_id` string,  
  `event_time` string,  
  `event_name` string,  
  `event_guests` int,  
  `event_type` string)  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe'  
STORED AS INPUTFORMAT  
  'org.apache.hudi.hadoop.HoodieParquetInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive.q1.io.parquet.MapredParquetOutputFormat'  
LOCATION  
  's3://DOC-EXAMPLE-BUCKET/folder/non_partition_cow/'
```

#### Tabla CoW particionada

En el siguiente ejemplo, se crea una tabla CoW particionada en Athena.

```
CREATE EXTERNAL TABLE `partition_cow`(  
  `_hoodie_commit_time` string,  
  `_hoodie_commit_seqno` string,
```

```

`_hoodie_record_key` string,
`_hoodie_partition_path` string,
`_hoodie_file_name` string,
`event_id` string,
`event_time` string,
`event_name` string,
`event_guests` int)
PARTITIONED BY (
  `event_type` string)
ROW FORMAT SERDE
  'org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
  'org.apache.hudi.hadoop.HoodieParquetInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.q1.io.parquet.MapredParquetOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET/folder/partition_cow/'

```

El ejemplo a continuación de ALTER TABLE ADD PARTITION agrega dos particiones a la tabla de ejemplo `partition_cow`.

```

ALTER TABLE partition_cow ADD
  PARTITION (event_type = 'one') LOCATION 's3://DOC-EXAMPLE-BUCKET/folder/
partition_cow/one/'
  PARTITION (event_type = 'two') LOCATION 's3://DOC-EXAMPLE-BUCKET/folder/
partition_cow/two/'

```

## Crear tablas de ejemplo de fusionar al leer (MoR)

Hudi crea dos tablas en el metaalmacén de MoR: una tabla para consultas de instantáneas y una tabla para consultas optimizadas de lectura. Ambas tablas se pueden consultar. En versiones de Hudi anteriores a 0.5.1, la tabla para consultas optimizadas de lectura tenía el nombre que había especificado al crear la tabla. A partir de la versión 0.5.1 de Hudi, el nombre de la tabla tiene el sufijo con `_ro` de forma predeterminada. El nombre de la tabla para las consultas de instantáneas es el nombre que especificó, al que se le agrega `_rt`.

### Tabla de fusionar al leer (MoR) no particionada

En el siguiente ejemplo, se crea una tabla MoR no particionada en Athena para consultas optimizadas de lectura. Tenga en cuenta que las consultas optimizadas de lectura utilizan el formato de entrada `HoodieParquetInputFormat`.

```
CREATE EXTERNAL TABLE `nonpartition_mor`(  
  `_hoodie_commit_time` string,  
  `_hoodie_commit_seqno` string,  
  `_hoodie_record_key` string,  
  `_hoodie_partition_path` string,  
  `_hoodie_file_name` string,  
  `event_id` string,  
  `event_time` string,  
  `event_name` string,  
  `event_guests` int,  
  `event_type` string)  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'  
STORED AS INPUTFORMAT  
  'org.apache.hudi.hadoop.HoodieParquetInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'  
LOCATION  
  's3://DOC-EXAMPLE-BUCKET/folder/nonpartition_mor/'
```

En el siguiente ejemplo, se crea una tabla MoR no particionada en Athena para consultas de instantáneas. Para consultas de instantáneas, utilice el formato de entrada `HoodieParquetRealtimeInputFormat`.

```
CREATE EXTERNAL TABLE `nonpartition_mor_rt`(  
  `_hoodie_commit_time` string,  
  `_hoodie_commit_seqno` string,  
  `_hoodie_record_key` string,  
  `_hoodie_partition_path` string,  
  `_hoodie_file_name` string,  
  `event_id` string,  
  `event_time` string,  
  `event_name` string,  
  `event_guests` int,  
  `event_type` string)  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'  
STORED AS INPUTFORMAT  
  'org.apache.hudi.hadoop.realtime.HoodieParquetRealtimeInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'  
LOCATION
```

```
's3://DOC-EXAMPLE-BUCKET/folder/nonpartition_mor/'
```

## Tabla de fusionar al leer (MoR) particionada

En el siguiente ejemplo se crea una tabla MoR particionada en Athena para consultas optimizadas de lectura.

```
CREATE EXTERNAL TABLE `partition_mor` (
  `_hoodie_commit_time` string,
  `_hoodie_commit_seqno` string,
  `_hoodie_record_key` string,
  `_hoodie_partition_path` string,
  `_hoodie_file_name` string,
  `event_id` string,
  `event_time` string,
  `event_name` string,
  `event_guests` int)
PARTITIONED BY (
  `event_type` string)
ROW FORMAT SERDE
  'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
  'org.apache.hudi.hadoop.HoodieParquetInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET/folder/partition_mor/'
```

El ejemplo a continuación de ALTER TABLE ADD PARTITION agrega dos particiones a la tabla de ejemplo partition\_mor.

```
ALTER TABLE partition_mor ADD
  PARTITION (event_type = 'one') LOCATION 's3://DOC-EXAMPLE-BUCKET/folder/
partition_mor/one/'
  PARTITION (event_type = 'two') LOCATION 's3://DOC-EXAMPLE-BUCKET/folder/
partition_mor/two/'
```

En el ejemplo siguiente se crea una tabla MoR particionada en Athena para consultas de instantáneas.

```
CREATE EXTERNAL TABLE `partition_mor_rt` (
```



```

`_hoodie_commit_time` string,
`_hoodie_commit_seqno` string,
`_hoodie_record_key` string,
`_hoodie_partition_path` string,
`_hoodie_file_name` string,
`event_id` string,
`event_time` string,
`event_name` string,
`event_guests` int)
PARTITIONED BY (
  `event_type` string)
ROW FORMAT SERDE
  'org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
  'org.apache.hudi.hadoop.realtime.HoodieParquetRealtimeInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.q1.io.parquet.MapredParquetOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET/folder/partition_mor/'

```

Del mismo modo, el ejemplo a continuación de ALTER TABLE ADD PARTITION agrega dos particiones a la tabla de ejemplo partition\_mor\_rt.

```

ALTER TABLE partition_mor_rt ADD
  PARTITION (event_type = 'one') LOCATION 's3://DOC-EXAMPLE-BUCKET/folder/
partition_mor/one/'
  PARTITION (event_type = 'two') LOCATION 's3://DOC-EXAMPLE-BUCKET/folder/
partition_mor/two/'

```

## Recursos adicionales de

- Si desea obtener información sobre cómo utilizar conectores personalizados de AWS Glue y trabajos de AWS Glue 2.0 para crear una tabla de Apache Hudi que se pueda consultar con Athena, consulte [Writing to Apache Hudi tables using AWS Glue custom connector](#) en el Blog sobre macrodatos de AWS.
- Para ver un artículo sobre el uso de Apache Hudi, AWS Glue y Amazon Athena a fin de crear un marco de procesamiento de datos para un lago de datos, consulte [Cómo simplificar el procesamiento de datos operativos en lagos de datos con AWS Glue y Apache Hudi](#) en el Blog de macrodatos de AWS.

## Uso de tablas de Apache Iceberg

Athena admite consultas de lectura, viaje en el tiempo, escritura y DDL para tablas de Apache Iceberg que utilizan el formato Apache Parquet para los datos y el catálogo de AWS Glue para su almacén de metadatos.

[Apache Iceberg](#) es un formato de tabla abierto para conjuntos de datos analíticos muy grandes. Iceberg administra grandes colecciones de archivos como tablas y admite operaciones modernas de lago de datos analíticos, como las consultas de inserción, actualización y eliminación de registros, y viajes en el tiempo. La especificación de Iceberg permite la evolución de las tablas sin problemas, como la evolución de esquemas y particiones, y se ha diseñado para el uso optimizado en Amazon S3. Iceberg también ayuda a garantizar la corrección de los datos en escenarios de escritura simultánea.

Para más información sobre Apache Iceberg, consulte <https://iceberg.apache.org/>.

### Consideraciones y limitaciones

La compatibilidad de Athena con las tablas de Iceberg tiene las siguientes consideraciones y limitaciones:

- Compatibilidad con la versión de Iceberg: Athena es compatible con la versión 1.4.2 de Apache Iceberg.
- Tablas de AWS Glue Catalog únicamente: solo las tablas de Iceberg creadas con AWS Glue Catalog basado en las especificaciones definidas por la [implementación de Glue Catalog de código abierto](#) son compatibles con Athena.
- Soporte de bloqueo de tablas de AWS Glue únicamente: a diferencia de la implementación del catálogo Glue de código abierto, que admite el bloqueo personalizado de complementos, Athena admite el bloqueo optimista de AWS Glue únicamente. El uso de Athena para modificar una tabla de Iceberg con cualquier otra implementación de bloqueo provocará posibles pérdidas de datos y la interrupción de transacciones.
- Formatos de archivo compatibles: la compatibilidad con el formato de archivo Iceberg en Athena depende de la versión del motor de Athena, como se muestra en la siguiente tabla.

Versión del motor Athena	Parquet	ORC	Avro
2	Sí	No	No

Versión del motor Athena	Parquet	ORC	Avro
3	Sí	Sí	Sí

- Tablas Iceberg v2: Athena solo crea y opera con tablas de Iceberg v2. Para ver la diferencia entre las tablas v1 y v2, consulte [Cambios de versión de formato](#) en la documentación de Apache Iceberg.
- Visualización de tipos de hora sin zona horaria: la hora y la marca de tiempo sin tipos de zona horaria se muestran en UTC. Si la zona horaria no se especifica en una expresión de filtro en una columna de hora, se usa UTC.
- Precisión de los datos relacionados con la marca de tiempo: si bien Iceberg admite una precisión de microsegundos para el tipo de datos de marca de tiempo, Athena solo admite una precisión de milisegundos para las marcas de tiempo tanto en las lecturas como en las escrituras. En el caso de los datos en las columnas relacionadas con la hora que se reescriben durante las operaciones de compactación manual, Athena solo retiene la precisión en milisegundos.
- Operaciones no admitidas: las siguientes operaciones de Athena no se admiten para las tablas de Iceberg.
  - [ALTER TABLE SET LOCATION](#)
- Vistas: se utiliza CREATE VIEW para crear vistas de Athena como se describe en [Uso de vistas](#). Si está interesado en utilizar la [especificación de vistas de Iceberg](#) para crear vistas, póngase en contacto con [athena-feedback@amazon.com](mailto:athena-feedback@amazon.com).
- Comandos de administración de TTF no compatibles en AWS Lake Formation: si bien puede utilizar Lake Formation a fin de administrar los permisos de acceso de lectura para los formatos de tabla de transacciones (TTF) como Apache Iceberg, Apache Hudi y Linux Foundation Delta Lake, no puede utilizar Lake Formation a fin de administrar los permisos de operaciones como VACUUM, MERGE, UPDATE o OPTIMIZE con estos formatos de tabla. Para obtener más información sobre la integración de Lake Formation con Athena, consulte [Uso de AWS Lake Formation con Amazon Athena](#) en la Guía para desarrolladores de AWS Lake Formation.
- Partición por campos anidados: no se admite la partición por campos anidados. Si lo intenta, aparecerá el mensaje NOT\_SUPPORTED: No se admite la partición por campos anidados: *column\_name.nested\_field\_name*.
- No se admite la omisión de objetos de S3 Glacier: si los objetos de la tabla de Apache Iceberg se encuentran en una clase de almacenamiento de Amazon S3 Glacier, establecer la propiedad de la tabla `read_restored_glacier_objects` en `false` no tendrá ningún efecto.

Suponga, por ejemplo, que ejecuta el siguiente comando:

```
ALTER TABLE table_name SET TBLPROPERTIES ('read_restored_glacier_objects' = 'false')
```

En el caso de las tablas de Iceberg y Delta Lake, el comando produce el error Clave de propiedad de tabla no compatible: `read_restored_glacier_objects`. En el caso de las tablas de Hudi, el comando `ALTER TABLE` no produce ningún error, pero los objetos de Amazon S3 Glacier siguen sin omitirse. Al ejecutar consultas `SELECT` después del comando `ALTER TABLE`, se siguen devolviendo todos los objetos.

Si desea que Athena admita una característica en particular, envíe sus comentarios a [athena-feedback@amazon.com](mailto:athena-feedback@amazon.com).

## Temas

- [Creación de tablas de Iceberg](#)
- [Administración de tablas de Iceberg](#)
- [Consulta de los metadatos de la tabla de Iceberg](#)
- [Esquema de tabla de Iceberg en evolución](#)
- [Consulta de tablas de Iceberg y realización de viajes en el tiempo](#)
- [Actualización de los datos de las tablas de Iceberg](#)
- [Optimización de las tablas de Iceberg](#)
- [Tipos de datos compatibles con las tablas de Iceberg en Athena](#)
- [Otras operaciones de Athena en tablas de Iceberg](#)
- [Recursos adicionales de](#)

## Creación de tablas de Iceberg

Para crear una tabla de Iceberg a fin de utilizarla en Athena, puede utilizar una instrucción `CREATE TABLE` como se documenta en esta página, o puede utilizar un rastreador de AWS Glue.

### Uso de la instrucción `CREATE TABLE`

Athena crea tablas de Iceberg v2. Para conocer la diferencia entre las tablas v1 y v2, consulta los [cambios de versión de formato](#) en la documentación de Apache Iceberg.

El comando `CREATE TABLE` de Athena crea una tabla de Iceberg sin datos. Puede consultar una tabla desde sistemas externos como Apache Spark directamente si la tabla utiliza el [catálogo de Glue de código abierto de Iceberg](#). No es necesario crear una tabla externa.

#### Warning

La ejecución del comando `CREATE EXTERNAL TABLE` da como resultado el mensaje de error `External keyword not supported for table type ICEBERG` (Palabra clave externa no admitida para el tipo de tabla ICEBERG).

Para crear una tabla de Iceberg en Athena, establezca la propiedad de tabla `'table_type'` como `'ICEBERG'` en la cláusula `TBLPROPERTIES`, como en el siguiente resumen de sintaxis.

```
CREATE TABLE
  [db_name.]table_name (col_name data_type [COMMENT col_comment] [, ...] )
  [PARTITIONED BY (col_name | transform, ... )]
  LOCATION 's3://DOC-EXAMPLE-BUCKET/your-folder/'
  TBLPROPERTIES ( 'table_type' = 'ICEBERG' [, property_name=property_value] )
```

Para obtener información sobre los tipos de datos que puede consultar en las tablas de Iceberg, consulte [Tipos de datos compatibles con las tablas de Iceberg en Athena](#).

## Particiones

Para crear tablas de Iceberg con particiones, use la sintaxis `PARTITIONED BY`. Las columnas que se usan para la partición deben especificarse primero en las instrucciones de las columnas. Dentro de la cláusula `PARTITIONED BY`, no se debe incluir el tipo de columna. También puede definir [transformaciones de partición](#) en la sintaxis `CREATE TABLE`. Para especificar varias columnas para la partición, sepárelas con el carácter coma (`,`), como en el siguiente ejemplo:

```
CREATE TABLE iceberg_table (id bigint, data string, category string)
  PARTITIONED BY (category, bucket(16, id))
  LOCATION 's3://DOC-EXAMPLE-BUCKET/your-folder/'
  TBLPROPERTIES ( 'table_type' = 'ICEBERG' )
```

En la siguiente tabla, se muestran las funciones de transformación de particiones disponibles.

Función	Descripción	Tipos admitidos
<code>year(ts)</code>	Partición por año	date, timestamp
<code>month(ts)</code>	Partición por mes	date, timestamp
<code>day(ts)</code>	Partición por día	date, timestamp
<code>hour(ts)</code>	Partición por hora	timestamp
<code>bucket(<i>N</i>, col)</code>	Partición por valor de hash mod <i>N</i> buckets. Este es el mismo concepto que el bucket de hash para tablas de Hive.	int, long, decimal, date, timestamp, string, binary
<code>truncate(<i>L</i>, col)</code>	Partición por valor truncado a <i>L</i>	int, long, decimal, string

Athena admite el particionamiento oculto de Iceberg. Para más información, consulte [Partición oculta de Iceberg](#) en la documentación de Apache Iceberg.

### Propiedades de la tabla

Esta sección describe las propiedades de la tabla que puedes especificar como pares valor de clave en la cláusula TBLPROPERTIES de la instrucción CREATE TABLE. Athena solo permite una lista predefinida de pares valor de clave en las propiedades de la tabla para crear o alterar tablas de Iceberg. En las siguientes tablas, se muestran las propiedades de la tabla que se pueden especificar. Para más información sobre las opciones de compactación, consulte [Optimización de las tablas de Iceberg](#) en este documento. Si desea que Athena incluya compatibilidad con una propiedad de configuración de tabla específica de código abierto, envíe sus comentarios a [athena-feedback@amazon.com](mailto:athena-feedback@amazon.com).

### formato

Descripción	Formato de datos del archivo
-------------	------------------------------

Valores de propiedad permitidos	Las combinaciones de formato de archivo y compresión compatibles varían en función de la versión del motor de Athena. Para obtener más información, consulte <a href="#">Compatibilidad de compresión de tablas de Iceberg por formato de archivo</a> .
Valor predeterminado	parquet

## write\_compression

Descripción	Códec de compresión de archivos
Valores de propiedad permitidos	Las combinaciones de formato de archivo y compresión compatibles varían en función de la versión del motor de Athena. Para obtener más información, consulte <a href="#">Compatibilidad de compresión de tablas de Iceberg por formato de archivo</a> .
Valor predeterminado	La compresión de escritura predeterminada varía en función de la versión del motor de Athena. Para obtener más información, consulte <a href="#">Compatibilidad de compresión de tablas de Iceberg por formato de archivo</a> .

## optimize\_rewrite\_data\_file\_threshold

Descripción	Configuración específica de optimización de datos. Si los archivos de datos que requieren optimización son menores que el límite determinado, los archivos no se reescriben. Esto permite acumular más archivos de datos para producir archivos más cercanos al tamaño objetivo y omitir cálculos innecesarios para ahorrar costos.
Valores de propiedad permitidos	Un número positivo. Debe ser menos de 50.
Valor predeterminado	5

## optimize\_rewrite\_delete\_file\_threshold

Descripción	Configuración específica de optimización de datos. Si los archivos de eliminación asociados a un archivo de datos son menores que el límite, el archivo de datos no se reescribe. Esto permite acumular más archivos de eliminación para cada archivo de datos para ahorrar costos.
Valores de propiedad permitidos	Un número positivo. Debe ser menos de 50.
Valor predeterminado	2

### vacuum\_min\_snapshots\_to\_keep

Descripción	<p>Cantidad mínima de instantáneas que se deben retener en la rama principal de una tabla.</p> <p>Este valor tiene prioridad sobre la propiedad <code>vacuum_max_snapshot_age_seconds</code>. Si el mínimo de instantáneas restantes tiene una antigüedad superior a la especificada por <code>vacuum_max_snapshot_age_seconds</code>, se conservan las instantáneas y se omite el valor de <code>vacuum_max_snapshot_age_seconds</code>.</p>
Valores de propiedad permitidos	Un número positivo.
Valor predeterminado	1

### vacuum\_max\_snapshot\_age\_seconds

Descripción	Antigüedad máxima de las instantáneas que se deben retener en la rama principal. Este valor se omite si el número mínimo restante de instantáneas especificado por <code>vacuum_min_snapshots_to_keep</code> es superior a la antigüedad especificada. Esta propiedad de comportamiento de la tabla corresponde a la propiedad <code>history.expire.max-snapshot-age-ms</code> de la configuración de Apache Iceberg.
-------------	---



Valores de propiedad permitidos	Un número positivo.
Valor predeterminado	432 000 segundos (5 días)

`vacuum_max_metadata_files_to_keep`

Descripción	Cantidad máxima de archivos de metadatos anteriores que se deben retener en la ramificación principal de la tabla.
Valores de propiedad permitidos	Un número positivo.
Valor predeterminado	100

## Ejemplo de instrucción CREATE TABLE

En el siguiente ejemplo, se crea una tabla de Iceberg que tiene tres columnas.

```
CREATE TABLE iceberg_table (
  id int,
  data string,
  category string)
PARTITIONED BY (category, bucket(16,id))
LOCATION 's3://DOC-EXAMPLE-BUCKET/iceberg-folder'
TBLPROPERTIES (
  'table_type'='ICEBERG',
  'format'='parquet',
  'write_compression'='snappy',
  'optimize_rewrite_delete_file_threshold'='10'
)
```

## CREATE TABLE AS SELECT (CTAS)

Para obtener información sobre la creación de una tabla de Iceberg mediante la instrucción `CREATE TABLE AS`, consulte [CREATE TABLE AS](#), con especial atención a la sección [Propiedades de la tabla CTAS](#).

## Uso de un rastreador de AWS Glue

Puede utilizar un rastreador de AWS Glue para registrar de forma automática sus tablas de Iceberg en AWS Glue Data Catalog. Si desea migrar desde otro catálogo de Iceberg, puede crear y programar un rastreador de AWS Glue y proporcionar las rutas de Amazon S3 en las que se encuentran las tablas de Iceberg. Puede especificar la profundidad máxima de las rutas de Amazon S3 que puede recorrer el rastreador de AWS Glue. Tras programar un rastreador de AWS Glue, este extrae la información del esquema y actualiza AWS Glue Data Catalog con los cambios del esquema cada vez que se ejecuta. El rastreador de AWS Glue admite la fusión de esquemas en todas las instantáneas y actualiza la ubicación más reciente del archivo de metadatos en AWS Glue Data Catalog. Para obtener más información, consulte [Catálogo de datos y rastreadores en AWS Glue](#).

## Administración de tablas de Iceberg

Athena admite las siguientes operaciones de DDL de una tabla para tablas de Iceberg.

### ALTER TABLE RENAME

Cambia el nombre de una tabla.

Dado que los metadatos de tabla de una tabla de Iceberg se almacenan en Simple Storage Service (Amazon S3), puede actualizar la base de datos y el nombre de la tabla de una tabla administrada de Iceberg sin afectar a la información subyacente de la tabla.

### Sinopsis

```
ALTER TABLE [db_name.]table_name RENAME TO [new_db_name.]new_table_name
```

### Ejemplo

```
ALTER TABLE my_db.my_table RENAME TO my_db2.my_table2
```

### ALTER TABLE SET PROPERTIES

Agrega propiedades a una tabla de Iceberg y establece sus valores asignados.

De acuerdo con las [especificaciones de Iceberg](#), las propiedades de la tabla se almacenan en el archivo de metadatos de la tabla de Iceberg en lugar de en el AWS Glue. Athena no acepta propiedades de tabla personalizadas. Consulte la sección [Propiedades de la tabla](#) para

conocer los pares clave-valor permitidos. Si desea que Athena incluya compatibilidad con una propiedad de configuración de tabla específica de código abierto, envíe sus comentarios a [athena-feedback@amazon.com](mailto:athena-feedback@amazon.com).

## Sinopsis

```
ALTER TABLE [db_name.]table_name SET TBLPROPERTIES ('property_name' =  
'property_value' [ , ... ])
```

## Ejemplo

```
ALTER TABLE iceberg_table SET TBLPROPERTIES (  
  'format'='parquet',  
  'write_compression'='snappy',  
  'optimize_rewrite_delete_file_threshold'='10'  
)
```

## ALTER TABLE UNSET PROPERTIES

Esto descarta las propiedades existentes de una tabla de Iceberg.

## Sinopsis

```
ALTER TABLE [db_name.]table_name UNSET TBLPROPERTIES ('property_name' [ , ... ])
```

## Ejemplo

```
ALTER TABLE iceberg_table UNSET TBLPROPERTIES ('write_compression')
```

## DESCRIBE TABLE

Describe información de la tabla.

## Sinopsis

```
DESCRIBE [FORMATTED] [db_name.]table_name
```

Cuando se especifica la opción FORMATTED, la salida muestra información adicional, como la ubicación y las propiedades de la tabla.

## Ejemplo

```
DESCRIBE iceberg_table
```

## DROP TABLE

Esto descarta una tabla de Iceberg.

### Warning

Dado que las tablas de Iceberg se consideran tablas administradas en Athena, cuando se descarta una tabla Iceberg también se eliminan todos los datos de la tabla.

## Sinopsis

```
DROP TABLE [IF EXISTS] [db_name.] table_name
```

## Ejemplo

```
DROP TABLE iceberg_table
```

## SHOW CREATE TABLE

Se muestra la instrucción DDL CREATE TABLE que se puede utilizar para recrear la tabla de Iceberg en Athena. Si Athena no puede reproducir la estructura de la tabla (por ejemplo, porque las propiedades de tabla personalizadas se especifican en la tabla), se genera el error UNSUPPORTED.

## Sinopsis

```
SHOW CREATE TABLE [db_name.] table_name
```

## Ejemplo

```
SHOW CREATE TABLE iceberg_table
```

## MOSTRAR LAS PROPIEDADES DE LA TABLA

Se muestra una o más propiedades de tabla de una tabla de Iceberg. Solo se muestran las propiedades de tabla compatibles con Athena.

## Sinopsis

```
SHOW TBLPROPERTIES [db_name.]table_name [('property_name']
```

## Ejemplo

```
SHOW TBLPROPERTIES iceberg_table
```

## Consulta de los metadatos de la tabla de Iceberg

En una consulta SELECT, puede utilizar las siguientes propiedades después de *table\_name* para consultar los metadatos de la tabla de Iceberg:

- *\$files*: muestra los archivos de datos actuales de una tabla.
- *\$manifests*: muestra los manifiestos de archivos actuales de una tabla.
- *\$history*: muestra el historial de una tabla.
- *\$partitions*: muestra las particiones actuales de una tabla.
- *\$snapshots*: muestra las instantáneas de una tabla.
- *\$refs*: muestra las referencias de una tabla.

## Sintaxis

En la siguiente instrucción se enumeran los archivos de una tabla de Iceberg.

```
SELECT * FROM "dbname". "tablename$files"
```

En la siguiente instrucción se enumeran los manifiestos de una tabla de Iceberg.

```
SELECT * FROM "dbname". "tablename$manifests"
```

En la siguiente instrucción se muestra el historial de una tabla de Iceberg.

```
SELECT * FROM "dbname". "tablename$history"
```

En el siguiente ejemplo se muestran las particiones de una tabla de Iceberg.

```
SELECT * FROM "dbname". "tablename$partitions"
```

En el siguiente ejemplo se muestran las instantáneas de una tabla de Iceberg.

```
SELECT * FROM "dbname". "tablename$snapshots"
```

En el siguiente ejemplo se muestran las referencias de una tabla de Iceberg.

```
SELECT * FROM "dbname". "tablename$refs"
```

## Esquema de tabla de Iceberg en evolución

Las actualizaciones de esquemas Iceberg son cambios de metadatos únicamente. No se modifican los archivos de datos cuando se realiza una actualización del esquema.

El formato Iceberg admite los siguientes cambios de evolución del esquema:

- Agregar: agrega una nueva columna a una tabla o `struct` anidada.
- Eliminar: elimina una columna existente de una tabla o `struct` anidada.
- Renombrar: cambia el nombre de una columna o campo existente en una `struct` anidada.
- Reordenar: cambia el orden de columnas.
- Promoción de tipos: amplía el tipo de una columna, campo de `struct`, clave de `map`, valor de `map` o elemento de `list`. En la actualidad, se admiten los siguientes casos para las tablas de Iceberg:
  - número entero a número entero grande
  - flotante a doble
  - aumento de la precisión de un tipo decimal

## ALTER TABLE ADD COLUMNS

Agrega una o más columnas a una tabla de Iceberg existente.

### Sinopsis

```
ALTER TABLE [db_name.] table_name ADD COLUMNS (col_name data_type [, ...])
```

### Ejemplos

En el siguiente ejemplo, se agrega una columna `comment` de tipo `string` a una tabla de Iceberg.

```
ALTER TABLE iceberg_table ADD COLUMNS (comment string)
```

En el siguiente ejemplo, se agrega una columna `point` de tipo `struct` a una tabla de Iceberg.

```
ALTER TABLE iceberg_table
ADD COLUMNS (point struct<x: double, y: double>)
```

En el siguiente ejemplo, se agrega una columna `points`, que es una matriz de estructuras, a una tabla de Iceberg.

```
ALTER TABLE iceberg_table
ADD COLUMNS (points array<struct<x: double, y: double>>)
```

## ALTER TABLE DROP COLUMN

Esto descarta una columna de una tabla de Iceberg existente.

### Sinopsis

```
ALTER TABLE [db_name.]table_name DROP COLUMN col_name
```

### Ejemplo

```
ALTER TABLE iceberg_table DROP COLUMN userid
```

## ALTER TABLE CHANGE COLUMN

Cambia el nombre, el tipo, el orden o el comentario de una columna.

### Note

`ALTER TABLE REPLACE COLUMNS` no se admite. Debido a que `REPLACE COLUMNS` elimina todas las columnas y, luego, agrega otras nuevas, no es compatible con Iceberg. `CHANGE COLUMN` es la sintaxis preferida para la evolución del esquema.

### Sinopsis

```
ALTER TABLE [db_name.]table_name
CHANGE [COLUMN] col_old_name col_new_name column_type
[COMMENT col_comment] [FIRST|AFTER column_name]
```

## Ejemplo

```
ALTER TABLE iceberg_table CHANGE comment blog_comment string AFTER id
```

## SHOW COLUMNS

Muestra las columnas de una tabla.

## Sinopsis

```
SHOW COLUMNS (FROM|IN) [db_name.] table_name
```

## Ejemplo

```
SHOW COLUMNS FROM iceberg_table
```

## Consulta de tablas de Iceberg y realización de viajes en el tiempo

Para consultar un conjunto de datos Iceberg, utilice una instrucción SELECT estándar como la siguiente. Las consultas siguen la [especificación del formato v2](#) de Apache Iceberg y hacen la fusión en lectura tanto de la posición como de los borrados de igualdad.

```
SELECT * FROM [db_name.] table_name [WHERE predicate]
```

Para optimizar los tiempos de consulta, todos los predicados se empujan hacia abajo, hacia donde viven los datos.

## Consultas de viaje en el tiempo y de viaje de versión

Cada tabla de Apache Iceberg mantiene un manifiesto con versiones de los objetos de Amazon S3 que contiene. Las versiones anteriores del manifiesto se pueden usar para las consultas de viaje en el tiempo y de viaje de versión.

Las consultas de viaje en el tiempo de Athena consultan en Amazon S3 para obtener los datos históricos de una instantánea coherente a partir de una fecha y hora específicas. Las consultas de viaje de versión en Athena consultan Amazon S3 para obtener datos históricos a partir de un ID de instantánea especificado.



## Consultas de viaje en el tiempo

Para ejecutar una consulta de viaje en el tiempo, use `FOR TIMESTAMP AS OF timestamp` después del nombre de la tabla en la instrucción `SELECT`, como en el siguiente ejemplo.

```
SELECT * FROM iceberg_table FOR TIMESTAMP AS OF timestamp
```

La hora del sistema que se especificará para viajar puede ser una marca temporal o una marca temporal con una zona horaria. Si no se especifica, Athena considera que el valor es una marca de tiempo en el huso horario UTC.

En el siguiente ejemplo de consultas de viaje en el tiempo, se seleccionan los datos de CloudTrail para la fecha y hora especificadas.

```
SELECT * FROM iceberg_table FOR TIMESTAMP AS OF TIMESTAMP '2020-01-01 10:00:00 UTC'
```

```
SELECT * FROM iceberg_table FOR TIMESTAMP AS OF (current_timestamp - interval '1' day)
```

## Consultas de viaje de versión

Para ejecutar una consulta de viaje de versiones (es decir, ver una instantánea coherente a partir de una versión especificada), utilice `FOR VERSION AS OF version` después del nombre de la tabla en la instrucción `SELECT`, como en el siguiente ejemplo.

```
SELECT * FROM [db_name.]table_name FOR VERSION AS OF version
```

El parámetro de la *versión* es el ID de instantánea `bigint` asociado a una versión de la tabla de Iceberg.

En el siguiente ejemplo de consulta de viaje de versiones, se seleccionan los datos de la versión especificada.

```
SELECT * FROM iceberg_table FOR VERSION AS OF 949530903748831860
```

**Note**

Las cláusulas `FOR SYSTEM_TIME AS OF` y `FOR SYSTEM_VERSION AS OF` de la versión 2 del motor de Athena se han sustituido por las cláusulas `FOR TIMESTAMP AS OF` y `FOR VERSION AS OF` de la versión 3 del motor de Athena.

## Recuperación del ID de la instantánea

Puede utilizar la clase [SnapshotUtil](#) de Java proporcionada por Iceberg para recuperar el ID de una instantánea de Iceberg, como se muestra en el siguiente ejemplo.

```
import org.apache.iceberg.Table;
import org.apache.iceberg.aws.glue.GlueCatalog;
import org.apache.iceberg.catalog.TableIdentifier;
import org.apache.iceberg.util.SnapshotUtil;

import java.text.SimpleDateFormat;
import java.util.Date;

Catalog catalog = new GlueCatalog();

Map<String, String> properties = new HashMap<String, String>();
properties.put("warehouse", "s3://DOC-EXAMPLE-BUCKET/my-folder");
catalog.initialize("my_catalog", properties);

Date date = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").parse("2022/01/01 00:00:00");
long millis = date.getTime();

TableIdentifier name = TableIdentifier.of("db", "table");
Table table = catalog.loadTable(name);
long oldestSnapshotIdAfter2022 = SnapshotUtil.oldestAncestorAfter(table, millis);
```

## Combinación de viajes en el tiempo y de versión

Se puede usar la sintaxis de viaje en el tiempo y de versión en la misma consulta para especificar diferentes condiciones de tiempo y de control de versiones, como en el siguiente ejemplo.

```
SELECT table1.*, table2.* FROM
  [db_name.]table_name FOR TIMESTAMP AS OF (current_timestamp - interval '1' day) AS
  table1
```

```
FULL JOIN
[db_name.]table_name FOR VERSION AS OF 5487432386996890161 AS table2
ON table1.ts = table2.ts
WHERE (table1.id IS NULL OR table2.id IS NULL)
```

## Creación y consulta de vistas con tablas de Iceberg

Para crear y consultar vistas de Athena en tablas de Iceberg, utilice las vistas CREATE VIEW tal y como se describe en [Uso de vistas](#).

Ejemplo:

```
CREATE VIEW view1 AS SELECT * FROM iceberg_table
```

```
SELECT * FROM view1
```

Si está interesado en utilizar la [especificación de vistas de Iceberg](#) para crear vistas, póngase en contacto con [athena-feedback@amazon.com](mailto:athena-feedback@amazon.com).

## Trabajar con el control de acceso detallado de Lake Formation

La versión 3 del motor de Athena admite el control de acceso detallado de Lake Formation con tablas de Iceberg, incluido el control de acceso de seguridad de nivel de columna y fila. Este control de acceso funciona con consultas de viajes en el tiempo y con tablas que han realizado la evolución del esquema. Para obtener más información, consulte [Control de acceso detallado de Lake Formation y grupos de trabajo de Athena](#).

Si ha creado su tabla de Iceberg fuera de Athena, utilice la versión 0.13.0 o superior del [SDK de Apache Iceberg](#) para que la información de la columna de la tabla de Iceberg aparezca en el AWS Glue Data Catalog. Si la tabla de Iceberg no contiene información sobre columnas en AWS Glue, puede utilizar la instrucción [ALTER TABLE SET PROPERTIES](#) de Athena o el último SDK de Iceberg para corregir la tabla y actualizar la información de la columna en AWS Glue.

## Actualización de los datos de las tablas de Iceberg

Los datos de la tabla de Iceberg se pueden administrar directamente en Athena mediante consultas INSERT, UPDATE y DELETE. Cada transacción de administración de datos produce una nueva instantánea, que se puede consultar por medio de un viaje en el tiempo. Las instrucciones UPDATE y DELETE siguen la especificación de [eliminación de posiciones](#) a nivel filas del formato Iceberg v2 y aplican el aislamiento de instantáneas.

Use los siguientes comandos para realizar operaciones de administración de datos en tablas de Iceberg.

## INSERT INTO

Este comando inserta datos en una tabla de Iceberg. El comando `INSERT INTO` de Athena Iceberg se carga igual que las consultas actuales `INSERT INTO` para tablas de Hive externas según la cantidad de datos analizados. Para insertar datos en una tabla de Iceberg, use la siguiente sintaxis, donde la *consulta* puede ser `VALUES (val1, val2, ...)` o `SELECT (col1, col2, ...)` FROM `[db_name.]table_name` WHERE *predicate*. Para obtener información sobre la sintaxis y la semántica de SQL, consulte [INSERT INTO](#).

```
INSERT INTO [db_name.]table_name [(col1, col2, ...)] query
```

Los siguientes ejemplos insertan valores en la tabla `iceberg_table`.

```
INSERT INTO iceberg_table VALUES (1,'a','c1')
```

```
INSERT INTO iceberg_table (col1, col2, ...) VALUES (val1, val2, ...)
```

```
INSERT INTO iceberg_table SELECT * FROM another_table
```

## DELETE

El comando `DELETE` de Athena Iceberg escribe los archivos de eliminación de posición Iceberg en una tabla. Esto se conoce como eliminación de tipo fusionar al leer. En contraste con la eliminación de tipo copiar al escribir, una eliminación de tipo fusionar al leer es más eficiente porque no se reescriben los datos de los archivos. Cuando Athena lee los datos de Iceberg, fusiona los archivos de eliminación de posición de Iceberg con los archivos de datos para producir la última vista de una tabla. Para eliminar estos archivos de eliminación de posición, puede ejecutar la acción de compactación [REWRITE DATA](#). Las operaciones `DELETE` se cobran según la cantidad de datos analizados. Para ver la sintaxis, consulte [DELETE](#).

En el siguiente ejemplo, se eliminan las filas de `iceberg_table` que tienen `c3` como valor de `category`.

```
DELETE FROM iceberg_table WHERE category='c3'
```

## UPDATE

El comando UPDATE de Athena Iceberg escribe los archivos de eliminación de posición de Iceberg y las filas recién actualizadas como archivos de datos en la misma transacción. Se puede considerar al comando UPDATE como una combinación de INSERT INTO y DELETE. Las operaciones UPDATE se cobran según la cantidad de datos analizados. Para ver la sintaxis, consulte [UPDATE](#).

En el siguiente ejemplo, se actualizan los valores especificados en la tabla `iceberg_table`.

```
UPDATE iceberg_table SET category='c2' WHERE category='c1'
```

## MERGE INTO

Actualiza, elimina o inserta filas en una tabla de Iceberg de forma condicional. Una sola instrucción puede combinar acciones de actualización, eliminación e inserción. Para ver la sintaxis, consulte [MERGE INTO](#).

### Note

MERGE INTO es transaccional y solo es compatible con las tablas de Apache Iceberg en la versión 3 del motor Athena.

El siguiente ejemplo elimina todos los clientes de la tabla `t` que se encuentran en la tabla de origen `s`.

```
MERGE INTO accounts t USING monthly_accounts_update s
ON t.customer = s.customer
WHEN MATCHED
THEN DELETE
```

En el siguiente ejemplo, se actualiza la tabla de destino `t` con la información del cliente de la tabla de origen `s`. Para las filas de clientes de la tabla `t` que tienen filas de clientes coincidentes en la tabla `s`, el ejemplo incrementa las compras de la tabla `t`. Si la tabla `t` no coincide con ninguna fila de clientes de la tabla `s`, en el ejemplo se inserta la fila de clientes de una tabla `s` a otra `t`.

```
MERGE INTO accounts t USING monthly_accounts_update s
ON (t.customer = s.customer)
```

```
WHEN MATCHED
  THEN UPDATE SET purchases = s.purchases + t.purchases
WHEN NOT MATCHED
  THEN INSERT (customer, purchases, address)
    VALUES(s.customer, s.purchases, s.address)
```

El siguiente ejemplo actualiza condicionalmente la tabla de destino `t` con información de la tabla de origen `s`. El ejemplo elimina cualquier fila de destino coincidente cuya dirección de origen sea Centreville. Para todas las demás filas coincidentes, el ejemplo agrega las compras de origen y establece la dirección de destino en la dirección de origen. Si no hay ninguna coincidencia en la tabla de destino, el ejemplo inserta la fila de la tabla de origen.

```
MERGE INTO accounts t USING monthly_accounts_update s
  ON (t.customer = s.customer)
  WHEN MATCHED AND s.address = 'Centreville'
    THEN DELETE
  WHEN MATCHED
    THEN UPDATE
      SET purchases = s.purchases + t.purchases, address = s.address
  WHEN NOT MATCHED
    THEN INSERT (customer, purchases, address)
      VALUES(s.customer, s.purchases, s.address)
```

## Optimización de las tablas de Iceberg

A medida que los datos se acumulan en una tabla de Iceberg, las consultas se vuelven gradualmente menos eficientes debido al aumento del tiempo de procesamiento necesario para abrir los archivos. Se incurre en un costo computacional adicional si la tabla contiene [archivos de eliminación](#). En Iceberg, los archivos de eliminación almacenan las filas eliminadas, y el motor debe aplicar las filas eliminadas a los resultados de la consulta.

Para optimizar el rendimiento de las consultas en las tablas de Iceberg, Athena admite la compactación manual como comando de mantenimiento de la tabla. Las compactaciones optimizan la disposición estructural de la tabla sin alterar su contenido.

### OPTIMIZE

La acción de compactación `OPTIMIZE table REWRITE DATA` reescribe los archivos de datos en una disposición más optimizada en función de su tamaño y del número de archivos de eliminación asociados. Para obtener detalles sobre la sintaxis y las propiedades de la tabla, consulte [OPTIMIZE](#).

## Ejemplo

El siguiente ejemplo fusiona archivos de borrado en archivos de datos y produce archivos cercanos al tamaño de archivo objetivo cuando el valor de `category` es `c1`.

```
OPTIMIZE iceberg_table REWRITE DATA USING BIN_PACK
WHERE category = 'c1'
```

## VACUUM

VACUUM lleva a cabo el [vencimiento de las instantáneas](#) y la [eliminación de archivos huérfanos](#). Estas acciones reducen el tamaño de los metadatos y eliminan los archivos que no están en el estado actual de la tabla y que también tienen una antigüedad superior al periodo de retención especificado para la tabla. Para obtener detalles sobre la sintaxis, consulte [VACUUM](#).

## Ejemplo

En el siguiente ejemplo, se usa una propiedad de tabla para configurar la tabla `iceberg_table` de manera que retenga los datos de los últimos tres días y, a continuación, VACUUM se usa para hacer vencer las instantáneas antiguas y eliminar los archivos huérfanos de la tabla.

```
ALTER TABLE iceberg_table SET TBLPROPERTIES (
  'vacuum_max_snapshot_age_seconds'='259200'
)

VACUUM iceberg_table
```

## Tipos de datos compatibles con las tablas de Iceberg en Athena

Athena puede consultar las tablas de Iceberg que contienen los siguientes tipos de datos:

```
binary
boolean
date
decimal
double
float
int
list
long
map
string
```

```
struct
timestamp without time zone
```

Para más información sobre los tipos de tablas de Iceberg, consulte la [página de esquemas para Iceberg](#) en la documentación de Apache.

En la siguiente tabla, se muestra la relación entre los tipos de datos de Athena y los tipos de datos de tabla de Iceberg.

Tipo Iceberg	Tipo Athena	Notas
boolean	boolean	
-	tinyint	No es compatible con las tablas de Iceberg en Athena.
-	smallint	No es compatible con las tablas de Iceberg en Athena.
int	int	En las instrucciones DML de Athena, este tipo INTEGER.
long	bigint	
double	double	
float	float	
decimal(P, S)	decimal(P, S)	P es la precisión, S es la escala.
-	char	No es compatible con las tablas de Iceberg en Athena.
string	string	En las instrucciones DML de Athena, este tipo VARCHAR.
binary	binary	
date	date	
time	-	Solo se admite la marca de tiempo de Iceberg (sin zona horaria) para las instrucciones DDL de Athena Iceberg como CREATE TABLE, pero todos los tipos de marca de tiempo pueden consultarse a través de Athena.
timestamp	timestamp	



Tipo Iceberg	Tipo Athena	Notas
timestamp tz	timestamp tz	
list<E>	array	
map<K,V>	map	
struct<.. >	struct	
fixed(L)	-	El tipo fixed(L) no se admite actualmente en Athena.

Para más información sobre los tipos de datos en Athena, consulte [Tipos de datos en Amazon Athena](#).

## Otras operaciones de Athena en tablas de Iceberg

### Operaciones en el nivel de base de datos

Cuando utiliza [DROP DATABASE](#) con la opción CASCADE, también se eliminan todos los datos de la tabla de Iceberg. Las siguientes operaciones de DDL no tienen efecto en las tablas de Iceberg.

- [CREATE DATABASE](#)
- [ALTER DATABASE SET DBPROPERTIES](#)
- [SHOW DATABASES](#)
- [SHOW TABLES](#)
- [SHOW VIEWS](#)

### Operaciones relacionadas con la partición

Debido a que las tablas de Iceberg utilizan la [partición oculta](#), no tiene que trabajar directamente con particiones físicas. Como resultado, las tablas de Iceberg de Athena no admiten las siguientes operaciones de DDL relacionadas con particiones:

- [SHOW PARTITIONS](#)

- [ALTER TABLE ADD PARTITION](#)
- [ALTER TABLE DROP PARTITION](#)
- [ALTER TABLE RENAME PARTITION](#)

Si quiere ver la [evolución de las particiones](#) de Iceberg en Athena, envíe comentarios a [athena-feedback@amazon.com](mailto:athena-feedback@amazon.com).

## Descarga de las tablas de Iceberg

Las tablas de Iceberg se pueden descargar en los archivos de una carpeta de Amazon S3. Para obtener más información, consulte [UNLOAD](#).

## MSCK REPAIR

Debido a que las tablas de Iceberg hacen un seguimiento de la información de diseño de la tabla, la ejecución de [MSCK REPAIR TABLE](#) como se hace con las tablas Hive no es necesaria y no es compatible.

## Recursos adicionales de

Para ver artículos detallados sobre el uso de Athena con tablas de Apache Iceberg, consulte las siguientes publicaciones en el Blog de macrodatos de AWS.

- [Acelerar la ingeniería de características de ciencia de datos en lagos de datos transaccionales mediante Amazon Athena con Apache Iceberg](#)
- [Crear un lago de datos de Apache Iceberg mediante Amazon Athena, Amazon EMR y AWS Glue](#)
- [Realizar cambios en un lago de datos mediante Amazon Athena y Apache Iceberg](#)
- [Crear un lago de datos transaccional mediante Apache Iceberg y AWS Glue, y compartir datos entre cuentas con AWS Lake Formation y Amazon Athena](#)
- [Utilizar Apache Iceberg en un lago de datos para respaldar el procesamiento incremental de datos](#)
- [Crear un lago de datos de Apache Iceberg en tiempo real alineado con el RGPD](#)
- [Automatizar la replicación de fuentes relacionales en un lago de datos transaccional con Apache Iceberg y AWS Glue](#)
- [Interactuar con las tablas de Apache Iceberg mediante Amazon Athena y con permisos detallados entre cuentas con AWS Lake Formation](#)

- [Crear un lago de datos transaccionales sin servidor con Apache Iceberg, Amazon EMR sin servidor y Amazon Athena](#)

## Seguridad de Amazon Athena

La seguridad en la nube de AWS es la máxima prioridad. Como cliente de AWS, se beneficia de una arquitectura de red y un centro de datos que se han diseñado para satisfacer los requisitos de seguridad de las organizaciones más exigentes.

La seguridad es una responsabilidad compartida entre AWS y el usuario. El [modelo de responsabilidad compartida](#) la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta Servicios de AWS en la nube de AWS. Además, AWS proporciona servicios que puede utilizar de forma segura. Auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad en el marco de los [programas de conformidad de AWS](#). Para obtener más información sobre los programas de conformidad que se aplican a Athena, consulte [Servicios de AWS en el ámbito del programa de conformidad](#).
- Seguridad en la nube: su responsabilidad viene determinada por el Servicio de AWS que utilice. Usted también es responsable de otros factores incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Esta documentación lo ayudará a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza Amazon Athena. En los siguientes temas, se le mostrará cómo configurar Athena para satisfacer sus objetivos de seguridad y conformidad. También puede aprender a utilizar otros Servicios de AWS que lo ayudan a monitorear y proteger sus recursos de Athena.

### Temas

- [Protección de los datos en Athena](#)
- [Identity and Access Management en Athena](#)
- [Registro y supervisión en Athena](#)
- [Validación de la conformidad para Amazon Athena](#)
- [Resiliencia en Athena](#)
- [Seguridad de la infraestructura en Athena](#)
- [Configuración y análisis de vulnerabilidades en Athena](#)

- [Uso de Athena para consultar datos registrados en AWS Lake Formation](#)

## Protección de los datos en Athena

El [modelo de responsabilidad compartida](#) de AWS se aplica a la protección de datos en Amazon Athena. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta toda la Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Usted también es responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las [Preguntas frecuentes sobre la privacidad de datos](#). Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el [Modelo de responsabilidad compartida de AWS y GDPR](#) en el Blog de seguridad de AWS.

Con fines de protección de datos, recomendamos proteger las credenciales de la Cuenta de AWS y configurar cuentas de usuario individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos de AWS. Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure los registros de API y de actividad de los usuarios con AWS CloudTrail.
- Utilice las soluciones de cifrado de AWS, junto con todos los controles de seguridad predeterminados dentro de los servicios de Servicios de AWS.
- Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados FIPS 140-2 al acceder a AWS a través de una interfaz de línea de comandos o una API, utilice un punto de conexión de FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte [Estándar de procesamiento de la información federal \(FIPS\) 140-2](#).

Se recomienda encarecidamente no ingresar nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Name (Nombre). Tampoco debe hacerlo cuando trabaje con Athena u otros Servicios de AWS mediante la consola, la API, la AWS CLI o los AWS SDK. Cualquier dato que

ingrese en etiquetas o campos de formato libre utilizados para nombres se pueden emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Como paso de seguridad adicional, puede utilizar la clave de contexto de condición global [aws:CalledVia](#) para limitar las solicitudes únicamente a las hechas desde Athena. Para obtener más información, consulte [Uso de Athena con claves de contexto CalledVia](#).

## Protección de varios tipos de datos

Intervienen varios tipos de datos cuando se utiliza Athena para crear bases de datos y tablas. Estos tipos de datos incluyen los datos de origen almacenados en Amazon S3, metadatos para bases de datos y tablas que se crean al ejecutar consultas o el rastreador de AWS Glue para detectar datos, datos de resultados de consultas e historial de consultas. En esta sección se analiza cada tipo de datos y se ofrece orientación sobre cómo protegerlo.

- Datos de origen: los datos de las bases de datos y tablas se almacenan en Amazon S3 y Athena no los modifica. Para obtener más información, consulte [Protección de datos en Simple Storage Service \(Amazon S3\)](#) en la Guía del usuario de Amazon Simple Storage Service. Usted controla el acceso a sus datos de origen y puede cifrarlos en Amazon S3. Puede utilizar Athena para [crear tablas basadas en conjuntos de datos cifrados en Amazon S3](#).
- Base de datos y metadatos de tabla (esquema): Athena utiliza tecnología de lectura de esquema, lo que significa que las definiciones de tabla se aplican a los datos de Amazon S3 cuando Athena ejecuta consultas. Cualquier esquema que defina se guardará automáticamente a menos que lo elimine explícitamente. En Athena, puede modificar los metadatos del catálogo de datos con instrucciones DDL. También puede eliminar las definiciones y los esquemas de las tablas sin que los datos subyacentes almacenados en Amazon S3 se vean afectados. Los metadatos de las bases de datos y tablas que se utilizan en Athena se almacenan en el AWS Glue Data Catalog.

Puede [definir políticas de acceso detallado a bases de datos y tablas](#) registradas en AWS Glue Data Catalog mediante AWS Identity and Access Management (IAM). También puede [cifrar los metadatos en AWS Glue Data Catalog](#). Si cifra los metadatos, para el acceso utilice los [permisos para metadatos cifrados](#).

- Resultados de consulta e historial de consultas, incluidas las consultas guardadas: los resultados de las consultas se almacenan en una ubicación en Amazon S3 que puede elegir especificar globalmente o para cada grupo de trabajo. Si no se especifica, Athena utiliza la ubicación predeterminada en cada caso. Usted controla el acceso a los buckets de Amazon S3 donde

almacena los resultados de las consultas y las consultas guardadas. Además, puede elegir cifrar los resultados de la consulta que almacena en Amazon S3. Sus usuarios deben tener los permisos adecuados para obtener acceso a las ubicaciones de Amazon S3 y descifrar archivos. Para obtener más información, consulte [Cifrado de los resultados de las consultas de Athena en Amazon S3](#) en este documento.

Athena conserva el historial de consultas durante 45 días. Puede [ver el historial de consultas](#) utilizando las API Athena, en la consola y con AWS CLI. Para almacenar las consultas durante más de 45 días, guárdelas. Para proteger el acceso a las consultas guardadas, [utilice grupos de trabajo](#) en Athena y restrinja el acceso a las consultas guardadas solo a los usuarios que estén autorizados a verlas.

## Temas

- [Cifrado en reposo](#)
- [Cifrado en tránsito](#)
- [Administración de claves](#)
- [Privacidad del tráfico entre redes](#)

## Cifrado en reposo

Puede ejecutar consultas en Amazon Athena sobre datos cifrados en Amazon S3 en la misma región y en un número limitado de regiones. También puede cifrar los resultados de la consulta en Amazon S3, y los datos del catálogo de datos de AWS Glue.

Puede cifrar los siguientes recursos en Athena:

- Los resultados de todas las consultas en Amazon S3, que Athena almacena en una ubicación conocida como la ubicación de resultados de Amazon S3. Puede cifrar los resultados de consultas almacenados en Amazon S3 tanto si el conjunto de datos subyacente está cifrado en Amazon S3 como si no. Para obtener más información, consulte [Cifrado de los resultados de las consultas de Athena en Amazon S3](#).
- Contenido del catálogo de datos de AWS Glue Para obtener más información, consulte [Permisos para metadatos cifrados en el catálogo de datos de AWS Glue](#).

**Note**

Cuando se utiliza Athena para leer una tabla cifrada, Athena utiliza las opciones de cifrado especificadas para los datos de la tabla, no la opción de cifrado para los resultados de la consulta. Si se configuran métodos o claves de cifrado independientes para los resultados de la consulta y los datos de la tabla, Athena lee los datos de la tabla sin utilizar la opción de cifrado y la clave utilizadas para cifrar o descifrar los resultados de la consulta.

Sin embargo, si utiliza Athena para insertar datos en una tabla que tiene datos cifrados, Athena utiliza la configuración de cifrado que se especificó para los resultados de la consulta para cifrar los datos insertados. Por ejemplo, si especifica el cifrado CSE\_KMS para los resultados de la consulta, Athena utiliza el mismo identificador de clave AWS KMS que utilizó para el cifrado de los resultados de la consulta para cifrar los datos de la tabla insertada con CSE\_KMS


**Temas**

- [Opciones de cifrado de Simple Storage Service \(Amazon S3\) compatibles](#)
- [Permisos para datos cifrados en Amazon S3](#)
- [Permisos para metadatos cifrados en el catálogo de datos de AWS Glue](#)
- [Cifrado de los resultados de las consultas de Athena en Amazon S3](#)
- [Creación de tablas basadas en conjuntos de datos cifrados en Amazon S3](#)

**Opciones de cifrado de Simple Storage Service (Amazon S3) compatibles**

Athena admite las siguientes opciones de cifrado para conjuntos de datos y resultados de consulta en Amazon S3.

Tipo de cifrado	Descripción	Compatibilidad entre regiones
<a href="#">SSE-S3</a>	Cifrado de lado del servidor (SSE) con una clave administrada por Amazon S3.	Sí
<a href="#">SSE-KMS</a>	Cifrado de lado del servidor (SSE) con una clave administrada por el cliente de AWS Key Management Service.	Sí

Tipo de cifrado	Descripción	Compatibilidad entre regiones
	<div data-bbox="354 289 472 331">  Note         </div> <p data-bbox="402 352 1089 478">Con este tipo de cifrado, Athena no requiere que indique que los datos están cifrados al crear una tabla.</p>	
<a href="#">CSE-KMS</a>	<p data-bbox="321 562 1179 884">Cifrado del cliente (CSE) con una clave administrada por el cliente de AWS KMS. En Athena, esta opción requiere que utilice una instrucción CREATE TABLE con una cláusula TBLPROPERTIES que especifica 'has_encrypted_data' = 'true' . Para obtener más información, consulte <a href="#">Creación de tablas basadas en conjuntos de datos cifrados en Amazon S3</a>.</p>	No

Para obtener más información sobre el cifrado de AWS KMS con Simple Storage Service (Amazon S3), consulte [¿Qué es AWS Key Management Service?](#) y [Cómo Amazon Simple Storage Service \(Amazon S3\) utiliza AWS KMS](#) en la Guía para desarrolladores de AWS Key Management Service. Para obtener más información sobre el uso de SSE-KMS o CSE-KMS con Athena, consulte [Lanzamiento: Amazon Athena agrega compatibilidad para las consultas de datos cifrados](#) en el Blog de macrodatos de AWS.

#### Opciones no admitidas

No se admiten las siguientes opciones de cifrado:

- SSE con claves proporcionadas por el cliente (SSE-C)
- Cifrado del cliente con una clave administrada del cliente.
- Claves asimétricas

Para comparar las opciones de cifrado de Amazon S3, consulte [Protección de datos mediante cifrado](#) en la Guía del usuario de Amazon Simple Storage Service.



## Herramientas para el cifrado del cliente

Tenga en cuenta que hay dos herramientas disponibles para el cifrado del cliente:

- [Amazon S3 encryption client](#): esto cifra los datos solo para Amazon S3 y es compatible con Athena.
- [AWS Encryption SDK](#): el SDK se puede utilizar para cifrar datos en cualquier lugar de AWS, pero no es directamente compatible con Athena.

Estas herramientas no son compatibles; además, los datos cifrados con una herramienta no pueden ser descifrados por la otra. Athena solo admite Amazon S3 Encryption Client directamente. Si se utiliza el SDK para cifrar los datos, se pueden ejecutar consultas desde Athena, pero los datos se devuelven como texto cifrado.

Si se desea utilizar Athena para consultar datos cifrados con el AWS Encryption SDK, es necesario descargar y descifrar los datos, y luego cifrarlos de nuevo con Amazon S3 Encryption Client.

## Permisos para datos cifrados en Amazon S3

En función del tipo de cifrado que se utiliza en Amazon S3, es posible que tenga que agregar permisos, también conocidos como acciones “Permitir”, a las políticas utilizadas en Athena:

- SSE-S3: si se utiliza SSE-S3 para el cifrado, los usuarios de Athena no requieren permisos adicionales en sus políticas. Basta con disponer de los permisos de Amazon S3 adecuados para la ubicación de Amazon S3 correspondiente y para las acciones de Athena. Para obtener más información sobre las políticas que permiten los permisos de Athena y Amazon S3 adecuados, consulte [Políticas administradas de AWS para Amazon Athena](#) y [Acceso a Amazon S3](#).
- AWS KMS: si utiliza AWS KMS para el cifrado, los usuarios de Athena deben poder realizar acciones de AWS KMS concretas, además de tener los permisos de Athena y Amazon S3. Puede permitir estas acciones mediante la edición de la política de claves para las claves administradas por el cliente (CMK) de AWS KMS que se utilizan para cifrar los datos en Amazon S3. Para agregar usuarios de claves a las políticas de claves de AWS KMS adecuadas, puede usar la consola de AWS KMS en <https://console.aws.amazon.com/kms>. Para obtener información sobre cómo agregar un usuario a una política de claves de AWS KMS, consulte [Permitir a los usuarios de claves utilizar la CMK](#) en la Guía para desarrolladores de AWS Key Management Service.

**Note**

Los administradores de políticas de claves avanzadas pueden ajustar las políticas de claves. `kms:Decrypt` es la acción mínima permitida para que un usuario de Athena pueda trabajar con un conjunto de datos cifrados. Para trabajar con resultados de consulta cifrados, las acciones mínimas permitidas son `kms:GenerateDataKey` y `kms:Decrypt`.

Cuando se utiliza Athena para consultar conjuntos de datos en Amazon S3 que tengan un gran número de objetos cifrados con AWS KMS, AWS KMS puede limitar los resultados de las consultas. Esto es más probable que se produzca cuando hay un gran número de objetos pequeños. Athena deja de utilizar las solicitudes de reintento, pero es posible que siga produciéndose un error de limitación controlada. Si trabaja con un gran número de objetos cifrados y experimenta este problema, una opción es habilitar las claves de bucket de Simple Storage Service (Amazon S3) para reducir el número de llamadas a KMS. Para obtener más información, consulte [Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3](#) en la Guía del usuario de Amazon Simple Storage Service. Otra opción es aumentar las Service Quotas para AWS KMS. Para obtener más información, consulte [Cuotas](#) en la Guía para desarrolladores de AWS Key Management Service.

Para obtener información sobre la solución de problemas de permisos al utilizar Amazon S3 con Athena, consulte la sección [Permisos](#) del tema [Solución de problemas en Athena](#).

Permisos para metadatos cifrados en el catálogo de datos de AWS Glue

Si [cifra metadatos en AWS Glue Data Catalog](#), debe agregar acciones `"kms:GenerateDataKey"`, `"kms:Decrypt"` y `"kms:Encrypt"` a las políticas que utiliza para el acceso a Athena. Para obtener más información, consulte [Acceso desde Athena a metadatos cifrados en AWS Glue Data Catalog](#).

Cifrado de los resultados de las consultas de Athena en Amazon S3

El cifrado de resultados de las consultas se configura en la consola de Athena o cuando se utiliza JDBC u ODBC. Los grupos de trabajo le permiten aplicar el cifrado de los resultados de la consulta.

En la consola, puede ajustar la configuración para el cifrado de los resultados de las consultas de dos formas:

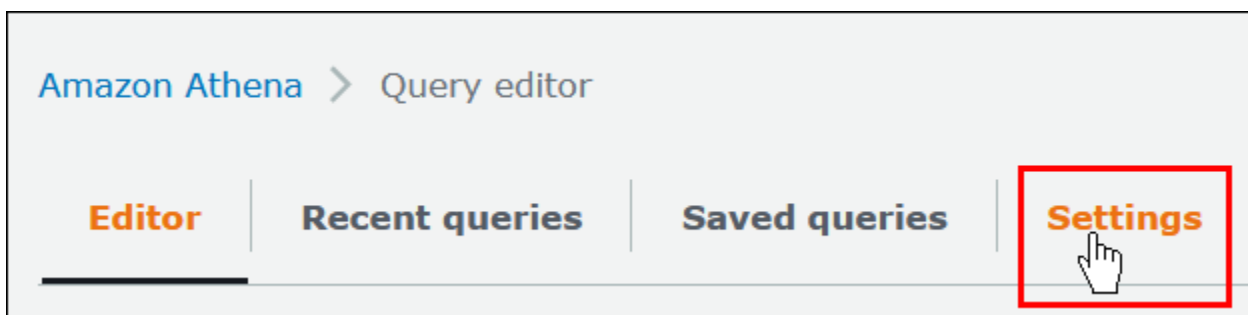
- Configuración del cliente: cuando se utiliza Configuración en la consola o las operaciones de la API para indicar que desea cifrar los resultados de las consultas, esto se conoce como uso de la configuración del cliente. La configuración del lado del cliente incluye la ubicación de los resultados de la consulta y cifrado. Si se especifican, se utilizan, a menos que se invalidan por la configuración del grupo de trabajo.
- Workgroup settings (Configuración de grupo de trabajo): al [crear o editar un grupo de trabajo](#) y seleccionar el campo Override client-side settings (Invalidar la configuración del lado del cliente), todas las consultas que se ejecutan en este grupo de trabajo utilizan el cifrado y la configuración de los resultados de las consultas del grupo de trabajo. Para obtener más información, consulte [La configuración del grupo de trabajo invalida la configuración del cliente](#).

Para cifrar resultados de la consulta almacenados en Amazon S3 mediante la consola

**⚠ Important**

Si su grupo de trabajo tiene seleccionado el campo Invalidar la configuración del cliente, todas las consultas en el grupo de trabajo utilizan la configuración del grupo de trabajo. No se utiliza la configuración de cifrado y la ubicación de los resultados de la consulta que se especifican en la pestaña Settings (Configuración) en la consola de Athena, por operaciones de la API y por controladores JDBC y ODBC. Para obtener más información, consulte [La configuración del grupo de trabajo invalida la configuración del cliente](#).

1. En la consola de Athena, elija Settings (Configuración).



2. Elija Administrar.
3. En Location of query result (Ubicación del resultado de la consulta), ingrese o elija una ruta de Amazon S3. Esta es la ubicación en Amazon S3 en la que se almacenan los resultados de las consultas.
4. Elija Encrypt query results (Cifrar resultados de la consulta).

Amazon Athena > Query editor > Manage settings

## Manage settings

### Query result location and encryption

Location of query result

[View](#)

[Browse S3](#)

**Encrypt query results**

### Encryption type

Choose server-side encryption (SSE) with an S3-managed encryption key (SSE-S3) or a customer master key (CMK) that you provide (SSE-KMS). Or choose client side encryption with a CMK (CSE-KMS).

CSE\_KMS

### Choose an AWS KMS key

This key will be used to encrypt and decrypt your resources. [Learn more](#)

[Create an AWS KMS key](#)




Cancel

Save

5. En Encryption type (Tipo de cifrado) elija CSE-KMS, SSE-KMS o SSE-S3. De estos tres, CSE-KMS ofrece el nivel más alto de cifrado, y SSE-S3 el más bajo.
6. Si eligió SSE-KMS o CSE-KMS. especifique una clave AWS KMS.

- En Elegir una clave AWS KMS, si la cuenta tiene acceso a una clave AWS KMS administrada por el cliente (CMK), elija su alias o ingrese el ARN de la clave AWS KMS.
- Si la cuenta no tiene acceso a una clave administrada por el cliente (CMK) existente, elija Crear clave de AWS KMS y luego abra la [consola de AWS KMS](#). Para obtener más información, consulte [Creación de claves](#) en la Guía para desarrolladores de AWS Key Management Service.

 Note

Athena admite solo claves simétricas para lectura y escritura de datos.

7. Vuelva a la consola de Athena y elija la clave creada por alias o ARN.
8. Seleccione Guardar.


### Cifrado de los resultados de la consulta de Athena cuando se utiliza JDBC u ODBC

Si se conecta utilizando el controlador JDBC u ODBC, configure las opciones del controlador para especificar el tipo de cifrado que desea utilizar y la ubicación del directorio de ensayo de Amazon S3. Para configurar el controlador JDBC u ODBC de modo que cifre los resultados de las consultas utilizando cualquiera de los protocolos de cifrado que admite Athena, consulte [Conexión a Amazon Athena con controladores ODBC y JDBC](#).

### Creación de tablas basadas en conjuntos de datos cifrados en Amazon S3

Cuando cree una tabla, indique a Athena que un conjunto de datos está cifrado en Amazon S3. Esto no es necesario cuando se utiliza SSE-KMS. Tanto para el cifrado SSE-S3 como para el cifrado AWS KMS, Athena determina los medios adecuados para descifrar el conjunto de datos y crear la tabla, por lo que no es necesario que proporcione información de clave.

Los usuarios que ejecutan consultas, incluido el usuario que crea la tabla, deben tener los permisos descritos anteriormente en este tema.

 Important

Si utiliza Amazon EMR junto con EMRFS para cargar archivos Parquet cifrados, debe desactivar las cargas multipart estableciendo `fs.s3n.multipart.uploads.enabled` el valor `false`. Si no lo hace, Athena no podrá determinar la longitud del archivo Parquet y se

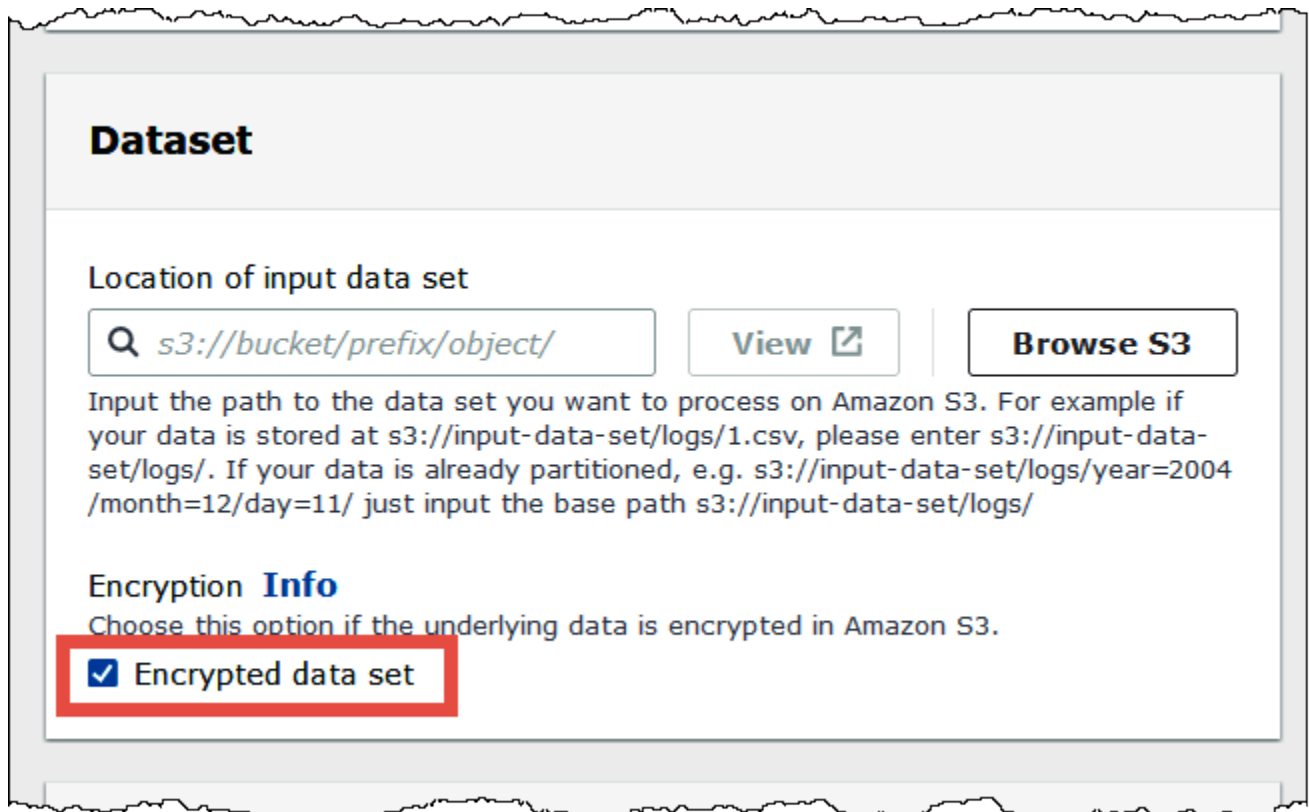
producirá un error `HIVE_CANNOT_OPEN_SPLIT`. Para obtener más información, consulte [Configuración de carga multiparte para Amazon S3](#) en la Guía de administración de Amazon EMR.

Para indicar que el conjunto de datos está cifrado en Amazon S3, realice uno de los pasos a continuación. Este paso no es necesario cuando se utiliza SSE-KMS.

- En una instrucción [CREATE TABLE](#) (CREAR TABLA), utilice una cláusula `TBLPROPERTIES` que especifica `'has_encrypted_data'='true'`, como en el siguiente ejemplo.

```
CREATE EXTERNAL TABLE 'my_encrypted_data' (  
  `n_nationkey` int,  
  `n_name` string,  
  `n_regionkey` int,  
  `n_comment` string)  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe'  
STORED AS INPUTFORMAT  
  'org.apache.hadoop.hive.q1.io.parquet.MapredParquetInputFormat'  
LOCATION  
  's3://DOC-EXAMPLE-BUCKET/folder_with_my_encrypted_data'  
TBLPROPERTIES (  
  'has_encrypted_data'='true')
```

- Utilice el [controlador JDBC](#) y establezca el valor `TBLPROPERTIES` como se indica en el ejemplo anterior cuando utilice `statement.executeQuery()` para ejecutar la instrucción [CREATE TABLE](#) (CREAR TABLA).
- Cuando utilice la consola de Athena para [crear una tabla mediante un formulario](#) y especificar la ubicación de la tabla, seleccione la opción Encrypted data set (Conjunto de datos cifrados).



## Dataset

### Location of input data set

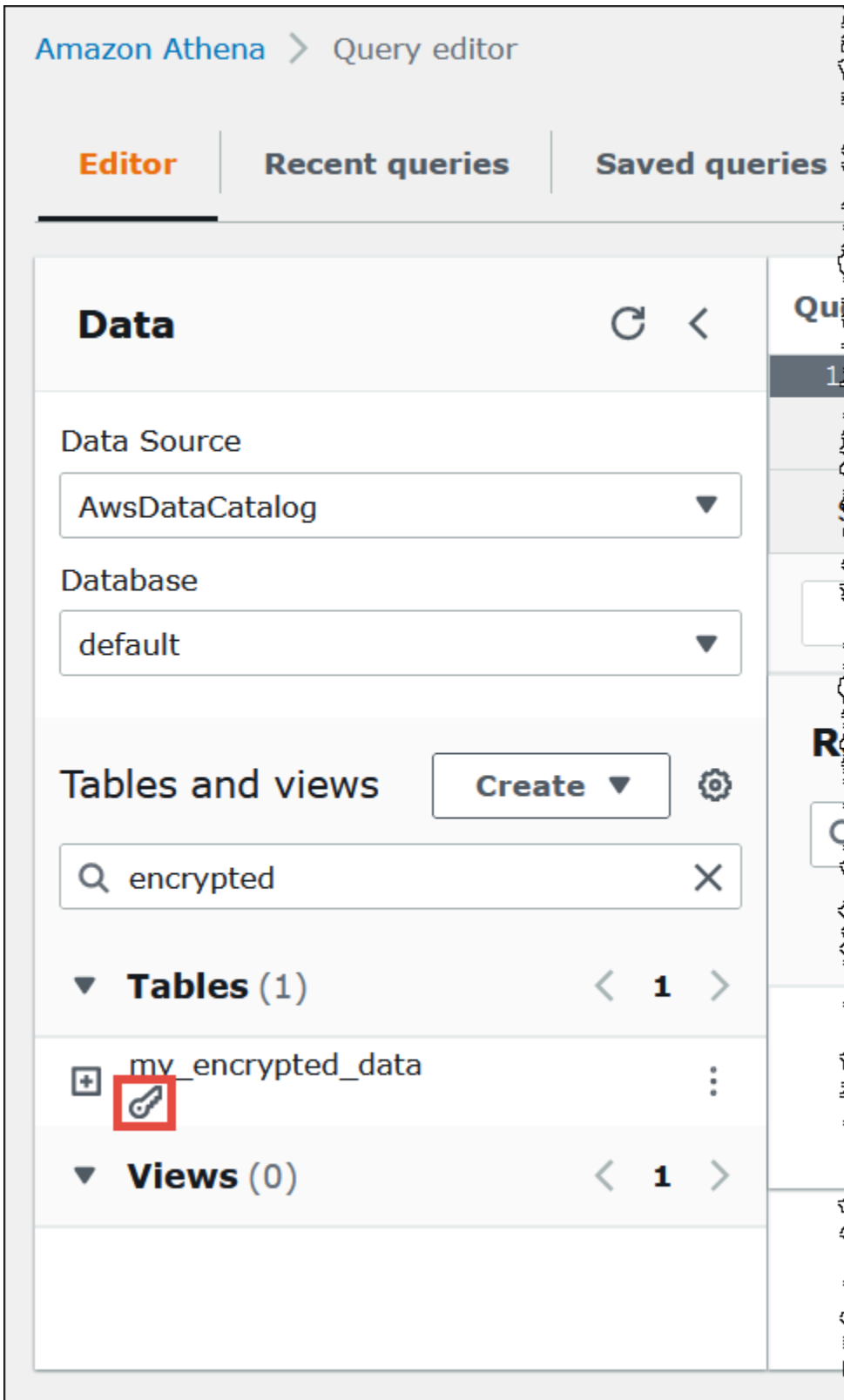
Input the path to the data set you want to process on Amazon S3. For example if your data is stored at `s3://input-data-set/logs/1.csv`, please enter `s3://input-data-set/logs/`. If your data is already partitioned, e.g. `s3://input-data-set/logs/year=2004/month=12/day=11/` just input the base path `s3://input-data-set/logs/`

### Encryption **Info**

Choose this option if the underlying data is encrypted in Amazon S3.

Encrypted data set

En la lista de tablas de la consola de Athena, las tablas cifradas muestran un icono con forma de llave.





## Cifrado en tránsito

Además de cifrar los datos en reposo en Amazon S3, Amazon Athena utiliza el cifrado Transport Layer Security (TLS) para los datos en tránsito entre Athena y Amazon S3, y entre Athena y las aplicaciones del cliente que obtienen acceso a ellos.

Solo debe permitir conexiones cifradas sobre HTTPS (TLS) utilizando [aws:SecureTransport condition](#) en las políticas de IAM del bucket de Amazon S3.

Los resultados de las consultas que se envían a clientes JDBC u ODBC se cifran mediante TLS. Para obtener información sobre las versiones más recientes de los controladores JDBC y ODBC y su documentación, consulte [Conexión a Amazon Athena con JDBC](#) y [Conexión a Amazon Athena con ODBC](#).

En el caso de los conectores de orígenes de datos federados de Athena, la compatibilidad con el cifrado en tránsito mediante TLS depende del conector individual. Para obtener más información, consulte la documentación de los [conectores de orígenes de datos](#) individuales.

## Administración de claves

Amazon Athena admite AWS Key Management Service (AWS KMS) para cifrar los conjuntos de datos en Amazon S3 y los resultados de consulta de Athena. AWS KMS utiliza claves administradas por el cliente (CMK) para cifrar los objetos de Amazon S3 y se basa en el [cifrado de sobres](#).

En AWS KMS, puede llevar a cabo las siguientes acciones:

- [Crear claves](#)
- [Importar su propio material de claves para las nuevas CMK](#)

### Note

Athena admite solo claves simétricas para lectura y escritura de datos.

Para obtener más información, consulte [¿Qué es AWS Key Management Service?](#) en la Guía para desarrolladores de AWS Key Management Service, y [Cómo Amazon Simple Storage Service utiliza AWS KMS](#). Si desea ver las claves de su cuenta que AWS crea y administra por usted, en el panel de navegación, elija Claves administradas por AWS.

Si carga o accede a objetos cifrados con SSE-KMS, use AWS Signature Version 4 para reforzar la seguridad. Para obtener más información, consulte [Especificación de Signature Version en la autenticación de solicitudes](#) en la Guía del usuario de Amazon Simple Storage Service.

Si sus cargas de trabajo de Athena cifran una gran cantidad de datos, puede utilizar claves de bucket de Amazon S3 para reducir costos. Para obtener más información, consulte [Reducción del costo de SSE-KMS con las claves de bucket de Amazon S3](#) en la Guía del usuario de Amazon Simple Storage Service.

## Privacidad del tráfico entre redes

El tráfico está protegido tanto entre Athena y las aplicaciones locales como entre Athena y Amazon S3. El tráfico entre Athena y otros servicios, como AWS Glue y AWS Key Management Service, utiliza HTTPS de forma predeterminada.

- Para el tráfico entre Athena y clientes y aplicaciones locales, los resultados de las consultas que se envían a clientes JDBC u ODBC se cifran mediante Transport Layer Security (TLS).

Puede utilizar una de las opciones de conectividad entre su red privada y AWS:

- Una conexión de Site-to-Site VPN de AWS VPN. Para obtener más información, consulte [¿Qué es Site-to-Site VPN AWS VPN?](#) en la Guía del usuario de AWS Site-to-Site VPN.
- Una conexión de AWS Direct Connect. Para obtener más información, consulte [¿Qué es AWS Direct Connect?](#) en la Guía del usuario de AWS Direct Connect.
- Para el tráfico entre Athena y buckets de Amazon S3, el protocolo Transport Layer Security (TLS) cifra los objetos en tránsito entre Athena y Amazon S3, y entre Athena y las aplicaciones de cliente que obtienen acceso a él, debe permitir solo las conexiones cifradas sobre HTTPS (TLS) utilizando [aws:SecureTransport condition](#) en las políticas de IAM del bucket de Amazon S3. Aunque Athena utiliza actualmente el punto de conexión público para acceder a los datos de los buckets de Amazon S3, esto no significa que los datos recorran la Internet pública. Todo el tráfico entre Athena y Amazon S3 se enruta a través de la red de AWS y se cifra mediante TLS.
- Programas de conformidad: Amazon Athena cumple con múltiples programas de conformidad de AWS, incluidos SOC, PCI, FedRAMP y otros. Para obtener más información, consulte [Servicios de AWS incluidos en el ámbito por programa de conformidad](#).

## Identity and Access Management en Athena

Amazon Athena utiliza [AWS Identity and Access Management \(IAM\)](#) para restringir el acceso a las operaciones de Athena. Para obtener una lista completa de los permisos de Athena, consulte [Acciones, recursos y claves de condición de Amazon Athena](#) en la Referencia de autorizaciones de servicio.

Siempre que utilice políticas de IAM, asegúrese de seguir las prácticas recomendadas de IAM. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM.

Entre los permisos necesarios para ejecutar las consultas de Athena se incluyen los siguientes:

- Ubicaciones de Amazon S3 donde se almacenan los datos subyacentes de la consulta. Para obtener más información, consulte [Identity and Access Management en Amazon S3](#) en la Guía del usuario de Amazon Simple Storage Service.
- Metadatos y recursos que se almacenan en AWS Glue Data Catalog, como bases de datos y tablas, incluidas acciones adicionales para metadatos cifrados. Para obtener más información, consulte [Configuración de permisos de IAM para AWS Glue](#) y [Configuración del cifrado en AWS Glue](#) en la Guía para desarrolladores de AWS Glue.
- Acciones de la API de Athena. Para obtener una lista completa de las acciones de la API en Athena, consulte [Acciones](#) en la Referencia de API de Amazon Athena.

Los siguientes temas proporcionan más información acerca de los permisos de áreas específicas de Athena.

### Temas

- [Políticas administradas de AWS para Amazon Athena](#)
- [Acceso a través de conexiones JDBC y ODBC](#)
- [Acceso a Amazon S3](#)
- [Acceso entre cuentas en Athena a los buckets de Simple Storage Service \(Amazon S3\)](#)
- [Acceso detallado a las bases de datos y las tablas en AWS Glue Data Catalog](#)
- [Acceso entre cuentas a los catálogos de datos de AWS Glue](#)
- [Acceso desde Athena a metadatos cifrados en AWS Glue Data Catalog](#)
- [Acceso a grupos de trabajo y etiquetas](#)

- [Permitir acceso a instrucciones preparadas](#)
- [Uso de Athena con claves de contexto CalledVia](#)
- [Permitir el acceso a un conector de datos de Athena para un almacén de metadatos externo de Hive](#)
- [Permitir a la función de Lambda el acceso a los almacenes de metadatos externos de Hive](#)
- [Ejemplos de políticas de permisos de IAM para permitir la consulta federada de Athena](#)
- [Ejemplos de políticas de permisos de IAM para permitir funciones definidas por el usuario \(UDF\) de Amazon Athena](#)
- [Permiso de acceso para ML con Athena](#)
- [Habilitación de acceso federado a la API de Athena](#)

## Políticas administradas de AWS para Amazon Athena

Una política administrada de AWS es una política independiente que AWS crea y administra. Las políticas administradas de AWS se diseñan para ofrecer permisos para muchos casos de uso comunes, por lo que puede empezar a asignar permisos a los usuarios, grupos y roles.

Considere que es posible que las políticas administradas por AWS no concedan permisos de privilegio mínimo para los casos de uso concretos, ya que están disponibles para que las utilicen todos los clientes de AWS. Se recomienda definir [políticas administradas por el cliente](#) específicas para sus casos de uso a fin de reducir aún más los permisos.

No puede cambiar los permisos definidos en las políticas administradas de AWS. Si AWS actualiza los permisos definidos en una política administrada de AWS, la actualización afecta a todas las identidades de entidades principales (usuarios, grupos y roles) a las que está adjunta la política. Lo más probable es que AWS actualice una política administrada de AWS cuando se lance un nuevo Servicio de AWS o las operaciones de la API nuevas estén disponibles para los servicios existentes.

Para obtener más información, consulte [Políticas administradas por AWS](#) en la Guía del usuario de IAM.

### Consideraciones al utilizar políticas administradas con Athena

Las políticas administradas son fáciles de utilizar y se actualizan automáticamente con las acciones necesarias a medida que el servicio evoluciona. Cuando utilice políticas administradas con Athena, tenga en cuenta los siguientes puntos:

- Para permitir o denegar acciones de servicio de Amazon Athena para usted u otros usuarios mediante AWS Identity and Access Management (IAM), asocie políticas basadas en identidad a entidades principales, como usuarios o grupos.
- Cada política basada en identidad se compone de instrucciones que definen las acciones que se permiten o se deniegan. Para obtener más información e instrucciones paso a paso para asociar una política a un usuario, consulte [Asociar políticas administradas](#) en la Guía del usuario de IAM. Para obtener una lista de acciones, consulte la sección de [referencia de API de Amazon Athena](#).
- Las políticas basadas en identidad administradas por el cliente e insertadas le permiten especificar acciones de Athena más detalladas para dar más precisión al acceso. Le recomendamos que utilice la política AmazonAthenaFullAccess como punto de partida y, a continuación, permita o deniegue acciones específicas que se muestran en la sección de [referencia de la API de Amazon Athena](#). Para obtener más información sobre las políticas insertadas, consulte [Políticas administradas e insertadas](#) en la Guía del usuario de IAM.
- Si también tiene entidades principales que se conectan a través de JDBC, debe proporcionar al controlador JDBC las credenciales para el acceso a su aplicación. Para obtener más información, consulte [Acceso a través de conexiones JDBC y ODBC](#).
- Si ha cifrado el catálogo de datos de AWS Glue, debe especificar acciones adicionales en las políticas de IAM basadas en identidad para Athena. Para obtener más información, consulte [Acceso desde Athena a metadatos cifrados en AWS Glue Data Catalog](#).
- Si crea y utiliza grupos de trabajo, asegúrese de que sus políticas incluyan el acceso correspondiente a las acciones de los grupos de trabajo. Para obtener más información, consulte [the section called “Políticas de IAM para acceder a los grupos de trabajo”](#) y [the section called “Ejemplos de políticas de grupos de trabajo”](#).

## Política administrada de AWS: AmazonAthenaFullAccess

La política administrada de AmazonAthenaFullAccess concede acceso total a Athena.

Para proporcionar acceso, agregue permisos a sus usuarios, grupos o roles:

- Usuarios y grupos en AWS IAM Identity Center:

Cree un conjunto de permisos. Siga las instrucciones descritas en [Crear un conjunto de permisos](#) en la Guía del usuario de AWS IAM Identity Center.

- Usuarios administrados en IAM a través de un proveedor de identidades:

Cree un rol para la federación de identidades. Siga las instrucciones descritas en [Creación de un rol para un proveedor de identidad de terceros \(federación\)](#) en la Guía del usuario de IAM.

- Usuarios de IAM:
  - Cree un rol que el usuario pueda aceptar. Siga las instrucciones descritas en [Creación de un rol para un usuario de IAM](#) en la Guía del usuario de IAM.
  - (No recomendado) Adjunte una política directamente a un usuario o añada un usuario a un grupo de usuarios. Siga las instrucciones de [Adición de permisos a un usuario \(consola\)](#) de la Guía del usuario de IAM.

## Agrupaciones de permisos

La política AmazonAthenaFullAccess se agrupa en los siguientes conjuntos de permisos.

- **athena**: permite el acceso de las entidades principales a los recursos de Athena.
- **glue**: permite el acceso de las entidades principales a bases de datos, tablas y particiones de AWS Glue. Esto es necesario para que la entidad principal pueda utilizar AWS Glue Data Catalog con Athena.
- **s3**: permite a la entidad principal escribir y leer los resultados de las consultas de Amazon S3, leer ejemplos de datos de Athena disponibles públicamente que residen en Amazon S3 y listar buckets. Esto es necesario para que la entidad principal pueda utilizar Athena para trabajar con Amazon S3.
- **sns**: permite a las entidades principales enumerar temas de Amazon SNS y obtener atributos de temas. Esto permite a las entidades principales utilizar temas de Amazon SNS con Athena para fines de monitoreo y alerta.
- **cloudwatch**: permite a las entidades principales crear, leer y eliminar alarmas de CloudWatch. Para obtener más información, consulte [Control de costos y supervisión de consultas con métricas y eventos de CloudWatch](#).
- **lakeformation**: permite a las entidades principales solicitar credenciales temporales para acceder a los datos en una ubicación de lago de datos registrada con Lake Formation. Para obtener más información, consulte [Control de acceso a datos subyacentes](#) en la Guía para desarrolladores de AWS Lake Formation.
- **datazone**: permite a las entidades principales enumerar proyectos, dominios y entornos de Amazon DataZone. Para obtener información acerca del uso de DataZone en Athena, consulte [Uso de Amazon DataZone en Athena](#).

- **pricing**: proporciona acceso a AWS Billing and Cost Management. Para obtener más información, consulte [GetProducts](#) en la Referencia de la API de AWS Billing and Cost Management.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BaseAthenaPermissions",
      "Effect": "Allow",
      "Action": [
        "athena:*"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "BaseGluePermissions",
      "Effect": "Allow",
      "Action": [
        "glue:CreateDatabase",
        "glue>DeleteDatabase",
        "glue:GetDatabase",
        "glue:GetDatabases",
        "glue:UpdateDatabase",
        "glue:CreateTable",
        "glue>DeleteTable",
        "glue:BatchDeleteTable",
        "glue:UpdateTable",
        "glue:GetTable",
        "glue:GetTables",
        "glue:BatchCreatePartition",
        "glue:CreatePartition",
        "glue>DeletePartition",
        "glue:BatchDeletePartition",
        "glue:UpdatePartition",
        "glue:GetPartition",
        "glue:GetPartitions",
        "glue:BatchGetPartition",
        "glue:StartColumnStatisticsTaskRun",
        "glue:GetColumnStatisticsTaskRun",

```

```

        "glue:GetColumnStatisticsTaskRuns"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid": "BaseQueryResultsPermissions",
    "Effect": "Allow",
    "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload",
        "s3:CreateBucket",
        "s3:PutObject",
        "s3:PutBucketPublicAccessBlock"
    ],
    "Resource": [
        "arn:aws:s3:::aws-athena-query-results-*"
    ]
},
{
    "Sid": "BaseAthenaExamplesPermissions",
    "Effect": "Allow",
    "Action": [
        "s3:GetObject",
        "s3:ListBucket"
    ],
    "Resource": [
        "arn:aws:s3:::athena-examples*"
    ]
},
{
    "Sid": "BaseS3BucketPermissions",
    "Effect": "Allow",
    "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets"
    ],
    "Resource": [

```



```
        "*"
    ]
},
{
    "Sid": "BaseSNSPermissions",
    "Effect": "Allow",
    "Action": [
        "sns:ListTopics",
        "sns:GetTopicAttributes"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid": "BaseCloudWatchPermissions",
    "Effect": "Allow",
    "Action": [
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:DescribeAlarms",
        "cloudwatch>DeleteAlarms",
        "cloudwatch:GetMetricData"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid": "BaseLakeFormationPermissions",
    "Effect": "Allow",
    "Action": [
        "lakeformation:GetDataAccess"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid": "BaseDataZonePermissions",
    "Effect": "Allow",
    "Action": [
        "datazone:ListDomains",
        "datazone:ListProjects",
        "datazone:ListAccountEnvironments"
    ]
}
```

```

    ],
    "Resource": [
        "*"
    ]
  },
  {
    "Sid": "BasePricingPermissions",
    "Effect": "Allow",
    "Action": [
        "pricing:GetProducts"
    ],
    "Resource": [
        "*"
    ]
  }
]
}

```

## Política administrada de AWS: AWSQuicksightAthenaAccess

`AWSQuicksightAthenaAccess` concede acceso a las acciones que Amazon QuickSight requiere para integrarse con Athena. Puede adjuntar la política `AWSQuicksightAthenaAccess` a las identidades de IAM. Asocie esta política solo a entidades principales que utilicen Amazon QuickSight con Athena. Esta política incluye algunas acciones para Athena que han quedado obsoletas y no se encuentran en la API pública actual o que se utilizan únicamente con los controladores JDBC y ODBC.

### Grupos de permisos

La política `AWSQuicksightAthenaAccess` se agrupa en los siguientes conjuntos de permisos.

- **athena**: permite que la entidad principal ejecute consultas sobre los recursos de Athena.
- **glue**: permite el acceso de las entidades principales a bases de datos, tablas y particiones de AWS Glue. Esto es necesario para que la entidad principal pueda utilizar AWS Glue Data Catalog con Athena.
- **s3**: permite a la entidad principal escribir y leer los resultados de las consultas de Amazon S3.
- **lakeformation**: permite a las entidades principales solicitar credenciales temporales para acceder a los datos en una ubicación de lago de datos registrada con Lake Formation. Para obtener más información, consulte [Control de acceso a datos subyacentes](#) en la Guía para desarrolladores de AWS Lake Formation.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:BatchGetQueryExecution",
        "athena:GetQueryExecution",
        "athena:GetQueryResults",
        "athena:GetQueryResultsStream",
        "athena:ListQueryExecutions",
        "athena:StartQueryExecution",
        "athena:StopQueryExecution",
        "athena:ListWorkGroups",
        "athena:ListEngineVersions",
        "athena:GetWorkGroup",
        "athena:GetDataCatalog",
        "athena:GetDatabase",
        "athena:GetTableMetadata",
        "athena:ListDataCatalogs",
        "athena:ListDatabases",
        "athena:ListTableMetadata"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "glue:CreateDatabase",
        "glue>DeleteDatabase",
        "glue:GetDatabase",
        "glue:GetDatabases",
        "glue:UpdateDatabase",
        "glue:CreateTable",
        "glue>DeleteTable",
        "glue:BatchDeleteTable",
        "glue:UpdateTable",
        "glue:GetTable",
        "glue:GetTables",
        "glue:BatchCreatePartition",
        "glue:CreatePartition",

```

```

        "glue:DeletePartition",
        "glue:BatchDeletePartition",
        "glue:UpdatePartition",
        "glue:GetPartition",
        "glue:GetPartitions",
        "glue:BatchGetPartition"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload",
        "s3:CreateBucket",
        "s3:PutObject",
        "s3:PutBucketPublicAccessBlock"
    ],
    "Resource": [
        "arn:aws:s3:::aws-athena-query-results-*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "lakeformation:GetDataAccess"
    ],
    "Resource": [
        "*"
    ]
}
]
}

```

## Actualizaciones de Athena en Políticas administradas de AWS

Es posible consultar los detalles sobre las actualizaciones de las políticas administradas de AWS para Athena debido a que este servicio comenzó a realizar el seguimiento de estos cambios.

Cambio	Descripción	Fecha
<a href="#">AmazonAthenaFullAccess:</a> actualización de la política existente	Se agregaron los permisos <code>datazone:ListDomains</code> , <code>datazone:ListProjects</code> y <code>datazone:ListAccountEnvironments</code> para permitir a los usuarios de Athena trabajar con dominios, proyectos y entornos de Amazon DataZone. Para obtener más información, consulte <a href="#">Uso de Amazon DataZone en Athena</a> .	3 de enero de 2024
<a href="#">AmazonAthenaFullAccess:</a> actualización de la política existente	Se agregaron los permisos <code>glue:StartColumnStatisticsTaskRun</code> , <code>glue:GetColumnStatisticsTaskRun</code> y <code>glue:GetColumnStatisticsTaskRuns</code> para dar a Athena el derecho a llamar a AWS Glue para recuperar estadísticas de la característica de optimización basada en costos. Para obtener más información, consulte <a href="#">Uso del optimizador basado en costes</a> .	3 de enero de 2024
<a href="#">AmazonAthenaFullAccess:</a> actualización de la política existente	Athena agregó <code>pricing:GetProducts</code> para proporcionar acceso a AWS Billing and Cost Management. Para obtener	25 de enero de 2023

Cambio	Descripción	Fecha
	más información, consulte <a href="#">GetProducts</a> en la Referencia de la API de AWS Billing and Cost Management.	
<a href="#">AmazonAthenaFullAccess</a> : actualización de la política existente	Athena agregó <code>cloudwatch:GetMetricData</code> para recuperar los valores de métricas de CloudWatch. Para obtener más información, consulte <a href="#">GetMetricData</a> en la Referencia de la API de los Registros de Amazon CloudWatch.	14 de noviembre de 2022
<a href="#">AmazonAthenaFullAccess</a> y <a href="#">AWSQuicksightAthenaAccess</a> : actualizaciones de las políticas existentes	Athena agregó <code>s3:PutBucketPublicAccessBlock</code> para permitir el bloqueo del acceso público en los buckets creados por Athena.	7 de julio de 2021
Athena comenzó a realizar el seguimiento de los cambios	Athena comenzó a realizar el seguimiento de los cambios de las políticas administradas de AWS.	7 de julio de 2021

## Acceso a través de conexiones JDBC y ODBC

Para obtener acceso a los recursos y Servicios de AWS como Athena y los buckets de Amazon S3, proporcione credenciales de controlador JDBC u ODBC a su aplicación. Si utiliza el controlador JDBC u ODBC, asegúrese de que la política de permisos de IAM incluya todas las acciones enumeradas en la Política administrada [Política administrada de AWS: AWSQuicksightAthenaAccess](#).

Siempre que utilice políticas de IAM, compruebe que sigue las mejores prácticas IAM. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM.

## Métodos de autenticación

Los controladores JDBC y ODBC de Athena admiten la autenticación basada en SAML 2.0, incluidos los siguientes proveedores de identidad:

- Active Directory Federation Services (AD FS)
- Azure Active Directory (AD)
- Okta
- PingFederate

Para obtener más información, consulte las guías de instalación y configuración de los controladores respectivos, que se pueden descargar en formato PDF desde las páginas de controladores [JDBC](#) y [ODBC](#). Para obtener más información relacionada, consulte los siguientes temas:

- [Habilitación de acceso federado a la API de Athena](#)
- [Uso de Lake Formation y los controladores JDBC y ODBC de Athena para el acceso federado a Athena](#)
- [Configuración del inicio de sesión único con ODBC, SAML 2.0 y el proveedor de identidades Okta](#)

Para obtener información sobre las versiones más recientes de los controladores JDBC y ODBC y su documentación, consulte [Conexión a Amazon Athena con JDBC](#) y [Conexión a Amazon Athena con ODBC](#).

## Acceso a Amazon S3

Puede conceder acceso a las ubicaciones de Amazon S3 mediante políticas basadas en identidad, políticas de recursos de bucket, políticas de puntos de acceso o cualquier combinación de las anteriores. Cuando los actores interactúan con Athena, sus permisos pasan por Athena para determinar a qué puede acceder Athena. Esto significa que los usuarios deben tener permiso para acceder a los buckets de Amazon S3 para poder consultarlos con Athena.

Siempre que utilice políticas de IAM, compruebe que sigue las mejores prácticas IAM. Para más información, consulte [Prácticas recomendadas de seguridad en IAM](#) en la Guía del usuario de IAM.

Cuando se configura `aws:SourceIp` en las políticas, Athena accede al bucket de Amazon S3 mediante la dirección IP que se especifique. No se puede restringir ni permitir el acceso a los recursos de Amazon S3 en función de las claves de condición `aws:SourceVpc` o `aws:SourceVpce`.

**Note**

Los grupos de trabajo de Athena que utilizan la autenticación de IAM Identity Center requieren que S3 Access Grants esté configurado para utilizar identidades de propagación fiables. Para obtener más información, consulte [S3 Access Grants e identidades de directorio](#) en la Guía del usuario de Amazon Simple Storage Service.

## Puntos de acceso y alias de puntos de acceso de Simple Storage Service (Amazon S3)

Si tiene un conjunto de datos compartido en un bucket de Amazon S3, puede resultar difícil mantener una política de bucket única que administre el acceso para cientos de casos de uso.

Los puntos de acceso de bucket de Amazon S3 ayudan a resolver este problema. Un bucket puede tener varios puntos de acceso, cada uno con una política que controla el acceso al bucket de forma diferente.

Para cada punto de acceso que cree, Amazon S3 genera un alias que representa el punto de acceso. Debido a que el alias está en formato de nombre de bucket de Amazon S3, puede utilizar el alias de la cláusula LOCATION de sus instrucciones CREATE TABLE en Athena. La política para el punto de acceso que representa el alias controla entonces el acceso de Athena al bucket.

Para obtener más información, consulte [Ubicación de las tablas en Simple Storage Service \(Amazon S3\)](#) y [Cómo utilizar los puntos de acceso](#) en la Guía del usuario de Amazon S3.

## Uso de claves de contexto CalledVia

Para mayor seguridad, puede utilizar la clave de contexto de condición global [aws:CalledVia](#). La clave `aws:CalledVia` contiene una lista ordenada de cada servicio de la cadena que realizó solicitudes en nombre de la entidad principal. Al especificar el nombre de la entidad principal del servicio de Athena `athena.amazonaws.com` para clave de contexto `aws:CalledVia`, puede limitar las solicitudes a solo las hechas desde Athena. Para obtener más información, consulte [Uso de Athena con claves de contexto CalledVia](#).

## Recursos adicionales

Para obtener información detallada y ejemplos acerca de cómo conceder acceso a Amazon S3, consulte los siguientes recursos:

- [Ejemplos de explicaciones: Administración de acceso](#) en la Guía del usuario de Simple Storage Service (Amazon S3).



- [¿Cómo puedo proporcionar acceso entre cuentas a objetos que se encuentran en buckets de Amazon S3?](#) en el Centro de conocimientos de AWS.
- [Acceso entre cuentas en Athena a los buckets de Simple Storage Service \(Amazon S3\)](#).

## Acceso entre cuentas en Athena a los buckets de Simple Storage Service (Amazon S3)

Es habitual en Amazon Athena que se conceda acceso a usuarios de una cuenta que no sea la cuenta del propietario del bucket, a fin de que dichos usuarios puedan realizar consultas. En este caso utilice una política de bucket para conceder acceso.

### Note

Para obtener información sobre el acceso entre cuentas a catálogos de datos de AWS Glue desde Athena, consulte [Acceso entre cuentas a los catálogos de datos de AWS Glue](#).

La siguiente política de bucket de ejemplo, creada y aplicada al bucket `s3://DOC-EXAMPLE-BUCKET` por el propietario del bucket, concede acceso a todos los usuarios de la cuenta `123456789123`, que es una cuenta diferente.

```
{
  "Version": "2012-10-17",
  "Id": "MyPolicyID",
  "Statement": [
    {
      "Sid": "MyStatementSid",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789123:root"
      },
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload"
      ],
      "Resource": [
```

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

Para conceder acceso a un usuario determinado de una cuenta, reemplace la clave `Principal` por una clave que especifique al usuario en lugar de `root`. Por ejemplo, para el perfil de usuario Dave utilice `arn:aws:iam::123456789123:user/Dave`.

### Acceso entre cuentas a un bucket cifrado con una clave de AWS KMS personalizada

Si tiene un bucket de Amazon S3 cifrado con una clave AWS Key Management Service (AWS KMS) personalizada, es posible que deba conceder acceso a él a los usuarios desde otra cuenta de Amazon Web Services.

La concesión de acceso a un bucket AWS KMS cifrado en la cuenta A a un usuario de la cuenta B requiere los siguientes permisos:

- La política de bucket de la cuenta A debe conceder acceso al rol asumido por la cuenta B.
- La política de claves de AWS KMS de la cuenta A debe conceder acceso al rol asumido por el usuario de la cuenta B.
- El rol de AWS Identity and Access Management (IAM) asumido por la cuenta B debe conceder acceso tanto al bucket como a la clave de la cuenta A.

En los siguientes procedimientos se describe cómo conceder cada uno de estos permisos.

Para conceder acceso al bucket de la cuenta A al usuario de la cuenta B

- En la cuenta A, [revise la política de bucket de S3](#) y confirme que existe una declaración que permite el acceso desde el ID de cuenta de la cuenta B.

Por ejemplo, la siguiente política de bucket permite el acceso `s3:GetObject` al ID de cuenta `111122223333`:

```
{
  "Id": "ExamplePolicy1",
  "Version": "2012-10-17",
  "Statement": [
```

```

{
  "Sid": "ExampleStmnt1",
  "Action": [
    "s3:GetObject"
  ],
  "Effect": "Allow",
  "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
  "Principal": {
    "AWS": [
      "111122223333"
    ]
  }
}
]
}

```

Para conceder acceso al usuario de la cuenta B desde la política clave AWS KMS de la cuenta A

1. En la política de claves de AWS KMS de la cuenta A, conceda al rol asumido por la cuenta B permisos para las siguientes acciones:
  - kms:Encrypt
  - kms:Decrypt
  - kms:ReEncrypt\*
  - kms:GenerateDataKey\*
  - kms:DescribeKey

En el siguiente ejemplo se concede acceso de clave a un solo rol de IAM.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowUseOfTheKey",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/role_name"
      },
      "Action": [

```

```

        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
]
}

```

2. En la cuenta A, revise la política de claves [mediante la vista de políticas de la AWS Management Console](#).
3. En la política clave, compruebe que la siguiente instrucción muestra la cuenta B como entidad principal.

```
"Sid": "Allow use of the key"
```

4. Si la instrucción "Sid": "Allow use of the key" no está presente, lleve a cabo los siguientes pasos:
  - a. Cambie para ver la política clave [mediante la vista predeterminada de la consola](#).
  - b. Añada el ID de cuenta de la cuenta B como una cuenta externa con acceso a la clave.

Para conceder acceso al bucket y a la clave de la cuenta A desde el rol de IAM asumido por la cuenta B

1. Desde la cuenta B, abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. Abra el rol de IAM asociado al usuario en la cuenta B.
3. Revise la lista de políticas de permisos aplicadas al rol de IAM.
4. Asegúrese de que se aplica una política que conceda acceso al bucket.

La siguiente instrucción de ejemplo concede al rol de IAM acceso a las operaciones s3:GetObject y s3:PutObject en el bucket DOC-EXAMPLE-BUCKET:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {

```

```

    "Sid": "ExampleStmt2",
    "Action": [
      "s3:GetObject",
      "s3:PutObject"
    ],
    "Effect": "Allow",
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
  }
]
}

```

5. Asegúrese de que se aplica una política que conceda acceso a la clave.

#### Note

Si el rol de IAM de la cuenta B ya tiene [acceso de administrador](#), no es necesario conceder acceso a la clave desde las políticas de IAM del usuario.

La siguiente instrucción de ejemplo concede al rol de IAM acceso para utilizar la clave `arn:aws:kms:us-west-2:123456789098:key/111aa2bb-333c-4d44-5555-a111bb2c33dd`.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ExampleStmt3",
      "Action": [
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:Encrypt",
        "kms:GenerateDataKey",
        "kms:ReEncrypt*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:kms:us-west-2:123456789098:key/111aa2bb-333c-4d44-5555-a111bb2c33dd"
    }
  ]
}

```

## Acceso entre cuentas a objetos de bucket

Los objetos cargados por una cuenta (cuenta C) distinta de la cuenta propietaria del bucket (cuenta A) pueden requerir ACL explícitas de nivel de objeto que concedan acceso de lectura a la cuenta que realiza la consulta (cuenta B). Para evitar este requisito, la cuenta C debe asumir un rol en la cuenta A antes de colocar objetos en el bucket de la cuenta A. Para obtener más información, consulte [¿Cómo puedo proporcionar acceso entre cuentas a objetos que se encuentran en buckets de Amazon S3?](#).

## Acceso detallado a las bases de datos y las tablas en AWS Glue Data Catalog

Si utiliza la opción AWS Glue Data Catalog con Amazon Athena, puede definir políticas de nivel de recursos para la base de datos y los objetos del catálogo de datos que se utilizan en Athena.

### Note

El término “fine-grained access control” (control de acceso detallado) aquí se refiere a la seguridad a nivel de base de datos y tabla. Para obtener información acerca de la seguridad a nivel de celda, fila y columna, consulte [Filtrado de datos y seguridad a nivel de celda en Lake Formation](#).

Puede definir los permisos de nivel de recursos en las políticas basadas en identidad de IAM.

### Important

En esta sección se analizan los permisos de nivel de recursos en las políticas basadas en identidad de IAM identity-based. Son diferentes de las políticas basadas en recursos. Para obtener más información sobre las diferencias, consulte [Políticas basadas en identidad y políticas basadas en recursos](#) en la Guía del usuario de IAM.

Consulte los siguientes temas sobre estas tareas:

Para llevar a cabo esta tarea	consulte el siguiente tema
Creación de una política de IAM que defina el	<a href="#">Creación de políticas de IAM</a> en la Guía del usuario de IAM.

Para llevar a cabo esta tarea	consulte el siguiente tema
acceso detallado a los recursos	
Más información acerca de las políticas basadas en identidad de IAM que se utilizan en AWS Glue	<a href="#">Políticas basadas en identidad (políticas de IAM)</a> en la Guía para desarrolladores de AWS Glue.

En esta sección

- [Limitaciones](#)
- [Acceso de AWS Glue a su catálogo y base de datos por Región de AWS](#)
- [Particiones y versiones de tabla en AWS Glue](#)
- [Ejemplos de permisos detallados para tablas y bases de datos](#)

## Limitaciones

Tenga en cuenta las siguientes limitaciones cuando use un control de acceso detallado con AWS Glue Data Catalog y Athena:

- Los grupos de trabajo de Athena habilitados para IAM Identity Center requieren que Lake Formation esté configurado para utilizar las identidades del IAM Identity Center. Para obtener más información, consulte [Integración del IAM Identity Center](#) en la Guía para desarrolladores de AWS Lake Formation.
- El acceso solo puede limitarse a bases de datos y tablas. Los controles de acceso detallado se aplican en el nivel de tabla y no puede limitar el acceso a las distintas particiones de una tabla. Para obtener más información, consulte [Particiones y versiones de tabla en AWS Glue](#).
- El AWS Glue Data Catalog contiene los siguientes recursos: CATALOG, DATABASE, TABLE y FUNCTION.

### Note

En esta lista, los recursos que son comunes entre Athena y AWS Glue Data Catalog son TABLE, DATABASE y CATALOG para cada cuenta. Function es específico de AWS Glue.

Para eliminar acciones en Athena, debe incluir los permisos a las acciones de AWS Glue. Consulte [Ejemplos de permisos detallados para tablas y bases de datos](#).

La jerarquía es la siguiente: CATALOG es un antecesor de todas las DATABASES de cada cuenta y cada DATABASE es un antecesor de todas sus TABLES y FUNCTIONS. Por ejemplo, en el caso de una tabla denominada `table_test` que pertenezca a una base de datos `db` del catálogo de su cuenta, sus antecesores son `db` y el catálogo de su cuenta. Para la base de datos `db`, su antecesor es el catálogo de su cuenta y sus descendientes son las tablas y funciones. Para obtener más información acerca de la estructura jerárquica de los recursos, consulte [Lista de ARN del catálogo de datos](#) en la Guía para desarrolladores de AWS Glue.

- Para cualquier acción de Athena en un registro distinta a la operación de eliminación, como `CREATE DATABASE`, `CREATE TABLE`, `SHOW DATABASE`, `SHOW TABLE` o `ALTER TABLE`, necesita permisos para llamar a esta acción en el recurso (tabla o base de datos) y en todos los antecesores de ese recurso en el catálogo de datos. Por ejemplo, en el caso de una tabla, sus antecesores son la base de datos a la que pertenece y el catálogo de la cuenta. En el caso de una base de datos, su antecesor es el catálogo de la cuenta. Consulte [Ejemplos de permisos detallados para tablas y bases de datos](#).
- Para una acción de eliminación en Athena, como `DROP DATABASE` o `DROP TABLE`, también necesita permisos para llamar a la acción de eliminación en todos los antecesores y descendientes del recurso en el catálogo de datos. Por ejemplo, para eliminar una base de datos necesita permisos en la base de datos, el catálogo, que es su antecesor, y todas las tablas y funciones definidas por el usuario, que son sus descendientes. Una tabla no tiene descendientes. Para ejecutar `DROP TABLE`, necesita permisos para esta acción en la tabla, la base de datos a la que pertenece y el catálogo. Consulte [Ejemplos de permisos detallados para tablas y bases de datos](#).

## Acceso de AWS Glue a su catálogo y base de datos por Región de AWS

Para que Athena pueda trabajar con AWS Glue, se requiere una política que conceda el acceso a su base de datos y al AWS Glue Data Catalog en su cuenta por Región de AWS. Para crear bases de datos, también se requiere el permiso `CreateDatabase`. En el siguiente ejemplo de política, sustituya la Región de AWS, el ID de la Cuenta de AWS y el nombre de la base de datos por los suyos.

```
{
  "Sid": "DatabasePermissions",
  "Effect": "Allow",
```



```
"Action": [  
  "glue:GetDatabase",  
  "glue:GetDatabases",  
  "glue:CreateDatabase"  
],  
"Resource": [  
  "arn:aws:glue:us-east-1:123456789012:catalog",  
  "arn:aws:glue:us-east-1:123456789012:database/default"  
]  
}
```

## Particiones y versiones de tabla en AWS Glue

En AWS Glue, las tablas pueden tener particiones y versiones. Las versiones y particiones de tabla no se consideran recursos independientes en AWS Glue. El acceso a las versiones y particiones de tabla se otorga concediendo acceso a la tabla y a los recursos antecesores de la tabla.

Para un control de acceso detallado, se aplican los siguientes permisos de acceso:

- Los controles de acceso detallado se aplican en el nivel de tabla. El acceso solo puede limitarse a bases de datos y tablas. Por ejemplo, si permite el acceso a una tabla particionada, este acceso se aplica a todas las particiones de la tabla. No se puede limitar el acceso a particiones individuales dentro de una tabla.

### Important

A fin de ejecutar acciones de AWS Glue en las particiones, se requieren permisos para las acciones de partición en los niveles de catálogo, base de datos y tabla. No basta con tener acceso a las particiones de una tabla. Por ejemplo, para ejecutar `GetPartitions` en la tabla `myTable` en la base de datos `myDB`, debe conceder permisos para `glue:GetPartitions` al catálogo, la base de datos `myDB` y los recursos `myTable`.

- Los controles de acceso detallado no se aplican a las versiones de tabla. Al igual que ocurre con las particiones, el acceso a las versiones anteriores de una tabla se concede mediante el acceso a las API de versión de tabla de AWS Glue en la tabla y en los antecesores de la tabla.

Para obtener información sobre los permisos para las acciones de AWS Glue, consulte [Permisos de API de AWS Glue: referencia de recursos y acciones](#) en la Guía para desarrolladores de AWS Glue.

## Ejemplos de permisos detallados para tablas y bases de datos

La siguiente tabla contiene ejemplos de políticas basadas en identidad de IAM que permiten un acceso detallado a las bases de datos y tablas de Athena. Le sugerimos que comience con estos ejemplos y, en función de sus necesidades, los ajuste para permitir o denegar acciones específicas con bases de datos y tablas determinadas.

Estos ejemplos incluyen el acceso a bases de datos y catálogos para que Athena y AWS Glue puedan trabajar juntos. En el caso de trabajar en varias regiones de AWS, incluya políticas similares para cada una de las bases de datos y catálogos, con una línea por cada región.

En los ejemplos, sustituya la base de datos `example_db` y la tabla `test` por sus propios nombres de base de datos y tabla.

Instrucción DDL	Ejemplo de una política de acceso de IAM que concede acceso al recurso
ALTER DATABASE	<p>Permite modificar las propiedades de la base de datos <code>example_db</code> .</p> <pre data-bbox="503 961 1507 1478"> {   "Effect": "Allow",   "Action": [     "glue:GetDatabase",     "glue:UpdateDatabase"   ],   "Resource": [     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :catalog",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :database / <i>example_db</i> "   ] } </pre>
CREATE DATABASE	<p>Permite crear la base de datos con el nombre <code>example_db</code> .</p> <pre data-bbox="503 1591 1507 1877"> {   "Effect": "Allow",   "Action": [     "glue:GetDatabase",     "glue:CreateDatabase"   ],   "Resource": [ </pre>

Instrucción DDL	Ejemplo de una política de acceso de IAM que concede acceso al recurso
	<pre> "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :catalog", "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :database / <i>example_db</i> " ] } </pre>
CREATE TABLE	<p>Permite crear una tabla con el nombre test en la base de datos example_db .</p> <pre> {   "Sid": "DatabasePermissions",   "Effect": "Allow",   "Action": [     "glue:GetDatabase",     "glue:GetDatabases"   ],   "Resource": [     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :catalog",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :database / <i>example_db</i> "   ] }, {   "Sid": "TablePermissions",   "Effect": "Allow",   "Action": [     "glue:GetTables",     "glue:GetTable",     "glue:GetPartitions",     "glue:CreateTable"   ],   "Resource": [     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :catalog",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :database / <i>example_db</i> ",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :table/<i>example_d</i> <i>b</i> /<i>test</i>"   ] } </pre>

Instrucción DDL	Ejemplo de una política de acceso de IAM que concede acceso al recurso
DROP DATABASE	<p>Permite borrar la base de datos <code>example_db</code> , incluidas todas las tablas que contiene.</p> <pre data-bbox="503 394 1507 1184">{   "Effect": "Allow",   "Action": [     "glue:GetDatabase",     "glue&gt;DeleteDatabase",     "glue:GetTables",     "glue:GetTable",     "glue&gt;DeleteTable"   ],   "Resource": [     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :catalog",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :database     / <i>example_db</i> ",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :table/<i>example_d</i> <i>b</i> /*",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :userDefi     nedFunction/ <i>example_db</i> /*"   ] }</pre>

Instrucción DDL	Ejemplo de una política de acceso de IAM que concede acceso al recurso
DROP TABLE	<p>Permite borrar una tabla particionada llamada <code>test</code> en la base de datos <code>example_db</code> . Si la tabla no tiene particiones, no incluya acciones de partición.</p> <pre data-bbox="505 443 1507 1192">{   "Effect": "Allow",   "Action": [     "glue:GetDatabase",     "glue:GetTable",     "glue&gt;DeleteTable",     "glue:GetPartitions",     "glue:GetPartition",     "glue&gt;DeletePartition"   ],   "Resource": [     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :catalog",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :database / <i>example_db</i> ",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :table/<i>example_d</i> <i>b</i> /<i>test</i>"   ] }</pre>

Instrucción DDL	Ejemplo de una política de acceso de IAM que concede acceso al recurso
MSCK REPAIR TABLE	<p>Permite actualizar los metadatos del catálogo después de agregar particiones compatibles con Hive a la tabla denominada <code>test</code> en la base de datos <code>example_db</code> .</p> <pre data-bbox="505 443 1507 1192"> {   "Effect": "Allow",   "Action": [     "glue:GetDatabase",     "glue:CreateDatabase",     "glue:GetTable",     "glue:GetPartitions",     "glue:GetPartition",     "glue:BatchCreatePartition"   ],   "Resource": [     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :catalog",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :database / <i>example_db</i> ",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :table/<i>example_db</i> /<i>test</i>"   ] } </pre>
SHOW DATABASES	<p>Permite enumerar todas las bases de datos en AWS Glue Data Catalog.</p> <pre data-bbox="505 1356 1507 1829"> {   "Effect": "Allow",   "Action": [     "glue:GetDatabase",     "glue:GetDatabases"   ],   "Resource": [     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :catalog",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :database/*"   ] } </pre>

Instrucción DDL	Ejemplo de una política de acceso de IAM que concede acceso al recurso
SHOW TABLES	<p>Permite enumerar todas las tablas de la base de datos <code>example_db</code> .</p> <pre data-bbox="505 348 1507 940"> {   "Effect": "Allow",   "Action": [     "glue:GetDatabase",     "glue:GetTables"   ],   "Resource": [     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :catalog",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :database / <i>example_db</i> ",     "arn:aws:glue: <i>us-east-1</i> :<i>123456789012</i> :table/<i>example_d</i> <i>b</i> /*"   ] }</pre>

## Acceso entre cuentas a los catálogos de datos de AWS Glue

Puede utilizar la característica de catálogo de AWS Glue de cuentas cruzadas de Athena para registrar un catálogo AWS Glue desde una cuenta que no sea la suya. Una vez configurados los permisos de IAM necesarios para AWS Glue y registrar el catálogo como un recurso de Athena [DataCatalog](#), puede utilizar Athena para ejecutar consultas entre cuentas. Para obtener información sobre cómo utilizar la consola de Athena para registrar un catálogo de otra cuenta, consulte [Registro de un AWS Glue Data Catalog de otra cuenta](#).

Para obtener más información sobre el acceso entre cuentas en AWS Glue, consulte [Concesión de acceso entre cuentas](#) en la Guía para desarrolladores de AWS Glue.

Antes de comenzar

Debido a que esta característica utiliza API y funciones de recursos `DataCatalog` de Athena existentes para habilitar el acceso entre cuentas, le recomendamos que lea los siguientes recursos antes de comenzar:

- [Conexión con orígenes de datos](#): contiene temas sobre el uso de Athena con fuentes del catálogo de datos de AWS Glue, Hive o Lambda.

- [Políticas de catálogos de datos de ejemplo](#): muestra cómo escribir políticas que controlan el acceso a catálogos de datos.
- [Uso de la AWS CLI con almacenes de metadatos de Hive](#): muestra cómo utilizar la AWS CLI con metaalmacenes de Hive, pero contiene casos de uso aplicables a otras fuentes de datos.

## Consideraciones y limitaciones

Actualmente, el acceso al catálogo AWS Glue entre cuentas de Athena tiene las siguientes limitaciones:

- La característica está disponible únicamente en las Regiones de AWS en las que se admite la versión 2 o una versión posterior del motor Athena. Para obtener más información acerca de las versiones de motor Athena, consulte [Control de versiones del motor Athena](#). Para actualizar la versión del motor de un grupo de trabajo, consulte [Cambio de las versiones del motor Athena](#).
- Cuando registra en su cuenta el AWS Glue Data Catalog de otra cuenta, crea un recurso DataCatalog regional que está vinculado a los datos de la otra cuenta en esa región en particular solamente.
- En la actualidad, las instrucciones CREATE VIEW que incluyen un catálogo AWS Glue entre cuentas no son compatibles.
- Los catálogos cifrados con claves administradas por AWS no se pueden consultar en todas las cuentas. Si desea consultar catálogos en todas las cuentas, utilice claves administradas por el cliente (KMS\_CMK). Para obtener más información sobre las diferencias entre las claves administradas por el cliente y las claves administradas por AWS, consulte [Claves del cliente y claves de AWS](#) en la Guía para desarrolladores de AWS Key Management Service.

## Introducción

En el siguiente escenario, la cuenta “Prestatario” (666666666666) quiere ejecutar una consulta SELECT que hace referencia al catálogo AWS Glue que pertenece a la cuenta “Propietario” (999999999999), como en el siguiente ejemplo:

```
SELECT * FROM ownerCatalog.tpch1000.customer
```

En el siguiente procedimiento, los pasos 1a y 1b muestran cómo concederle acceso de la cuenta del prestatario a los recursos de AWS Glue de la cuenta del propietario, tanto desde la perspectiva del propietario como desde la perspectiva del prestatario. El ejemplo concede acceso a la base de



datos `tpch1000` y la tabla `customer`. Cambie estos nombres de ejemplo para adaptarlos a sus necesidades.

Paso 1a: creación un rol de prestatario con una política para acceder a los recursos de AWS Glue del propietario

Para crear un rol de cuenta del prestatario con una política para acceder a los recursos de AWS Glue de la cuenta del propietario, puede utilizar la consola de AWS Identity and Access Management (IAM) o la [API de IAM](#). El siguiente procedimiento utiliza la consola de IAM.

Cómo crear un rol de prestatario y una política para acceder a los recursos de AWS Glue de la cuenta del propietario

1. Inicie sesión en la consola de IAM en <https://console.aws.amazon.com/iam/> desde la cuenta del prestatario.
2. En el panel de navegación, amplíe Administración de acceso y, a continuación, seleccione Políticas.
3. Elija Crear política.
4. En el editor de políticas, seleccione JSON.
5. En el editor de políticas, ingrese la siguiente política y, a continuación, modifíquela de acuerdo con sus requisitos:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "glue:*",
      "Resource": [
        "arn:aws:glue:us-east-1:999999999999:catalog",
        "arn:aws:glue:us-east-1:999999999999:database/tpch1000",
        "arn:aws:glue:us-east-1:999999999999:table/tpch1000/customer"
      ]
    }
  ]
}
```

6. Elija Siguiente.
7. En la página Revisar y crear, en Nombre de la política, ingrese un nombre para la política, (por ejemplo, **CrossGluePolicyForBorrowerRole**).

8. Elija Crear política.
9. Seleccione Roles en el panel de navegación.
10. Elija Crear rol.
11. En la página Seleccionar entidad de confianza elija Cuenta de AWS, y luego Siguiente.
12. En la página Añadir permisos, escriba el nombre de la política que creó en el cuadro de búsqueda (por ejemplo, **CrossGluePolicyForBorrowerRole**).
13. Elija la casilla de verificación situada junto al nombre de la política y, a continuación, elija Siguiente.
14. En la página Name, review, and create (Nombrar, revisar y crear), en Role name (Nombre del rol), ingrese un nombre para el rol (por ejemplo, **CrossGlueBorrowerRole**).
15. Elija Crear rol.

Paso 1b: creación de una política de propietario para concederle acceso a AWS Glue al prestatario

Para conceder acceso a AWS Glue desde la cuenta del propietario (999999999999) al rol de prestatario, puede utilizar la consola de AWS Glue o la operación de la API [PutResourcePolicy](#) de AWS Glue. El siguiente procedimiento utiliza la consola de AWS Glue.

Para conceder acceso a AWS Glue a la cuenta de prestatario desde el propietario

1. Inicie sesión en la consola de AWS Glue en <https://console.aws.amazon.com/glue/> desde la cuenta del propietario.
2. En el panel de navegación, amplíe Catálogo de datos y, a continuación, elija Configuración del catálogo.
3. En el navegador Permissions (Permisos), ingrese una política como la siguiente.  
Para *rolename*, introduzca el rol que el prestatario creó en el paso 1a (por ejemplo, **CrossGlueBorrowerRole**). Si desea aumentar el alcance de los permisos, puede utilizar el carácter comodín \* para los tipos de recursos de base de datos y tabla.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::666666666666:user/username",
```

```
        "arn:aws:iam::666666666666:role/rolename"
    ]
  },
  "Action": "glue:*",
  "Resource": [
    "arn:aws:glue:us-east-1:999999999999:catalog",
    "arn:aws:glue:us-east-1:999999999999:database/tpch1000",
    "arn:aws:glue:us-east-1:999999999999:table/tpch1000/customer"
  ]
}
]
```

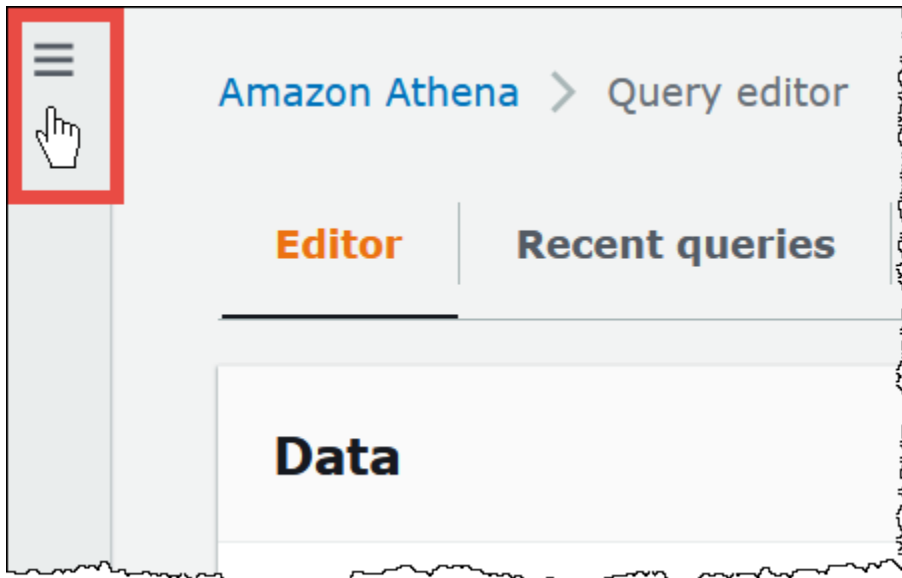
Después de terminar, se recomienda que utilice la [API de AWS Glue](#) para realizar algunas llamadas de prueba entre cuentas y confirmar que los permisos están configurados de la manera esperada.

Paso 2: El prestatario registra la AWS Glue Data Catalog que pertenece a la cuenta de propietario

El siguiente procedimiento muestra cómo utilizar la consola de Athena para configurar la AWS Glue Data Catalog en la cuenta de Amazon Web Services del propietario como origen de datos. Para obtener información sobre el uso de operaciones de API en lugar de la consola para registrar el catálogo, consulte [Uso de la API para registrar un catálogo de datos de Athena que pertenece a la cuenta del propietario](#).

Para registrar una AWS Glue Data Catalog que pertenece a otra cuenta

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.



3. Amplíe Administración y, a continuación, elija Orígenes de datos.
4. En la parte superior derecha de la consola, seleccione Create data source (Crear origen de datos).
5. En la página Elegir un origen de datos, para Orígenes de datos, seleccione S3: AWS Glue Data Catalog y, a continuación, elija Siguiente.
6. En la página Introducir detalles del origen de datos, en la sección AWS Glue Data Catalog, para Elegir un AWS Glue Data Catalog, elija AWS Glue Data Catalog en otra cuenta.
7. En Dataset details (Detalles del origen de datos), ingrese la siguiente información:
  - Data source name (Nombre del origen de datos): ingrese el nombre que desea utilizar en las consultas SQL para hacer referencia al catálogo de datos de la otra cuenta.
  - Descripción: (opcional) ingrese una descripción del catálogo de datos en la otra cuenta.
  - ID del catálogo: ingrese el ID de cuenta de Amazon Web Services de 12 dígitos de la cuenta a la que pertenece el catálogo de datos. El ID de cuenta de Amazon Web Services es el ID del catálogo.
8. (Opcional) Amplíe Etiquetas y luego ingrese los pares clave-valor que quiera asociar con el origen de datos. Para obtener más información acerca de las etiquetas, consulte [Etiquetado de recursos de Athena](#).
9. Elija Siguiente.
10. En la página Review and create (Revisar y crear), revise la información que ha proporcionado y, a continuación, elija Create data source (Crear un origen de datos). En la página Data source

details (Detalles de origen de datos) se enumeran las bases de datos y etiquetas del catálogo de datos que ha registrado.

11. Elija Edit Data Source (Editar origen de datos). El catálogo de datos que ha registrado se muestra en la columna Data source name (Nombre de origen de datos).
12. Para ver o editar información sobre el nuevo catálogo de datos, elija el catálogo y, a continuación, elija Actions (Acciones) y Edit (Editar).
13. Para eliminar el nuevo catálogo de datos, elija el catálogo y, a continuación, elija Actions (Acciones) y Delete (Eliminar).

### Paso 3: El prestatario envía una consulta

El prestatario envía una consulta que hace referencia al catálogo mediante la sintaxis *catalog.database.table*, como en el ejemplo siguiente:

```
SELECT * FROM ownerCatalog.tpch1000.customer
```

En lugar de utilizar la sintaxis completa, el prestatario también puede especificar el catálogo según el contexto si lo hace pasar por [QueryExecutionContext](#).

### Permisos de Simple Storage Service (Amazon S3) adicionales

- Si la cuenta del prestatario utiliza una consulta de Athena para escribir nuevos datos en una tabla de la cuenta del propietario, el propietario no tendrá acceso de manera automática a estos datos en Amazon S3, aunque la tabla exista en la cuenta del propietario. Esto se debe a que, a menos que se configure lo contrario, el prestatario es el propietario del objeto de la información en Amazon S3. Para concederle acceso a los datos al propietario, establezca los permisos en los objetos de manera apropiada como paso adicional.
- Ciertas operaciones de DDL entre cuentas como [MSCK REPAIR TABLE](#) requieren permisos de Amazon S3. Por ejemplo, si la cuenta del prestatario está realizando una operación MSCK REPAIR entre cuentas contra una tabla de la cuenta del propietario que tiene sus datos en un bucket de S3 de la cuenta de propietario, ese bucket debe concederle permisos al rol asumido por el prestatario para que la consulta se realice de forma correcta.

Para obtener más información acerca de cómo conceder permisos de buckets, consulte [¿Cómo configuro permisos de buckets con ACL?](#) en la Guía del usuario de Amazon Simple Storage Service.

## Uso de un catálogo de forma dinámica

En algunos casos, es posible que desee realizar pruebas rápidamente contra un catálogo de AWS Glue entre cuentas sin el paso previo de registrarlo. Puede realizar consultas entre cuentas de forma dinámica sin crear el objeto de recurso `DataCatalog` si los permisos necesarios de IAM y Amazon S3 están configurados correctamente como se describió anteriormente en este documento.

Para hacer referencia explícita a un catálogo sin registro, utilice la sintaxis del ejemplo siguiente:

```
SELECT * FROM "glue:arn:aws:glue:us-east-1:999999999999:catalog".tpch1000.customer
```

Utilice el formato "glue:<arn>", donde <arn> es el [ARN de AWS Glue Data Catalog](#) que desea utilizar. En el ejemplo, Athena utiliza esta sintaxis para apuntar dinámicamente al catálogo de datos AWS Glue de la cuenta 999999999999 como si hubiera creado por separado un objeto de `DataCatalog` para él.

### Notas para el uso de catálogos dinámicos

Cuando utilice catálogos dinámicos, recuerde los siguientes puntos.

- El uso de un catálogo dinámico requiere los permisos de IAM que utiliza normalmente para las operaciones de API de catálogo de datos de Athena. La principal diferencia es que el nombre de recurso del catálogo de datos sigue la convención de nomenclatura de `glue:*`.
- El ARN del catálogo debe pertenecer a la misma región en la que se está ejecutando la consulta.
- Cuando utilice un catálogo dinámico en una consulta o vista DML, indíquelo con comillas dobles escapadas (`\"`). Cuando utilice un catálogo dinámico en una consulta DDL, indíquelo con acentos graves (```).

### Uso de la API para registrar un catálogo de datos de Athena que pertenece a la cuenta del propietario

En lugar de utilizar la consola de Athena como se describe en el paso 2, se pueden utilizar las operaciones de la API para registrar el catálogo de datos que pertenece a la cuenta del propietario.

El creador del recurso [DataCatalog](#) de Athena debe tener los permisos necesarios para ejecutar la operación de la API [CreateDataCatalog](#) de Athena. En función de sus requisitos, es posible que sea necesario acceder a operaciones de API adicionales. Para obtener más información, consulte [Políticas de catálogos de datos de ejemplo](#).

El siguiente cuerpo de solicitud de `CreateDataCatalog` registra un catálogo de AWS Glue para acceso entre cuentas:

```
# Example CreateDataCatalog request to register a cross-account Glue catalog:
{
  "Description": "Cross-account Glue catalog",
  "Name": "ownerCatalog",
  "Parameters": {"catalog-id" : "999999999999" # Owner's account ID
},
  "Type": "GLUE"
}
```

El siguiente código de muestra utiliza un cliente Java para crear el objeto `DataCatalog`.

```
# Sample code to create the DataCatalog through Java client
CreateDataCatalogRequest request = new CreateDataCatalogRequest()
    .withName("ownerCatalog")
    .withType(DataCatalogType.GLUE)
    .withParameters(ImmutableMap.of("catalog-id", "999999999999"));

athenaClient.createDataCatalog(request);
```

Después de estos pasos, el prestatario debe ver el `ownerCatalog` cuando llama a la operación de la API [ListDataCatalogs](#).

Recursos adicionales de

- [Registro de un AWS Glue Data Catalog de otra cuenta](#)
- [Configurar el acceso entre cuentas a un AWS Glue Data Catalog compartido mediante Amazon Athena](#) en los Patrones de recomendaciones de AWS.
- [Consultar un AWS Glue Data Catalog entre cuentas con Amazon Athena](#) en el Blog de macrodatos de AWS
- [Concesión de acceso entre cuentas](#) en la Guía para desarrolladores de AWS Glue

## Acceso desde Athena a metadatos cifrados en AWS Glue Data Catalog

Si utiliza el AWS Glue Data Catalog con Amazon Athena, puede habilitar el cifrado en el AWS Glue Data Catalog mediante el uso de la consola de AWS Glue o la API. Para obtener más información, consulte [Cifrado del catálogo de datos](#) en la Guía para desarrolladores de AWS Glue.

Si el AWS Glue Data Catalog está cifrado, debe agregar las siguientes acciones a todas las políticas que se utilizan para obtener acceso a Athena:

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:GenerateDataKey",
      "kms:Decrypt",
      "kms:Encrypt"
    ],
    "Resource": "(arn of the key used to encrypt the catalog)"
  }
}
```

Siempre que utilice políticas de IAM, asegúrese de seguir las prácticas recomendadas de IAM. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM.

## Acceso a grupos de trabajo y etiquetas

Un grupo de trabajo es un recurso administrado por Athena. Por tanto, si su política de grupos de trabajo utiliza acciones que toman a `workgroup` como entrada, deberá especificar el ARN del grupo de trabajo de la siguiente manera, donde *workgroup-name* es el nombre del grupo de trabajo:

```
"Resource": [arn:aws:athena:region:AWSacctID:workgroup/workgroup-name]
```

Por ejemplo, en el caso de un grupo de trabajo denominado `test_workgroup` en la región `us-west-2` de la cuenta de Amazon Web Services `123456789012`, especifique el grupo de trabajo como un recurso usando el siguiente ARN:

```
"Resource": ["arn:aws:athena:us-east-2:123456789012:workgroup/test_workgroup"]
```

Para acceder a los grupos de trabajo habilitados para la propagación de identidades fiables (TIP), los usuarios del IAM Identity Center deben estar asignados al `IdentityCenterApplicationArn` que devuelve la respuesta de la acción de la API [GetWorkGroup](#) de Athena.

- Para ver una lista de políticas de grupos de trabajo, consulte [the section called “Ejemplos de políticas de grupos de trabajo”](#).



- Para obtener una lista de políticas basadas en etiquetas para grupos de trabajo, consulte [Políticas de control de acceso de IAM basado en etiquetas](#).
- Para obtener más información acerca de la creación de políticas de IAM para grupos de trabajo, consulte [Políticas de IAM para acceder a los grupos de trabajo](#).
- Para obtener una lista completa de las acciones de Amazon Athena, consulte los nombres de acciones de la API en la [Referencia de API de Amazon Athena](#).
- Para obtener más información sobre las políticas de IAM, consulte [Creación de políticas con el editor visual](#) en la Guía del usuario de IAM.

Siempre que utilice políticas de IAM, asegúrese de seguir las prácticas recomendadas de IAM. Para más información, consulte [Prácticas recomendadas de seguridad en IAM](#) en la Guía del usuario de IAM.

## Permitir acceso a instrucciones preparadas

En este tema se tratan los permisos de IAM para las instrucciones preparadas en Amazon Athena. Siempre que utilice políticas de IAM, asegúrese de seguir las prácticas recomendadas de IAM. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM.

Para obtener más información acerca de las instrucciones preparadas, consulte [Uso de consultas parametrizadas](#).

Los siguientes permisos de IAM son necesarios para crear, administrar y ejecutar instrucciones preparadas.

```
athena:CreatePreparedStatement
athena:UpdatePreparedStatement
athena:GetPreparedStatement
athena:ListPreparedStatements
athena>DeletePreparedStatement
```

Utilice estos permisos como se muestra en la tabla a continuación.

Para ello	Utilice estos permisos
Ejecute una consulta PREPARE	athena:StartQueryExecution athena:CreatePreparedStatement

Para ello	Utilice estos permisos
Vuelva a ejecutar una consulta PREPARE para actualizar una instrucción preparada existente	athena:StartQueryExecution    athena:UpdatePreparedStatement
Ejecute una consulta EXECUTE	athena:StartQueryExecution    athena:GetPreparedStatement
Ejecute una consulta DEALLOCATE PREPARE	athena:StartQueryExecution    athena>DeletePreparedStatement

## Ejemplo

En el siguiente ejemplo de política de IAM se otorgan permisos para administrar y ejecutar instrucciones preparadas en un ID de cuenta y grupo de trabajo especificados.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:StartQueryExecution",
        "athena:CreatePreparedStatement",
        "athena:UpdatePreparedStatement",
        "athena:GetPreparedStatement",
        "athena>DeletePreparedStatement",
        "athena:ListPreparedStatements"
      ],
      "Resource": [
        "arn:aws:athena:*:111122223333:workgroup/<workgroup-name>"
      ]
    }
  ]
}
```

## Uso de Athena con claves de contexto CalledVia

Cuando una [entidad principal](#) hace una [solicitud](#) a AWS, AWS recopila la información de la solicitud en un contexto de solicitud que la evalúa y la autoriza. Puede utilizar el elemento `Condition` de una política JSON para comparar las claves de la solicitud de contexto con los valores de claves que especifique en su política. Las claves de contexto de condición globales son claves de condición con un prefijo `aws:`.

La clave de contexto `aws:CalledVia`

Puede utilizar la clave de contexto de condición global [aws:CalledVia](#) para comparar los servicios de la política con los servicios que realizaron solicitudes en nombre de la entidad principal de IAM (usuario o rol). Cuando una entidad principal hace una solicitud a un Servicio de AWS, ese servicio puede utilizar las credenciales de la entidad principal para hacer solicitudes posteriores a otros servicios. La clave `aws:CalledVia` contiene una lista ordenada de cada servicio de la cadena que realizó solicitudes en nombre de la entidad principal.

Al especificar el nombre de una entidad principal del servicio para la clave de contexto `aws:CalledVia`, puede hacer que la clave de contexto sea específica del Servicio de AWS. Por ejemplo, puede utilizar la clave de condición `aws:CalledVia` para limitar las solicitudes únicamente a las realizadas desde Athena. Para utilizar la clave de condición `aws:CalledVia` en una política con Athena, especifica el nombre de la entidad principal de servicio de Athena `athena.amazonaws.com`, como en el siguiente ejemplo.

```
...
  "Condition": {
    "ForAnyValue:StringEquals": {
      "aws:CalledVia": "athena.amazonaws.com"
    }
  }
...

```

Puede utilizar la clave de contexto `aws:CalledVia` para asegurarse de que las personas que llaman solo tengan acceso a un recurso (como una función de Lambda) si llaman al recurso desde Athena.

**Note**

La clave de contexto `aws:CalledVia` no es compatible con la característica de propagación de identidades fiables.

Agregar una clave de contexto `CalledVia` opcional para el acceso detallado a una función de Lambda

Athena requiere que la persona que llama tenga permisos `lambda:InvokeFunction` para invocar la función de Lambda asociada a la consulta. La siguiente instrucción permite el acceso detallado a una función de Lambda de modo que el usuario puede utilizar solo Athena para invocar la función de Lambda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor3",
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:us-east-1:111122223333:function:OneAthenaLambdaFunction",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:CalledVia": "athena.amazonaws.com"
        }
      }
    }
  ]
}
```

En el siguiente ejemplo, se muestra la adición de la instrucción anterior a una política que permite a un usuario ejecutar y leer una consulta federada. Las entidades principales a las que se les permite realizar estas acciones pueden ejecutar consultas que especifican catálogos de Athena asociados a un origen de datos federado. Sin embargo, la entidad principal no puede acceder a la función de Lambda asociada a menos que la función se invoque través de Athena.

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

```

    "Sid": "VisualEditor0",
    "Effect": "Allow",
    "Action": [
        "athena:GetWorkGroup",
        "s3:PutObject",
        "s3:GetObject",
        "athena:StartQueryExecution",
        "s3:AbortMultipartUpload",
        "athena:StopQueryExecution",
        "athena:GetQueryExecution",
        "athena:GetQueryResults",
        "s3:ListMultipartUploadParts"
    ],
    "Resource": [
        "arn:aws:athena:*:111122223333:workgroup/WorkGroupName",
        "arn:aws:s3:::MyQueryResultsBucket/*",
        "arn:aws:s3:::MyLambdaSpillBucket/MyLambdaSpillPrefix*"
    ]
},
{
    "Sid": "VisualEditor1",
    "Effect": "Allow",
    "Action": "athena:ListWorkGroups",
    "Resource": "*"
},
{
    "Sid": "VisualEditor2",
    "Effect": "Allow",
    "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation"
    ],
    "Resource": "arn:aws:s3:::MyLambdaSpillBucket"
},
{
    "Sid": "VisualEditor3",
    "Effect": "Allow",
    "Action": "lambda:InvokeFunction",
    "Resource": [
        "arn:aws:lambda:*:111122223333:function:OneAthenaLambdaFunction",
        "arn:aws:lambda:*:111122223333:function:AnotherAthenaLambdaFunction"
    ],
    "Condition": {

```

```
        "ForAnyValue:StringEquals": {
            "aws:CalledVia": "athena.amazonaws.com"
        }
    }
]
```

Para obtener más información acerca de las claves de condición `CalledVia`, consulte [claves de contexto de condición globales de AWS](#) en la Guía del usuario de IAM.

## Permitir el acceso a un conector de datos de Athena para un almacén de metadatos externo de Hive

Los ejemplos de políticas de permisos de este tema muestran las acciones permitidas necesarias y los recursos para las que están permitidas. Examine estas políticas detenidamente y modifíquelas de acuerdo con sus requisitos antes de adjuntar políticas de permisos similares a las identidades de IAM.

- [Example Policy to Allow an IAM Principal to Query Data Using Athena Data Connector for External Hive Metastore](#)
- [Example Policy to Allow an IAM Principal to Create an Athena Data Connector for External Hive Metastore](#)

Example : permitir que una entidad principal de IAM consulte datos con el conector de datos de Athena para un almacén de metadatos externo de Hive

La siguiente política se conecta a las entidades principales de IAM, además de [Política administrada de AWS: AmazonAthenaFullAccess](#), que concede acceso completo a las acciones de Athena.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": [
        "lambda:GetFunction",
        "lambda:GetLayerVersion",
        "lambda:InvokeFunction"
      ]
    }
  ]
}
```

```

    ],
    "Resource": [
      "arn:aws:lambda:*:111122223333:function:MyAthenaLambdaFunction",
      "arn:aws:lambda:*:111122223333:function:AnotherAthenaLambdaFunction",
      "arn:aws:lambda:*:111122223333:layer:MyAthenaLambdaLayer:*"
    ]
  },
  {
    "Sid": "VisualEditor2",
    "Effect": "Allow",
    "Action": [
      "s3:GetBucketLocation",
      "s3:GetObject",
      "s3:ListBucket",
      "s3:PutObject",
      "s3:ListMultipartUploadParts",
      "s3:AbortMultipartUpload"
    ],
    "Resource": "arn:aws:s3:::MyLambdaSpillBucket/MyLambdaSpillLocation"
  }
]
}

```

## Explicación de permisos

Acciones permitidas	Explicación
<pre> "s3:GetBucketLocation", "s3:GetObject", "s3:ListBucket", "s3:PutObject", "s3:ListMultipartUploadParts", "s3:AbortMultipartUpload" </pre>	<p>Las acciones de s3 que permiten leer y escribir en el recurso especificado como "arn:aws:s3::: <i>MyLambdaSpillBucket/MyLambdaSpillLocation</i> ", donde <i>MyLambdaSpillLocation</i> identifica el bucket de desbordamiento que se especifica en la configuración de la función o funciones Lambda que se invocan. El identificador de recursos <i>arn:aws:lambda:*: MyAWSacct Id :layer:MyAthenaLambdaLayer :*</i> es necesario solo si utiliza una capa Lambda para crear dependencias de tiempo de ejecución personalizadas para reducir el tamaño del artefacto de la función en el momento de la</p>

Acciones permitidas	Explicación
<pre>"lambda:GetFunction", "lambda:GetLayerVersion", "lambda:InvokeFunction"</pre>	<p>implementación. * en la última posición es un comodín para la versión de capa.</p> <p>Permite que las consultas invoquen las funciones AWS Lambda especificadas en el bloque Resource. Por ejemplo, <code>arn:aws:lambda:*:MyAWSAcctId:function:MyAthenaLambdaFunction</code>, donde <code>MyAthenaLambdaFunction</code> especifica el nombre de una función Lambda que se va a invocar. Se pueden especificar varias funciones como se muestra en el ejemplo.</p>

Example : permitir que una entidad principal de IAM cree un conector de datos de Athena para un almacén de metadatos externo de Hive

La siguiente política se conecta a las entidades principales de IAM, además de [Política administrada de AWS: AmazonAthenaFullAccess](#), que concede acceso completo a las acciones de Athena.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "lambda:GetFunction",
        "lambda:ListFunctions",
        "lambda:GetLayerVersion",
        "lambda:InvokeFunction",
        "lambda:CreateFunction",
        "lambda>DeleteFunction",
        "lambda:PublishLayerVersion",
        "lambda>DeleteLayerVersion",
        "lambda:UpdateFunctionConfiguration",
        "lambda:PutFunctionConcurrency",
        "lambda>DeleteFunctionConcurrency"
      ]
    }
  ],
```



```
    "Resource": "arn:aws:lambda:*:111122223333:  
function: MyAthenaLambdaFunctionsPrefix*"  
  }  
]  
}
```

## Explicación de permisos

Permite que las consultas invoquen las funciones AWS Lambda para las funciones de AWS Lambda especificadas en el bloque Resource. Por ejemplo, `arn:aws:lambda:*:MyAWSacctId:function:MyAthenaLambdaFunction`, donde `MyAthenaLambdaFunction` especifica el nombre de una función Lambda que se va a invocar. Se pueden especificar varias funciones como se muestra en el ejemplo.

## Permitir a la función de Lambda el acceso a los almacenes de metadatos externos de Hive

Para invocar una función de Lambda en su cuenta, debe crear un rol que tenga los siguientes permisos:

- `AWSLambdaVPCLambdaAccessExecutionRole`: un permiso [de rol de ejecución de AWS Lambda](#) para administrar interfaces de red elásticas que conectan la función a una VPC. Asegúrese de disponer de un número suficiente de interfaces de red y direcciones IP disponibles.
- `AmazonAthenaFullAccess`: la política administrada [AmazonAthenaFullAccess](#) concede acceso total a Athena.
- Una política de Amazon S3 para permitir que la función de Lambda escriba en S3 y para permitir que Athena lea desde S3.

Por ejemplo, la siguiente política define el permiso para la ubicación del desbordamiento `s3:\mybucket\spill`.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:GetBucketLocation",  
        "s3:GetObject",
```

```

        "s3:ListBucket",
        "s3:PutObject"
    ],
    "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/spill"
    ]
}
]
}

```

Siempre que utilice políticas de IAM, asegúrese de seguir las prácticas recomendadas de IAM. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM.

### Creación de funciones de Lambda

Para crear una función de Lambda en su cuenta, se requieren permisos de desarrollo de funciones o el rol `AWSLambdaFullAccess`. Para obtener más información, consulte [Políticas de IAM basadas en identidades para AWS Lambda](#).

Dado que Athena utiliza el AWS Serverless Application Repository para crear funciones de Lambda, el superusuario o administrador que crea funciones de Lambda también debe tener políticas de IAM para [permitir consultas federadas de Athena](#).

### Operaciones de API de registro de catálogos y API de metadatos

Para acceder a las operaciones de API de registro de catálogos y API de metadatos, utilice la [política administrada de AmazonAthenaFullAccess](#). Si no utiliza esta política, agregue las siguientes operaciones de API a sus políticas de Athena:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:ListDataCatalogs",
        "athena:GetDataCatalog",
        "athena:CreateDataCatalog",
        "athena:UpdateDataCatalog",
        "athena>DeleteDataCatalog",

```

```
        "athena:GetDatabase",
        "athena:ListDatabases",
        "athena:GetTableMetadata",
        "athena:ListTableMetadata"
    ],
    "Resource": [
        "*"
    ]
}
]
```

## Invocación de Lambda entre regiones

Para invocar una función de Lambda en una región distinta de la región en la que está ejecutando consultas de Athena, utilice el ARN completo de la función de Lambda. De forma predeterminada, Athena invoca funciones de Lambda definidas en la misma región. Si necesita invocar una función de Lambda para acceder a un metaalmacén de Hive en una región distinta de la región en la que ejecuta consultas de Athena, debe proporcionar el ARN completo de la función de Lambda.

Por ejemplo, supongamos que define el catálogo ehms en la región Europa (Fráncfort) eu-central-1 para utilizar la siguiente función de Lambda en Este de EE. UU. (Norte de Virginia).

```
arn:aws:lambda:us-east-1:111122223333:function:external-hms-service-new
```

Cuando se especifica el ARN completo de esta manera, Athena puede llamar a la función de Lambda `external-hms-service-new` en `us-east-1` para obtener los datos de metaalmacén de Hive de `eu-central-1`.

### Note

El catálogo ehms debe estar registrado en la misma región en la que se ejecutan las consultas de Athena.

## Invocación de Lambda entre cuentas

A veces es posible que necesite acceso a una metaalmacén de Hive desde una cuenta diferente. Por ejemplo, para ejecutar un metaalmacén de Hive, podría lanzar un clúster de EMR desde una cuenta

distinta de la que utiliza para las consultas de Athena. Diferentes grupos o equipos pueden ejecutar el metaalmacén de Hive con distintas cuentas dentro de su VPC. O es posible que desee acceder a metadatos de distintos metaalmacenes de Hive de distintos grupos o equipos.

Athena utiliza el [soporte de AWS Lambda para el acceso entre cuentas](#) para habilitar el acceso entre cuentas para metaalmacenes de Hive.

#### Note

Tenga en cuenta que el acceso entre cuentas para Athena normalmente implica el acceso entre cuentas tanto para metadatos como para datos en Amazon S3.

Imagine la siguiente situación:

- La cuenta 111122223333 configura la función de Lambda `external-hms-service-new` en `us-east-1` en Athena para acceder a un metaalmacén de Hive que se ejecuta en un clúster de EMR.
- La cuenta 111122223333 quiere permitir que la cuenta 444455556666 acceda a los datos del metaalmacén de Hive.

Para conceder a la cuenta 444455556666 acceso a la función de Lambda `external-hms-service-new`, la cuenta 111122223333 utiliza el siguiente comando AWS CLI de `add-permission`. El comando se ha formateado para fines de legibilidad.

```
$ aws --profile perf-test lambda add-permission
  --function-name external-hms-service-new
  --region us-east-1
  --statement-id Id-ehms-invocation2
  --action "lambda:InvokeFunction"
  --principal arn:aws:iam::444455556666:user/perf1-test
{
  "Statement": [{"Sid": "Id-ehms-invocation2",
    "Effect": "Allow",
    "Principal": {"AWS": "arn:aws:iam::444455556666:user/perf1-test"}},
    {"Action": "lambda:InvokeFunction",
    "Resource": "arn:aws:lambda:us-east-1:111122223333:function:external-hms-service-new"}]
}
```

Para verificar el permiso de Lambda, utilice el comando `get-policy`, como en el siguiente ejemplo. El comando se ha formateado para fines de legibilidad.

```
$ aws --profile perf-test lambda get-policy
  --function-name arn:aws:lambda:us-east-1:111122223333:function:external-hms-
service-new
  --region us-east-1
{
  "RevisionId": "711e93ea-9851-44c8-a09f-5f2a2829d40f",
  "Policy": "{\"Version\":\"2012-10-17\",
    \"Id\":\"default\",
    \"Statement\": [{\"Sid\":\"Id-ehms-invocation2\",
      \"Effect\":\"Allow\",
      \"Principal\":{\"AWS\":
\"arn:aws:iam::444455556666:user/perf1-test\"},
      \"Action\":\"lambda:InvokeFunction\",
      \"Resource\":\"arn:aws:lambda:us-
east-1:111122223333:function:external-hms-service-new\"}]}"
}
```

Después de agregar el permiso, puede usar un ARN completo de la función de Lambda de `us-east-1` como el siguiente al definir el catálogo `ehms`:

```
arn:aws:lambda:us-east-1:111122223333:function:external-hms-service-new
```

Para obtener información acerca de la invocación entre regiones, consulte [Invocación de Lambda entre regiones](#) anteriormente en este tema.

### Concesión de acceso a datos entre cuentas

Antes de poder ejecutar consultas de Athena, debe conceder acceso entre cuentas a los datos de Amazon S3. Puede hacerlo de una de las siguientes formas:

- Actualice la política de lista de control de acceso del bucket de Amazon S3 con un [ID de usuario canónico](#).
- Agregue acceso entre cuentas a la política de bucket de Amazon S3.

Por ejemplo, agregue la siguiente política a la política de bucket de Amazon S3 en la cuenta `111122223333` para permitir que la cuenta `444455556666` lea datos de la ubicación de Amazon S3 especificada.

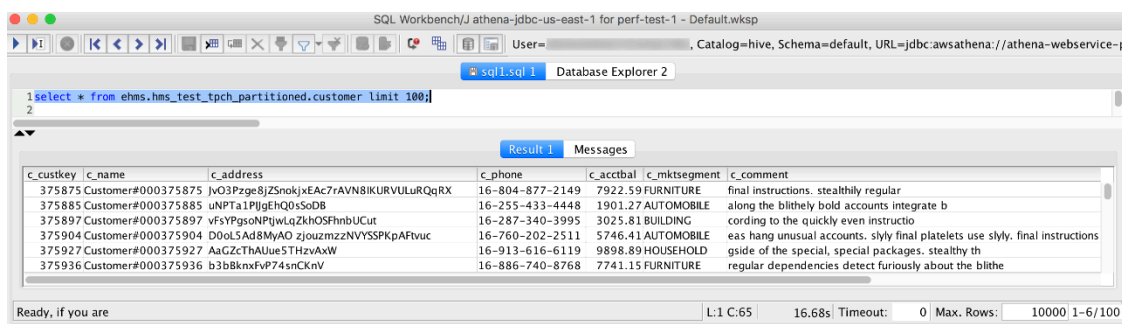
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1234567890123",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::444455556666:user/perf1-test"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::athena-test/lambda/dataset/*"
    }
  ]
}
```

### Note

Es posible que tenga que conceder acceso entre cuentas a Amazon S3 no solo a sus datos, sino también a su ubicación de desbordamiento de Amazon S3. Su función de Lambda desborda datos adicionales a la ubicación de desbordamiento cuando el tamaño del objeto de respuesta supera un umbral determinado. Consulte al principio de este tema para obtener una política de ejemplo.

En el ejemplo actual, después de conceder acceso entre cuentas a 444455556666, 444455556666 puede utilizar el catálogo ehms en su propia account para consultar tablas definidas en la cuenta 111122223333.

En el siguiente ejemplo, el perfil de SQL Workbench perf-test-1 es para la cuenta 444455556666. La consulta utiliza el catálogo ehms para acceder al metaalmacén de Hive y a los datos de Amazon S3 en la cuenta 111122223333.



c_custkey	c_name	c_address	c_phone	c_acctbal	c_mktsegment	c_comment
375875	Customer#000375875	JvO3Pzge8jZSnokjxEAc7rAVN8IKURVULuRQqRX	16-804-877-2149	7922.59	FURNITURE	final instructions. stealthily regular
375885	Customer#000375885	uNPTa1PIJgEHQ0sSoDB	16-255-433-4448	1901.27	AUTOMOBILE	along the blithely bold accounts integrate b
375897	Customer#000375897	vFsYPgsoNPjwLqZkhOSFhmbUCut	16-287-340-3995	3025.81	BUILDING	cording to the quickly even instructio
375904	Customer#000375904	D0oL5Ad8MyAO zjouzmzzNVYSSPKpAFvuc	16-760-202-2511	5746.41	AUTOMOBILE	eas hang unusual accounts. slyly final platelets use slyly. final instructions
375927	Customer#000375927	AaGzCThAUue5THzvAxw	16-913-616-6119	9898.89	HOUSEHOLD	gside of the special, special packages. stealthy th
375936	Customer#000375936	b3bBknxFvP74snCKnV	16-886-740-8768	7741.15	FURNITURE	regular dependencies detect furiously about the blithe

## Ejemplos de políticas de permisos de IAM para permitir la consulta federada de Athena

Los ejemplos de políticas de permisos de este tema muestran las acciones permitidas necesarias y los recursos para las que están permitidas. Examine estas políticas detenidamente y modifíquelas de acuerdo con sus requisitos antes de adjuntarlas a las identidades de IAM.

Para obtener información sobre cómo adjuntar políticas a identidades de IAM, consulte [Adición y eliminación de permisos de identidad de IAM](#) en la [Guía del usuario de IAM](#).

- [Example policy to allow an IAM principal to run and return results using Athena Federated Query](#)
- [Example Policy to Allow an IAM Principal to Create a Data Source Connector](#)

Example : permitir que una entidad principal de IAM ejecute y devuelva resultados mediante la consulta federada de Athena

La siguiente política de permisos basada en identidad permite acciones que un usuario u otra entidad principal de IAM necesita para utilizar la consulta federada de Athena. Las entidades principales a las que se les permite realizar estas acciones pueden ejecutar consultas que especifican catálogos de Athena asociados a un origen de datos federado.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Athena",
      "Effect": "Allow",
      "Action": [
        "athena:GetDataCatalog",
        "athena:GetQueryExecution",
        "athena:GetQueryResults",
        "athena:GetWorkGroup",
        "athena:StartQueryExecution",
        "athena:StopQueryExecution"
      ],
      "Resource": [
        "arn:aws:athena:*:111122223333:workgroup/WorkgroupName",
        "arn:aws:athena:aws_region:111122223333:datacatalog/DataCatalogName"
      ]
    },
  ],
}
```

```

    "Sid": "ListAthenaWorkGroups",
    "Effect": "Allow",
    "Action": "athena:ListWorkGroups",
    "Resource": "*"
  },
  {
    "Sid": "Lambda",
    "Effect": "Allow",
    "Action": "lambda:InvokeFunction",
    "Resource": [
      "arn:aws:lambda:*:111122223333:function:OneAthenaLambdaFunction",
      "arn:aws:lambda:*:111122223333:function:AnotherAthenaLambdaFunction"
    ]
  },
  {
    "Sid": "S3",
    "Effect": "Allow",
    "Action": [
      "s3:AbortMultipartUpload",
      "s3:GetBucketLocation",
      "s3:GetObject",
      "s3:ListBucket",
      "s3:ListMultipartUploadParts",
      "s3:PutObject"
    ],
    "Resource": [
      "arn:aws:s3:::MyLambdaSpillBucket",
      "arn:aws:s3:::MyLambdaSpillBucket/*",
      "arn:aws:s3:::MyQueryResultsBucket",
      "arn:aws:s3:::MyQueryResultsBucket/*"
    ]
  }
]
}

```

## Explicación de permisos

Acciones permitidas	Explicación
<pre> "athena:GetQueryExecution", "athena:GetQueryResults", "athena:GetWorkGroup", "athena:StartQueryExecution", </pre>	<p>Permisos de Athena necesarios para ejecutar consultas federadas.</p>



Acciones permitidas	Explicación
"athena:StopQueryExecution"	
"athena:GetDataCatalog", "athena:GetQueryExecution", "athena:GetQueryResults", "athena:GetWorkGroup", "athena:StartQueryExecution", "athena:StopQueryExecution"	Permisos de Athena necesarios para ejecutar consultas de vistas federadas. La acción <code>GetDataCatalog</code> es obligatoria para las vistas.
"lambda:InvokeFunction"	Permite que las consultas invoquen las funciones AWS Lambda para las funciones de AWS Lambda especificadas en el bloque <code>Resource</code> . Por ejemplo, <code>arn:aws:lambda:*:MyAWSAcctId:function:MyAthenaLambdaFunction</code> , donde <code>MyAthenaLambdaFunction</code> especifica el nombre de una función Lambda que se va a invocar. Se pueden especificar varias funciones, como se muestra en el ejemplo.

Acciones permitidas	Explicación
<pre>"s3:AbortMultipartUpload", "s3:GetBucketLocation", "s3:GetObject", "s3:ListBucket", "s3:ListMultipartUploadParts", "s3:PutObject"</pre>	<p>Los permisos <code>s3:ListBucket</code> y <code>s3:GetBucketLocation</code> son necesarios para acceder al bucket de resultados de la consulta para las entidades principales de IAM que ejecutan <code>StartQueryExecution</code>.</p> <p><code>s3:PutObject</code>, <code>s3:ListMultipartUploadParts</code> y <code>s3:AbortMultipartUpload</code> permiten escribir los resultados de la consulta en todas las subcarpetas del bucket de resultados de la consulta según lo especificado por el identificador de recursos <code>arn:aws:s3::: <i>MyQueryResultsBucket</i> /*</code>, donde <i>MyQueryResultsBucket</i> es el bucket de resultados de la consulta de Athena. Para obtener más información, consulte <a href="#">Trabajo con resultados de las consultas, consultas recientes y archivos de salida</a>.</p> <p><code>s3:GetObject</code> permite leer los resultados de la consulta y el historial de consultas para el recurso especificado como <code>arn:aws:s3::: <i>MyQueryResultsBucket</i></code>, donde <i>MyQueryResultsBucket</i> es el bucket de resultados de la consulta de Athena.</p> <p><code>s3:GetObject</code> también permite leer desde el recurso especificado como <code>"arn:aws:s3::: <i>MyLambdaSpillBucket</i> /<i>MyLambdaSpillPrefix</i> *"</code>, donde <i>MyLambdaSpillPrefix</i> se especifica en la configuración de la función o funciones Lambda invocadas.</p>

## Example : permitir que una entidad principal de IAM cree un conector de origen de datos

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "lambda:CreateFunction",
        "lambda:ListVersionsByFunction",
        "iam:CreateRole",
        "lambda:GetFunctionConfiguration",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy",
        "lambda:PutFunctionConcurrency",
        "iam:PassRole",
        "iam:DetachRolePolicy",
        "lambda:ListTags",
        "iam:ListAttachedRolePolicies",
        "iam>DeleteRolePolicy",
        "lambda>DeleteFunction",
        "lambda:GetAlias",
        "iam:ListRolePolicies",
        "iam:GetRole",
        "iam:GetPolicy",
        "lambda:InvokeFunction",
        "lambda:GetFunction",
        "lambda:ListAliases",
        "lambda:UpdateFunctionConfiguration",
        "iam>DeleteRole",
        "lambda:UpdateFunctionCode",
        "s3:GetObject",
        "lambda:AddPermission",
        "iam:UpdateRole",
        "lambda>DeleteFunctionConcurrency",
        "lambda:RemovePermission",
        "iam:GetRolePolicy",
        "lambda:GetPolicy"
      ],
      "Resource": [
        "arn:aws:lambda:*:111122223333:function:MyAthenaLambdaFunctionsPrefix*",
        "arn:aws:s3:::awsserverlessrepo-changesets-1iiv3xa62ln3m/*",

```

```

        "arn:aws:iam::*:role/RoleName",
        "arn:aws:iam::111122223333:policy/*"
    ]
},
{
    "Sid": "VisualEditor1",
    "Effect": "Allow",
    "Action": [
        "cloudformation:CreateUploadBucket",
        "cloudformation:DescribeStackDriftDetectionStatus",
        "cloudformation:ListExports",
        "cloudformation:ListStacks",
        "cloudformation:ListImports",
        "lambda:ListFunctions",
        "iam:ListRoles",
        "lambda:GetAccountSettings",
        "ec2:DescribeSecurityGroups",
        "cloudformation:EstimateTemplateCost",
        "ec2:DescribeVpcs",
        "lambda:ListEventSourceMappings",
        "cloudformation:DescribeAccountLimits",
        "ec2:DescribeSubnets",
        "cloudformation:CreateStackSet",
        "cloudformation:ValidateTemplate"
    ],
    "Resource": "*"
},
{
    "Sid": "VisualEditor2",
    "Effect": "Allow",
    "Action": "cloudformation:*",
    "Resource": [
        "arn:aws:cloudformation:*:111122223333:stack/aws-serverless-
repository-MyCFStackPrefix/*",
        "arn:aws:cloudformation:*:111122223333:stack/
serverlessrepo-MyCFStackPrefix/*",
        "arn:aws:cloudformation:*:*:transform/Serverless-*",
        "arn:aws:cloudformation:*:111122223333:stackset/aws-serverless-
repository-MyCFStackPrefix*:*",
        "arn:aws:cloudformation:*:111122223333:stackset/
serverlessrepo-MyCFStackPrefix*:*"
    ]
},
{

```

```

        "Sid": "VisualEditor3",
        "Effect": "Allow",
        "Action": "serverlessrepo:*",
        "Resource": "arn:aws:serverlessrepo:*:*:applications/*"
    }
]
}

```

## Explicación de permisos

Acciones permitidas	Explicación
<pre> "lambda:CreateFunction", "lambda:ListVersionsByFunction", "lambda:GetFunctionConfiguration", "lambda:PutFunctionConcurrency", "lambda:ListTags", "lambda&gt;DeleteFunction", "lambda:GetAlias", "lambda:InvokeFunction", "lambda:GetFunction", "lambda:ListAliases", "lambda:UpdateFunctionConfiguration", "lambda:UpdateFunctionCode", "lambda:AddPermission", "lambda&gt;DeleteFunctionConcurrency", "lambda:RemovePermission", "lambda:GetPolicy" "lambda:GetAccountSettings", "lambda:ListFunctions", "lambda:ListEventSourceMappings", </pre>	<p>Permita la creación y administración de las funciones de Lambda enumeradas como recursos. En el ejemplo, se utiliza un prefijo de nombre en el identificador de recurso <code>arn:aws:lambda:*: <i>MyAWSAccount</i> :function: <i>MyAthenaLambdaFunctionsPrefix</i> *</code>, donde <i>MyAthenaLambdaFunctionsPrefix</i> es un prefijo compartido utilizado en el nombre de un grupo de funciones de Lambda para que no sea necesario especificarlas individualmente como recursos. Puede especificar uno o más recursos de funciones de Lambda.</p>
<pre> "s3:GetObject" </pre>	<p>Permite la lectura de un bucket que AWS Serverless Application Repository requiere según lo especificado por el identificador de recurso <code>arn:aws:s3:::awsserverlessrepo-changesets- <i>1iiv3xa62ln3m</i> /*</code>. Este bucket puede ser específico de su cuenta.</p>

Acciones permitidas	Explicación
<pre>"cloudformation:*"</pre>	<p>Permite la creación y administración de las pilas de AWS CloudFormation especificadas por el recurso <i>MyCFStackPrefix</i> . Estas pilas y conjuntos de pilas indican la forma en la que AWS Serverless Application Repository implementa los conectores y las UDF.</p>
<pre>"serverlessrepo:*"</pre>	<p>Permite buscar, ver, publicar y actualizar aplicaciones en AWS Serverless Application Repository, especificado por el identificador de recurso <code>arn:aws:serverlessrepo:*:*:applications/*</code> .</p>

## Ejemplos de políticas de permisos de IAM para permitir funciones definidas por el usuario (UDF) de Amazon Athena

Los ejemplos de políticas de permisos de este tema muestran las acciones permitidas necesarias y los recursos para las que están permitidas. Examine estas políticas detenidamente y modifíquelas de acuerdo con sus requisitos antes de adjuntar políticas de permisos similares a las identidades de IAM.

- [Example Policy to Allow an IAM Principal to Run and Return Queries that Contain an Athena UDF Statement](#)
- [Example Policy to Allow an IAM Principal to Create an Athena UDF](#)

Example : permitir que una entidad principal de IAM ejecute y devuelva consultas que contengan una instrucción UDF de Athena

La siguiente política de permisos basada en identidad permite acciones que un usuario u otra entidad principal de IAM necesitan para ejecutar consultas que usan instrucciones UDF de Athena.

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

```

    "Sid": "VisualEditor0",
    "Effect": "Allow",
    "Action": [
        "athena:StartQueryExecution",
        "lambda:InvokeFunction",
        "athena:GetQueryResults",
        "s3:ListMultipartUploadParts",
        "athena:GetWorkGroup",
        "s3:PutObject",
        "s3:GetObject",
        "s3:AbortMultipartUpload",
        "athena:StopQueryExecution",
        "athena:GetQueryExecution",
        "s3:GetBucketLocation"
    ],
    "Resource": [
        "arn:aws:athena:*:MyAWSacctId:workgroup/MyAthenaWorkGroup",
        "arn:aws:s3:::MyQueryResultsBucket/*",
        "arn:aws:lambda:*:MyAWSacctId:function:OneAthenaLambdaFunction",
        "arn:aws:lambda:*:MyAWSacctId:function:AnotherAthenaLambdaFunction"
    ]
},
{
    "Sid": "VisualEditor1",
    "Effect": "Allow",
    "Action": "athena:ListWorkGroups",
    "Resource": "*"
}
]
}

```

## Explicación de permisos

Acciones permitidas	Explicación
<pre> "athena:StartQueryExecution", "athena:GetQueryResults", "athena:GetWorkGroup", "athena:StopQueryExecution", "athena:GetQueryExecution", </pre>	<p>Permisos de Athena necesarios para ejecutar consultas en el grupo de trabajo de MyAthenaWorkGroup .</p>

Acciones permitidas	Explicación
<pre>"s3:PutObject", "s3:GetObject", "s3:AbortMultipartUpload"</pre>	<p>s3:PutObject y s3:AbortMultipartUpload permiten escribir los resultados de la consulta en todas las subcarpetas del bucket de resultados de la consulta según lo especificado por el identificador de recursos <code>arn:aws:s3::: <i>MyQueryResultsBucket</i> /*</code>, donde <i>MyQueryResultsBucket</i> es el bucket de resultados de la consulta de Athena. Para obtener más información, consulte <a href="#">Trabajo con resultados de las consultas, consultas recientes y archivos de salida</a>.</p> <p>s3:GetObject permite leer los resultados de la consulta y el historial de consultas para el recurso especificado como <code>arn:aws:s3::: <i>MyQueryResultsBucket</i></code>, donde <i>MyQueryResultsBucket</i> es el bucket de resultados de la consulta de Athena. Para obtener más información, consulte <a href="#">Trabajo con resultados de las consultas, consultas recientes y archivos de salida</a>.</p> <p>s3:GetObject también permite leer desde el recurso especificado como <code>"arn:aws:s3::: <i>MyLambdaSpillBucket</i> /<i>MyLambdaSpillPrefix</i> *"</code>, donde <i>MyLambdaSpillPrefix</i> se especifica en la configuración de la función o funciones Lambda invocadas.</p>



Acciones permitidas	Explicación
<pre>"lambda:InvokeFunction"</pre>	<p>Permite que las consultas invoquen las funciones AWS Lambda especificadas en el bloque Resource. Por ejemplo, <code>arn:aws:lambda:*: <i>MyAWSAcctId</i> :function : <i>MyAthenaLambdaFunction</i></code> , donde <i>MyAthenaLambdaFunction</i> especifica el nombre de una función Lambda que se va a invocar. Se pueden especificar varias funciones como se muestra en el ejemplo.</p>

Example : permitir que una entidad principal de IAM cree una UDF de Athena

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "lambda:CreateFunction",
        "lambda:ListVersionsByFunction",
        "iam:CreateRole",
        "lambda:GetFunctionConfiguration",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy",
        "lambda:PutFunctionConcurrency",
        "iam:PassRole",
        "iam:DetachRolePolicy",
        "lambda:ListTags",
        "iam:ListAttachedRolePolicies",
        "iam>DeleteRolePolicy",
        "lambda>DeleteFunction",
        "lambda:GetAlias",
        "iam:ListRolePolicies",
        "iam:GetRole",
        "iam:GetPolicy",
        "lambda:InvokeFunction",
        "lambda:GetFunction",

```

```

        "lambda:ListAliases",
        "lambda:UpdateFunctionConfiguration",
        "iam:DeleteRole",
        "lambda:UpdateFunctionCode",
        "s3:GetObject",
        "lambda:AddPermission",
        "iam:UpdateRole",
        "lambda>DeleteFunctionConcurrency",
        "lambda:RemovePermission",
        "iam:GetRolePolicy",
        "lambda:GetPolicy"
    ],
    "Resource": [
        "arn:aws:lambda:*:111122223333:function:MyAthenaLambdaFunctionsPrefix",
        "arn:aws:s3:::awsserverlessrepo-changesets-1iiv3xa62ln3m/*",
        "arn:aws:iam::*:role/RoleName",
        "arn:aws:iam::111122223333:policy/*"
    ]
},
{
    "Sid": "VisualEditor1",
    "Effect": "Allow",
    "Action": [
        "cloudformation:CreateUploadBucket",
        "cloudformation:DescribeStackDriftDetectionStatus",
        "cloudformation:ListExports",
        "cloudformation:ListStacks",
        "cloudformation:ListImports",
        "lambda:ListFunctions",
        "iam:ListRoles",
        "lambda:GetAccountSettings",
        "ec2:DescribeSecurityGroups",
        "cloudformation:EstimateTemplateCost",
        "ec2:DescribeVpcs",
        "lambda:ListEventSourceMappings",
        "cloudformation:DescribeAccountLimits",
        "ec2:DescribeSubnets",
        "cloudformation:CreateStackSet",
        "cloudformation:ValidateTemplate"
    ],
    "Resource": "*"
},
{

```

```

        "Sid": "VisualEditor2",
        "Effect": "Allow",
        "Action": "cloudformation:*",
        "Resource": [
            "arn:aws:cloudformation:*:111122223333:stack/aws-serverless-
repository-MyCFStackPrefix/*/*",
            "arn:aws:cloudformation:*:111122223333:stack/
serverlessrepo-MyCFStackPrefix/*/*",
            "arn:aws:cloudformation:*:*:transform/Serverless-*",
            "arn:aws:cloudformation:*:111122223333:stackset/aws-serverless-
repository-MyCFStackPrefix*:*",
            "arn:aws:cloudformation:*:111122223333:stackset/
serverlessrepo-MyCFStackPrefix*:*"
        ],
    },
    {
        "Sid": "VisualEditor3",
        "Effect": "Allow",
        "Action": "serverlessrepo:*",
        "Resource": "arn:aws:serverlessrepo:*:*:applications/*"
    }
]
}

```

## Explicación de permisos

Acciones permitidas	Explicación
<pre> "lambda:CreateFunction", "lambda:ListVersionsByFunction", "lambda:GetFunctionConfiguration", "lambda:PutFunctionConcurrency", "lambda:ListTags", "lambda&gt;DeleteFunction", "lambda:GetAlias", "lambda:InvokeFunction", "lambda:GetFunction", "lambda:ListAliases", "lambda:UpdateFunctionConfigur ation", "lambda:UpdateFunctionCode", "lambda:AddPermission", "lambda&gt;DeleteFunctionConcurrency", </pre>	<p>Permita la creación y administración de las funciones de Lambda enumeradas como recursos. En el ejemplo, se utiliza un prefijo de nombre en el identificador de recurso <code>arn:aws:lambda:*: <i>MyAWSAcctId</i> :function: <i>MyAthenaLambdaFunctionsPrefix</i> *</code>, donde <i>MyAthenaLambdaFunctionsPrefix</i> es un prefijo compartido utilizado en el nombre de un grupo de funciones de Lambda para que no sea necesario especificarlas individualmente como recursos. Puede especificar uno o más recursos de funciones de Lambda.</p>

Acciones permitidas	Explicación
<pre>"lambda:RemovePermission", "lambda:GetPolicy" "lambda:GetAccountSettings", "lambda:ListFunctions", "lambda:ListEventSourceMappings",</pre>	
<pre>"s3:GetObject"</pre>	<p>Permite la lectura de un bucket que AWS Serverless Application Repository requiere según lo especificado por el identificador de recurso <code>arn:aws:s3:::awsserverlessrepo-changesets- <i>1iiv3xa62ln3m</i> /*</code>.</p>
<pre>"cloudformation:*"</pre>	<p>Permite la creación y administración de las pilas de AWS CloudFormation especificadas por el recurso <code>MyCFStackPrefix</code>. Estas pilas y conjuntos de pilas indican la forma en la que AWS Serverless Application Repository implementa los conectores y las UDF.</p>
<pre>"serverlessrepo:*"</pre>	<p>Permite buscar, ver, publicar y actualizar aplicaciones en AWS Serverless Application Repository, especificado por el identificador de recurso <code>arn:aws:serverlessrepo:*:*:applications/*</code>.</p>

## Permiso de acceso para ML con Athena

Las entidades principales de IAM que ejecutan consultas de ML de Athena deben tener permiso para realizar la acción `sagemaker:invokeEndpoint` para los puntos de conexión de SageMaker que utilizan. Incluya una instrucción de política similar a la siguiente en las políticas de permisos basadas en identidad asociadas a identidades de usuario. Además, adjunte [Política administrada de AWS: AmazonAthenaFullAccess](#), que proporciona acceso completo a las acciones de Athena, o una política insertada modificada que permite un subconjunto de acciones.

Reemplace `arn:aws:sagemaker:region:AWSacctID:ModelEndpoint` en el ejemplo por el ARN o los ARN de los puntos de conexión del modelo que se utilizarán en las consultas. Para

obtener más información, consulte [Acciones, recursos y claves de condición de SageMaker](#) en la Referencia de autorizaciones de servicio.

```
{
    "Effect": "Allow",
    "Action": [
        "sagemaker:invokeEndpoint"
    ],
    "Resource": "arn:aws:sagemaker:us-west-2:123456789012:workteam/public-crowd/default"
}
```

Siempre que utilice políticas de IAM, asegúrese de seguir las prácticas recomendadas de IAM. Para más información, consulte [Prácticas recomendadas de seguridad en IAM](#) en la Guía del usuario de IAM.

## Habilitación de acceso federado a la API de Athena

En esta sección se trata el acceso federado que permite a un usuario o aplicación cliente en la organización llamar a operaciones de la API de Amazon Athena. En este caso, los usuarios de la organización no tienen acceso directo a Athena. En su lugar, administra credenciales de usuario fuera de AWS en Microsoft Active Directory. Active Directory es compatible con [SAML 2.0](#) (Lenguaje de marcado para confirmaciones de seguridad 2.0).

Para autenticar a los usuarios en este escenario, utilice el controlador JDBC u ODBC con compatibilidad con SAML.2.0 para obtener acceso a Active Directory Federation Services (AD FS) 3.0 y permitir que una aplicación cliente llame las operaciones de la API de Athena.

Para obtener más información sobre la compatibilidad con SAML 2.0 en AWS, consulte [Acerca de la federación SAML 2.0](#) en la Guía del usuario de IAM.

### Note

El acceso federado a la API de Athena es compatible con un determinado tipo de proveedor de identidades (IdP), el Active Directory Federation Service (ADFS 3.0), que forma parte de Windows Server. El acceso de federación no es compatible con la característica de propagación de identidades fiables del IAM Identity Center. El acceso se establece a través de las versiones de los controladores JDBC u ODBC que admiten SAML 2.0. Para obtener

más información, consulte [Conexión a Amazon Athena con JDBC](#) y [Conexión a Amazon Athena con ODBC](#).

## Temas

- [Antes de empezar](#)
- [Diagrama de arquitectura](#)
- [Procedimiento: acceso federado basado en SAML a la API de Athena](#)

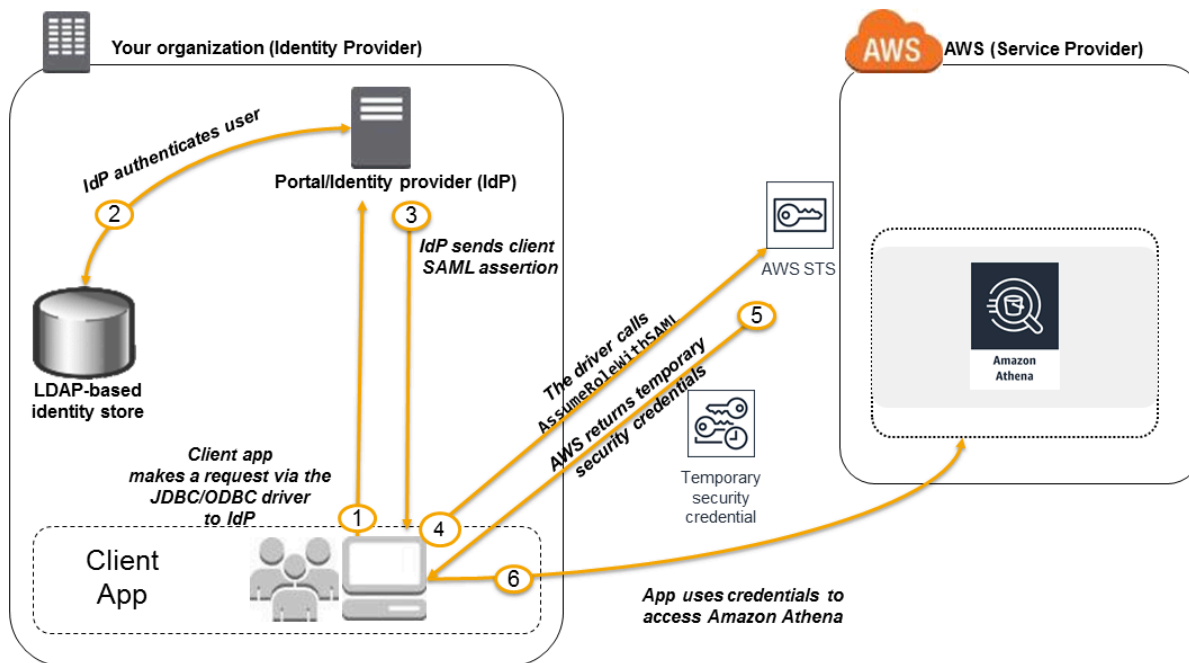
## Antes de empezar

Antes de comenzar, complete los siguientes requisitos previos:

- Dentro de su organización, instale y configure ADFS 3.0 como proveedor de identidad.
- Instale y configure las últimas versiones disponibles de los controladores JDBC u ODBC en los clientes que se utilizan para obtener acceso a Athena. El controlador debe incluir compatibilidad con el acceso federado compatible con SAML 2.0. Para obtener más información, consulte [Conexión a Amazon Athena con JDBC](#) y [Conexión a Amazon Athena con ODBC](#).

## Diagrama de arquitectura

En el siguiente diagrama se ilustra este proceso.



1. Un usuario de su organización utiliza una aplicación cliente con el controlador JDBC u ODBC para solicitar autenticación del proveedor de identidad de su organización. El proveedor de identidad es ADFS 3.0.
2. El proveedor de identidad autentica al usuario en función de Active Directory, que es el almacén de identidades de la organización.
3. El proveedor de identidad construye una aserción de SAML con información sobre el usuario y envía dicha aserción a la aplicación cliente a través del controlador JDBC u ODBC.
4. El controlador JDBC u ODBC llama a la operación API AWS Security Token Service [AssumeRoleWithSAML](#), transfiriéndole los parámetros siguientes:
  - El ARN del proveedor SAML.
  - El ARN del rol que suponer
  - La aserción SAML del proveedor de identidades

Para obtener más información, consulte [AssumeRoleWithSAML](#) en la Referencia de la API de AWS Security Token Service.

5. La respuesta de la API a la aplicación cliente a través del controlador JDBC u ODBC incluye credenciales de seguridad temporales.

6. La aplicación cliente utiliza credenciales de seguridad temporales para llamar a las operaciones de la API de Athena, lo que permite a los usuarios obtener acceso a las operaciones de la API de Athena.

Procedimiento: acceso federado basado en SAML a la API de Athena

El procedimiento establece una relación de confianza entre el proveedor de identidad de la organización y su cuenta de AWS para habilitar el acceso federado basado en SAML a la operación API de Amazon Athena.

Para habilitar el acceso federado a la API de Athena:

1. En su organización, registre AWS como proveedor de servicios (SP) en su proveedor de identidad. Este proceso se conoce como relación de confianza. Para obtener más información, consulte [Configuración de una relación de confianza para usuario autenticado en el proveedor de identidades SAML 2.0](#) en la Guía del usuario de IAM. Como parte de esta tarea, siga estos pasos:
  - a. Obtenga el documento de metadatos de SAML de muestra de esta URL: <https://signin.aws.amazon.com/static/saml-metadata.xml>.
  - b. En el proveedor de identidad de la organización (ADFS), puede generar un archivo XML de metadatos equivalente que describa su proveedor de identidad como proveedor de identidad de AWS. El archivo de metadatos debe incluir el nombre de emisor, una fecha de creación, una fecha de vencimiento y claves que AWS utiliza para validar las respuestas de autenticación (aserciones) de la organización.
2. En la consola de IAM, cree una entidad de proveedor de identidad SAML. Para obtener más información, consulte [Creación de proveedores de identidad SAML](#) en la Guía del usuario de IAM. Como parte de este paso, haga lo siguiente:
  - a. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
  - b. Cargue el documento de metadatos de SAML producido por el proveedor de identidad (ADFS) en el paso 1 de este procedimiento.
3. En la consola de IAM, cree uno o varios roles de IAM para su proveedor de identidad. Para obtener más información, consulte [Creación de un rol para un proveedor de identidad de terceros \(federación\)](#) en la Guía del usuario de IAM. Como parte de este paso, haga lo siguiente:



- En la política de permisos del rol, enumere las acciones que pueden realizar los usuarios de la organización en AWS.
- En la política de confianza del rol, establezca la entidad de proveedor de SAML que creó en el Paso 2 de este procedimiento como principal.

Esto establece una relación de confianza entre la organización y AWS.

4. En el proveedor de identidad de la organización (ADFS), defina aserciones que asignen usuarios o grupos de la organización a los roles de IAM. El mapeo de usuarios y grupos a los roles de IAM también se conoce como regla de reclamación. Tenga en cuenta que los distintos usuarios y grupos de la organización pueden asignarse a diferentes roles de IAM.

Para obtener información sobre cómo configurar la asignación en ADFS, consulte la publicación de blog: [Enabling Federation to AWS Using Windows Active Directory, ADFS, and SAML 2.0](#) (Habilitación de la federación por medio de Windows Active Directory, ADFS y SAML 2.0).

5. Instale y configure el controlador JDBC u ODBC con compatibilidad con SAML 2.0. Para obtener más información, consulte [Conexión a Amazon Athena con JDBC](#) y [Conexión a Amazon Athena con ODBC](#).
6. Especifique la cadena de conexión de la aplicación al controlador JDBC u ODBC. Para obtener más información acerca de la cadena de conexión que su aplicación debe utilizar, consulte el tema “Using the Active Directory Federation Services (ADFS) Credentials Provider” (Uso del proveedor de credenciales de los servicios de federación de Active Directory [ADFS]) en la Guía de instalación y configuración del controlador JDBC, o un tema similar en la Guía de instalación y configuración del controlador ODBC disponible como descarga en formato PDF en los temas [Conexión a Amazon Athena con JDBC](#) y [Conexión a Amazon Athena con ODBC](#).

A continuación se ofrece el resumen general de configuración de la cadena de conexión para los controladores:

1. En la `AwsCredentialsProviderClass` configuration, establezca `com.simba.athena.iamsupport.plugin.AdfsCredentialsProvider` para indicar que desea utilizar la autenticación basada en SAML 2.0 a través del proveedor de identidad ADFS.
2. Para `idp_host`, proporcione el nombre de host del servidor de proveedor de identidad de ADFS.

3. Para `idp_port`, proporcione el número de puerto en el que escucha el proveedor de identidad ADFS para la solicitud de aserción de SAML.
4. Para `UID` y `PWD`, proporcione las credenciales de usuario de dominio de AD. Cuando se utiliza el controlador en Windows, si no se proporcionan `UID` y `PWD`, el controlador intenta obtener las credenciales de usuario del usuario que ha iniciado sesión en el equipo Windows.
5. Opcionalmente, configure `ssl_insecure` en `true`. En este caso, el controlador no comprueba la autenticidad del certificado SSL para el servidor de proveedor de identidad ADFS. Es necesario definirlo en `true` si no se ha configurado que el controlador confíe en el certificado SSL del proveedor de identidad ADFS.
6. Para habilitar el mapeo de un grupo o usuario de dominio de Active Directory en uno o varios roles de IAM (tal como se indicó en el paso 4 de este procedimiento), en el `preferred_role` de la conexión JDBC u ODBC, especifique el rol de IAM (ARN) que suponer para la conexión del controlador. Especificar `preferred_role` es opcional y es útil si el rol no es el primero indicado en la regla de reclamación.

Como resultado de este procedimiento, se producen las siguientes acciones:

1. El controlador ODBC o JDBC llama a la API [AssumeRoleWithSAML](#) de AWS STS, y le pasa las aserciones, como se muestra en el paso 4 del [Diagrama de la arquitectura](#).
2. AWS se asegurará de que la solicitud para suponer que rol procede del proveedor de identidad al que se hace referencia en la entidad del proveedor SAML.
3. Si la solicitud se realiza correctamente, la operación de la API [AssumeRoleWithSAML](#) de AWS STS devuelve un conjunto de credenciales de seguridad temporales, que su aplicación cliente utiliza para realizar solicitudes firmadas a Athena.

La aplicación ahora tiene información sobre el usuario actual y puede acceder a Athena mediante programación.

## Registro y supervisión en Athena

Para detectar incidentes, recibir alertas cuando ocurran y responder a ellas, utilice estas opciones con Amazon Athena:

- Monitoreo de Athena con AWS CloudTrail: [AWS CloudTrail](#) proporciona un registro de las medidas adoptadas por un usuario, rol o Servicio de AWS en Athena. Captura las llamadas desde la consola de Athena y las llamadas de código a las operaciones de la API de Athena como

eventos. Esto le permite determinar la solicitud que se envió a Athena, la dirección IP desde la que se realizó la solicitud, quién realizó la solicitud, cuándo la realizó y detalles adicionales. Para obtener más información, consulte [Registro de las llamadas a la API de Amazon Athena con AWS CloudTrail](#).

También puede usar Athena para consultar los archivos de registro de CloudTrail no solo para Athena, sino también para otros Servicios de AWS. Para obtener más información, consulte [Consulta de registros de AWS CloudTrail](#).

- Monitoree el uso de Athena con CloudTrail y Amazon QuickSight: [Amazon QuickSight](#) es un servicio de inteligencia empresarial totalmente administrado y basado en la nube que le permite crear paneles interactivos a los que la organización puede acceder desde cualquier dispositivo. Para ver un ejemplo de una solución en la que se utiliza CloudTrail y Amazon QuickSight a fin de supervisar el uso de Athena, consulte la publicación del blog sobre macrodatos de AWS [How Realtor.com monitors Amazon Athena usage with AWS CloudTrail and Amazon QuickSight](#).
- Utilizar Amazon EventBridge con Athena: Amazon EventBridge proporciona una transmisión casi en tiempo real de eventos de sistema que describen cambios en los recursos de AWS. EventBridge descubre los cambios operativos a medida que ocurren, responde a ellos y toma medidas correctivas según sea necesario al enviar mensajes para responder al entorno, activar funciones, realizar cambios y registrar información de estado. Los eventos se emiten en la medida de lo posible. Para obtener más información, consulte [Introducción a Amazon EventBridge](#) en la Guía del usuario de Amazon EventBridge.
- Utilice grupos de trabajo para separar usuarios, equipos, aplicaciones o cargas de trabajo y establecer límites de consulta y controlar los costos de las consultas: puede ver métricas relacionadas con las consultas en Amazon CloudWatch, controlar los costos de las consultas mediante la configuración de los límites de la cantidad de datos escaneados, crear umbrales y desencadenar acciones, como alarmas de Amazon SNS, cuando se superan estos umbrales. Para un procedimiento de alto nivel, consulte [Configuración de los grupos de trabajo](#). Utilice permisos de IAM de nivel de recursos para controlar el acceso a un grupo de trabajo específico. Para obtener más información, consulte [Uso de grupos de trabajo para la ejecución de consultas](#) y [Control de costos y supervisión de consultas con métricas y eventos de CloudWatch](#).

## Temas

- [Registro de las llamadas a la API de Amazon Athena con AWS CloudTrail](#)

## Registro de las llamadas a la API de Amazon Athena con AWS CloudTrail

Athena está integrada con AWS CloudTrail, un servicio que proporciona un registro de las acciones del usuario, el rol o un Servicio de AWS en Athena.

CloudTrail captura todas las llamadas a la API de Athena como eventos. Las llamadas capturadas incluyen las llamadas desde la consola de Athena y las llamadas desde el código a las operaciones de la API de Athena. Si crea un registro de seguimiento, puede habilitar la entrega continua de eventos de CloudTrail a un bucket de Amazon S3, incluidos los eventos para Athena. Si no configura un registro de seguimiento, puede ver los eventos más recientes en la consola de CloudTrail en el Historial de eventos.

Mediante la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a Athena, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo la realizó y los detalles adicionales.

Para más información sobre CloudTrail, consulte la [Guía del usuario de AWS CloudTrail](#).

Puede usar Athena para consultar archivos de registro de CloudTrail desde Athena en sí y desde otros Servicios de AWS. Para obtener más información, consulte [Consulta de registros de AWS CloudTrail](#), [El SerDe JSON de Hive](#) y la publicación del Blog de macrodatos de AWS [Utilizar instrucciones CTAS con Amazon Athena para reducir los costos y mejorar el rendimiento](#), que utiliza CloudTrail para proporcionar información sobre el uso de Athena.

### Información de Athena en CloudTrail

CloudTrail se habilita en su cuenta de Amazon Web Services cuando crea la cuenta. Cuando se produce una actividad en Athena, esa actividad se registra en un evento de CloudTrail junto con otros eventos de servicio de AWS en el Historial de eventos. Puede ver, buscar y descargar los últimos eventos de la cuenta de Amazon Web Services. Para obtener más información, consulte [Ver eventos con el historial de eventos de CloudTrail](#).

Para mantener un registro continuo de eventos en la cuenta de Amazon Web Services, incluidos los eventos de Athena, cree un registro de seguimiento. Un registro de seguimiento permite a CloudTrail enviar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las Regiones de AWS. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de Amazon S3 especificado. También es posible configurar otros Servicios de AWS para analizar en profundidad y actuar en función de los datos de eventos recopilados en los registros de CloudTrail. Para más información, consulte los siguientes temas:

- [Introducción a la creación de registros de seguimiento](#)
- [Servicios e integraciones compatibles con CloudTrail](#)
- [Configuración de notificaciones de Amazon SNS para CloudTrail](#)
- [Recibir archivos de registro de CloudTrail de varias regiones](#) y [Recibir archivos de registro de CloudTrail de varias cuentas](#)

CloudTrail registra todas las acciones de Athena, que están documentadas en la [Referencia de la API de Amazon Athena](#). Por ejemplo, las llamadas a las acciones [StartQueryExecution](#) y [GetQueryResults](#) generan entradas en los archivos de registro de CloudTrail.

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario lo ayuda a determinar lo siguiente:

- Si la solicitud se realizó con credenciales de usuario de AWS Identity and Access Management (IAM) o credenciales de usuario raíz.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro Servicio de AWS.

Para más información, consulte [Elemento userIdentity de CloudTrail](#).

## Comprensión de las entradas de archivos de registro de Athena

Un registro de seguimiento es una configuración que permite la entrega de eventos como archivos de registros en un bucket de Amazon S3 que especifique. Los archivos de registro de CloudTrail pueden contener una o varias entradas de registro. Un evento representa una solicitud específica realizada desde un origen cualquiera, y contiene información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. Los archivos de registro de CloudTrail no rastrean el orden en la pila de las llamadas públicas a la API, por lo que estas no aparecen en ningún orden específico.

### Note

Para evitar la divulgación involuntaria de información confidencial, la entrada `queryString` en los registros `StartQueryExecution` y `CreateNamedQuery` tiene un valor de `***OMITTED***`. Este comportamiento es así por diseño. Para acceder a la cadena

de consulta real, puede utilizar la API [GetQueryExecution](#) de Athena y pasar el valor de `responseElements.queryExecutionId` del registro de CloudTrail.

Los siguientes son ejemplos de entradas de registro de CloudTrail para:

- [StartQueryExecution \(correcto\)](#)
- [StartQueryExecution \(error\)](#)
- [CreateNamedQuery](#)

StartQueryExecution (correcto)

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EXAMPLE_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:user/johndoe",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "johndoe"
  },
  "eventTime": "2017-05-04T00:23:55Z",
  "eventSource": "athena.amazonaws.com",
  "eventName": "StartQueryExecution",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "77.88.999.69",
  "userAgent": "aws-internal/3",
  "requestParameters": {
    "clientRequestToken": "16bc6e70-f972-4260-b18a-db1b623cb35c",
    "resultConfiguration": {
      "outputLocation": "s3://DOC-EXAMPLE-BUCKET/test/"
    },
    "queryString": "****OMITTED****"
  },
  "responseElements": {
    "queryExecutionId": "b621c254-74e0-48e3-9630-78ed857782f9"
  },
  "requestID": "f5039b01-305f-11e7-b146-c3fc56a7dc7a",
  "eventID": "c97cf8c8-6112-467a-8777-53bb38f83fd5",
  "eventType": "AwsApiCall",
}
```

```
"recipientAccountId":"123456789012"  
}
```

## StartQueryExecution (error)

```
{  
  "eventVersion":"1.05",  
  "userIdentity":{  
    "type":"IAMUser",  
    "principalId":"EXAMPLE_PRINCIPAL_ID",  
    "arn":"arn:aws:iam::123456789012:user/johndoe",  
    "accountId":"123456789012",  
    "accessKeyId":"EXAMPLE_KEY_ID",  
    "userName":"johndoe"  
  },  
  "eventTime":"2017-05-04T00:21:57Z",  
  "eventSource":"athena.amazonaws.com",  
  "eventName":"StartQueryExecution",  
  "awsRegion":"us-east-1",  
  "sourceIPAddress":"77.88.999.69",  
  "userAgent":"aws-internal/3",  
  "errorCode":"InvalidRequestException",  
  "errorMessage":"Invalid result configuration. Should specify either output location or  
result configuration",  
  "requestParameters":{  
    "clientRequestToken":"ca0e965f-d6d8-4277-8257-814a57f57446",  
    "queryString":"***OMITTED***"  
  },  
  "responseElements":null,  
  "requestID":"aefbc057-305f-11e7-9f39-bbc56d5d161e",  
  "eventID":"6e1fc69b-d076-477e-8dec-024ee51488c4",  
  "eventType":"AwsApiCall",  
  "recipientAccountId":"123456789012"  
}
```

## CreateNamedQuery

```
{  
  "eventVersion":"1.05",  
  "userIdentity":{  
    "type":"IAMUser",  
    "principalId":"EXAMPLE_PRINCIPAL_ID",  
    "arn":"arn:aws:iam::123456789012:user/johndoe",
```

```
"accountId":"123456789012",
"accessKeyId":"EXAMPLE_KEY_ID",
"userName":"johndoe"
},
"eventTime":"2017-05-16T22:00:58Z",
"eventSource":"athena.amazonaws.com",
"eventName":"CreateNamedQuery",
"awsRegion":"us-west-2",
"sourceIPAddress":"77.88.999.69",
"userAgent":"aws-cli/1.11.85 Python/2.7.10 Darwin/16.6.0 botocore/1.5.48",
"requestParameters":{
  "name":"johndoetest",
  "queryString":"***OMITTED***",
  "database":"default",
  "clientRequestToken":"fc1ad880-69ee-4df0-bb0f-1770d9a539b1"
},
"responseElements":{
  "namedQueryId":"cdd0fe29-4787-4263-9188-a9c8db29f2d6"
},
"requestID":"2487dd96-3a83-11e7-8f67-c9de5ac76512",
"eventID":"15e3d3b5-6c3b-4c7c-bc0b-36a8dd95227b",
"eventType":"AwsApiCall",
"recipientAccountId":"123456789012"
},
```

## Validación de la conformidad para Amazon Athena

Audidores externos evalúan la seguridad y la conformidad de Amazon Athena en numerosos programas de conformidad de AWS. Estos incluye SOC, PCI, FedRAMP y otros.

Para obtener una lista de Servicios de AWS en el ámbito de programas de conformidad específicos, consulte los [Servicios de AWS en el ámbito del programa de conformidad](#). Para obtener información general, consulte [Programas de conformidad de AWS](#).

Puede descargar los informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte [Descarga de informes en AWS Artifact](#).

Su responsabilidad en el ámbito de la conformidad al usar Athena viene determinada por la confidencialidad de los datos, los objetivos de conformidad de su empresa y las leyes y regulaciones aplicables. AWS proporciona los siguientes recursos para ayudarlo con los requisitos de conformidad:



- [Guías de inicio rápido de seguridad y conformidad](#): estas guías de implementación tratan consideraciones sobre arquitectura y ofrecen pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- [Arquitectura de las medidas de seguridad y cumplimiento de HIPAA en Amazon Web Services](#): en este documento técnico, se describe cómo las empresas pueden utilizar AWS para crear aplicaciones que cumplan los requisitos de la HIPAA.
- [Recursos de conformidad de AWS](#): este conjunto de manuales y guías podría aplicarse a su sector y ubicación.
- [AWS Config](#): este Servicio de AWS evalúa en qué medida las configuraciones de los recursos cumplen las prácticas internas, las directrices del sector y la normativa.
- [AWS Security Hub](#): este Servicio de AWS proporciona una vista integral de su estado de seguridad en AWS que lo ayuda a verificar la conformidad con los estándares y las prácticas recomendadas del sector de seguridad.

## Resiliencia en Athena

La infraestructura global de AWS se divide en Regiones de AWS y zonas de disponibilidad. Las Regiones de AWS proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja latencia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre zonas de disponibilidad sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

Para obtener más información sobre las Regiones de AWS y las zonas de disponibilidad, consulte [Infraestructura global de AWS](#).

Además de la infraestructura global de AWS, Athena ofrece varias características que lo ayudan con sus necesidades de resiliencia y copia de seguridad de los datos.

Athena no usa servidor, por lo que no hay ninguna infraestructura que configurar ni administrar. Athena tiene alta disponibilidad y ejecuta consultas utilizando recursos informáticos en varias zonas de disponibilidad; además, enruta automáticamente las consultas de forma adecuada si no se puede obtener acceso a una zona de disponibilidad concreta. Athena utiliza Amazon S3 como su almacén de datos subyacente. De este modo, sus datos tienen alta disponibilidad y son duraderos. Amazon S3 proporciona una infraestructura duradera para almacenar datos importantes y se ha diseñado

para ofrecer una durabilidad del 99,999999999 % de los objetos. Sus datos se almacenan de forma redundante en varias instalaciones y en diferentes dispositivos dentro de ellas.

## Seguridad de la infraestructura en Athena

Como se trata de un servicio administrado, Amazon Athena se encuentra protegido por la seguridad de red global de AWS. Para obtener información sobre los servicios de seguridad de AWS y cómo AWS protege la infraestructura, consulte [Seguridad en la nube de AWS](#). Para diseñar su entorno de AWS con las prácticas recomendadas de seguridad de infraestructura, consulte [Protección de la infraestructura](#) en Portal de seguridad de AWS Well-Architected Framework.

Puede utilizar llamadas a la API publicadas en AWS para obtener acceso a Athena a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puede utilizar [AWS Security Token Service](#) (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Utilice políticas de política de IAM para restringir el acceso a las operaciones de Athena. Siempre que utilice políticas de IAM, asegúrese de seguir las prácticas recomendadas de IAM. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM.

Las [políticas administradas](#) por Athena son fáciles de utilizar y se actualizan automáticamente con las acciones necesarias a medida que el servicio evoluciona. Las políticas administradas por el cliente e insertadas le permiten ajustar las políticas especificando más acciones de Athena detalladas en la política. Conceda un acceso adecuado a la ubicación de Amazon S3 de los datos. Para obtener información detallada y casos sobre cómo otorgar acceso a Simple Storage Service (Amazon S3), consulte [Explicaciones de ejemplo: Administración de acceso](#) en la Guía para desarrolladores de Amazon Simple Storage Service. Para obtener más información y un ejemplo de las acciones de Amazon S3 que debe permitir, consulte el ejemplo de política de bucket en [Acceso entre cuentas](#).

## Temas

- [Conexión a Amazon Athena mediante un punto de conexión de VPC de tipo interfaz](#)

### Conexión a Amazon Athena mediante un punto de conexión de VPC de tipo interfaz

Puede mejorar la posición de seguridad de su VPC mediante el uso de un [punto de conexión de VPC de interfaz \(AWS PrivateLink\)](#) y un [punto de conexión de VPC de AWS Glue](#) en su nube privada virtual (VPC). Un punto de conexión de VPC de interfaz mejora la seguridad, ya que le permite controlar a qué destinos se puede llegar desde dentro de su VPC. Cada punto de conexión de VPC está representado por una o varias [Interfases de red elásticas](#) (ENI) con direcciones IP privadas en las subredes de la VPC.

El punto de enlace de la VPC de tipo interfaz conecta directamente la VPC con Athena sin necesidad de gateway de Internet, dispositivos NAT, conexiones de VPN ni conexiones de AWS Direct Connect. Las instancias de la VPC no necesitan direcciones IP públicas para comunicarse con la API de Athena.

Para utilizar Athena a través de la VPC, debe conectarse desde una instancia que está dentro de la VPC o conectar su red privada a la VPC a través de una red privada virtual (VPN) de Amazon o AWS Direct Connect. Para obtener más información sobre Amazon VPN, consulte [Conexiones VPN](#) en la Guía del usuario de Amazon Virtual Private Cloud. Para obtener información sobre AWS Direct Connect, consulte [Creación de una conexión](#) en la Guía del usuario de AWS Direct Connect.

Athena admite puntos de conexión de VPC en todas las Regiones de AWS en las que estén disponibles [Amazon VPC](#) y [Athena](#).

Puede crear un punto de enlace de la VPC de tipo interfaz para conectarse a Athena mediante AWS Management Console o comandos de AWS Command Line Interface (AWS CLI). Para obtener más información, consulte [Creación de un punto de conexión de interfaz](#).

Después de crear un punto de enlace de la VPC de tipo interfaz, si habilita nombres de host de [DNS privados](#) para el punto de enlace, el punto de enlace predeterminado de Athena (<https://athena.Region.amazonaws.com>) se resuelve en el punto de enlace de la VPC.

Si no habilita nombres de host de DNS privados, Amazon VPC proporciona un nombre de punto de enlace de DNS que puede utilizar en el siguiente formato:

```
VPC_Endpoint_ID.athena.Region.vpce.amazonaws.com
```

Para obtener más información, consulte [Puntos de conexión de VPC de tipo interfaz \(AWS PrivateLink\)](#) en la Guía del usuario de Amazon VPC.

Athena admite la realización de llamadas a todas sus [acciones de API](#) dentro de su VPC.

Crear una política de punto de conexión de VPC para Athena

Puede crear una política para los puntos de conexión de VPC de Amazon para Athena a fin de especificar restricciones como las siguientes:

- Entidad principal: la entidad principal que puede realizar acciones.
- Acciones: las acciones que se pueden realizar.
- Recursos: los recursos en los que se pueden llevar a cabo las acciones.
- Solo identidades de confianza: utilice la condición `aws:PrincipalOrgId` para restringir el acceso solo a las credenciales que forman parte de su organización de AWS. Esto puede ayudar a evitar el acceso no previsto por parte de entidades principales no deseadas.
- Solo recursos de confianza: utilice la condición `aws:ResourceOrgId` para evitar el acceso a recursos no deseados.
- Solo identidades y recursos de confianza: cree una política combinada para un punto de conexión de VPC que ayude a evitar el acceso a entidades principales y recursos no deseados.

Para obtener más información, consulte [Control del acceso a los servicios con puntos de conexión de VPC](#) en la Guía del usuario de Amazon VPC y el [Apéndice 2: ejemplos de políticas de puntos de conexión de VPC](#) en el documento técnico de AWS Creación de un perímetro de datos en AWS.

Example Política de punto de conexión de VPC

En el siguiente ejemplo se permiten las solicitudes por identidades de la organización a los recursos de la organización y se permiten las solicitudes por parte de las entidades principales del servicio de AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRequestsByOrgsIdentitiesToOrgsResources",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      }
    }
  ]
}
```

```

    },
    "Action": "*",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:PrincipalOrgID": "my-org-id",
        "aws:ResourceOrgID": "my-org-id"
      }
    }
  },
  {
    "Sid": "AllowRequestsByAWSServicePrincipals",
    "Effect": "Allow",
    "Principal": {
      "AWS": "*"
    },
    "Action": "*",
    "Resource": "*",
    "Condition": {
      "Bool": {
        "aws:PrincipalIsAWSService": "true"
      }
    }
  }
]
}

```

Siempre que utilice políticas de IAM, compruebe que sigue las mejores prácticas IAM. Para más información, consulte [Prácticas recomendadas de seguridad en IAM](#) en la Guía del usuario de IAM.

### Subredes compartidas

No puede crear, describir, modificar ni eliminar puntos de conexión de VPC en subredes que se compartan con usted. No obstante, puede usar los puntos de conexión de VPC en las subredes que se compartan con usted. Para obtener información sobre el uso compartido de VPC, consulte [Compartir su VPC con otras cuentas](#) en la Guía del usuario de Amazon VPC.

## Configuración y análisis de vulnerabilidades en Athena

Athena no tiene servidor, por lo que no hay una infraestructura para configurar o administrar. AWS gestiona las tareas de seguridad básicas como el sistema operativo (SO) invitado y la aplicación de parches en la base de datos, la configuración del firewall y la recuperación de desastres. Estos

procedimientos han sido revisados y certificados por los terceros pertinentes. Para obtener más detalles, consulte los siguientes recursos de AWS:

- [Modelo de responsabilidad compartida](#)
- [Prácticas recomendadas para seguridad, identidad y conformidad](#)

## Uso de Athena para consultar datos registrados en AWS Lake Formation

[AWS Lake Formation](#) le permite definir y aplicar políticas de acceso en el nivel de columnas, bases de datos y tablas al utilizar las consultas de Athena para leer datos almacenados en Amazon S3. Lake Formation proporciona una capa de autorización y gobernanza sobre datos almacenados en Amazon S3. Es posible utilizar una jerarquía de permisos en Lake Formation para conceder o revocar permisos para leer objetos del catálogo de datos como bases de datos, tablas y columnas. Lake Formation simplifica la administración de permisos y le permite implementar un control de acceso detallado (FGAC, fine-grained access control) para los datos.

Se puede utilizar Athena para consultar tanto los datos registrados en Lake Formation como los datos que no están registrados en Lake Formation.

Los permisos de Lake Formation se aplican al utilizar Athena para consultar datos de origen desde ubicaciones de Amazon S3 registradas en Lake Formation. Estos permisos también se aplican al crear bases de datos y tablas que apuntan a ubicaciones de datos de Amazon S3 registradas. Para utilizar Athena con datos registrados mediante Lake Formation, se debe configurar Athena para utilizar el AWS Glue Data Catalog.

Los permisos de Lake Formation no se aplican al escribir objetos en Amazon S3, ni al consultar datos almacenados en Amazon S3 o metadatos que no están registrados en Lake Formation. Para los datos de origen en Amazon S3 y los metadatos que no están registrados en Lake Formation, el acceso se determina mediante las políticas de permisos de IAM para Amazon S3 y acciones de AWS Glue. Las ubicaciones de resultados de consultas de Athena en Amazon S3 no se pueden registrar con Lake Formation y las políticas de permisos de IAM para el acceso de control de Amazon S3. Además, los permisos de Lake Formation no se aplican al historial de consultas de Athena. Puede utilizar grupos de trabajo de Athena para controlar el acceso al historial de consultas.

Para obtener más información acerca de Lake Formation, consulte [Preguntas frecuentes sobre Lake Formation](#) en la [Guía para desarrolladores de AWS Lake Formation](#).

### Temas

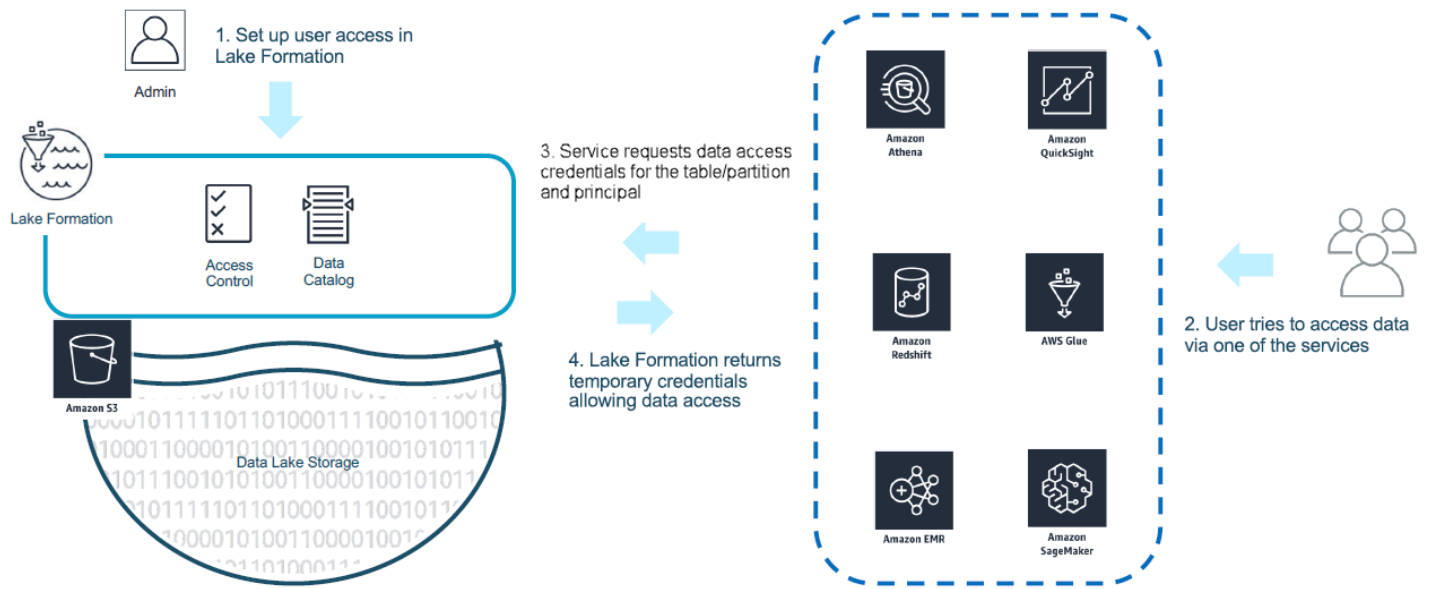
- [Cómo accede Athena a los datos registrados en Lake Formation](#)
- [Consideraciones y limitaciones al momento de utilizar Athena para consultar datos registrados en Lake Formation](#)
- [Administración de permisos de usuario de Lake Formation y Athena](#)
- [Aplicación de permisos de Lake Formation a bases de datos y tablas existentes](#)
- [Uso de Lake Formation y los controladores JDBC y ODBC de Athena para el acceso federado a Athena](#)

## Cómo accede Athena a los datos registrados en Lake Formation

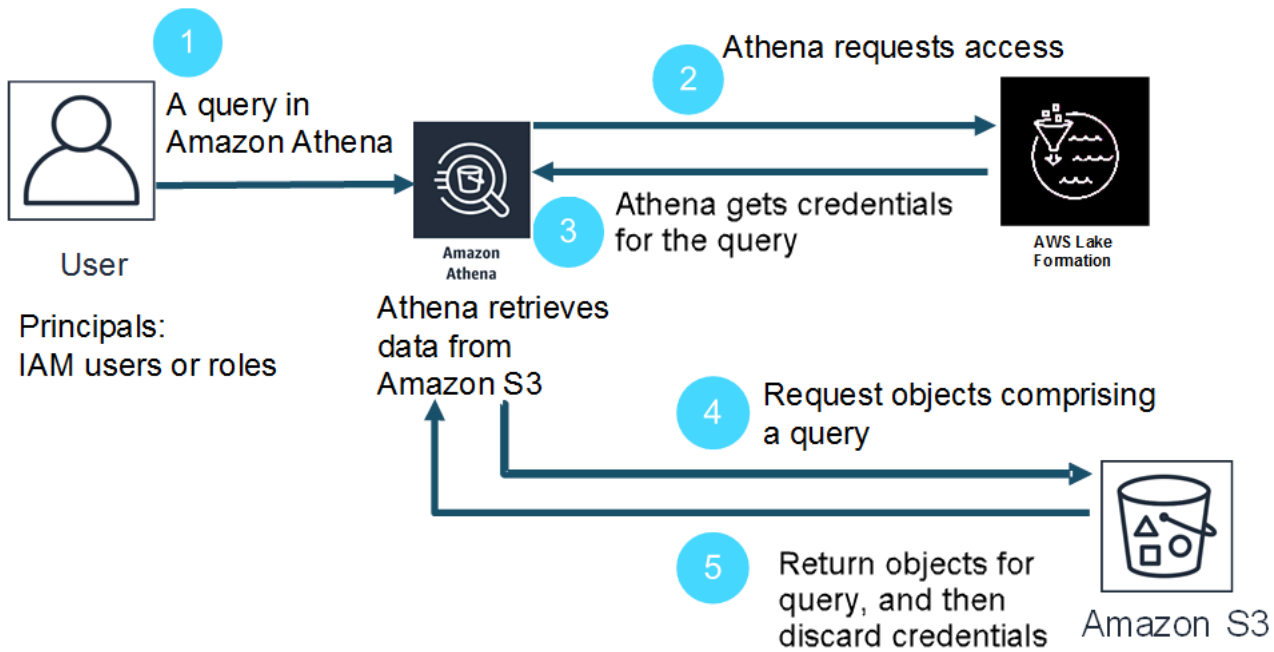
El flujo de trabajo de acceso descrito en esta sección solo se aplica cuando se ejecutan consultas de Athena en ubicaciones de Amazon S3 y objetos de metadatos que están registrados en Lake Formation. Para obtener más información, consulte [Registro de un lago de datos](#) en la Guía para desarrolladores de AWS Lake Formation. Además de registrar los datos, el administrador de Lake Formation aplica permisos de Lake Formation que conceden o revocan el acceso a los metadatos en el catálogo de datos y la ubicación de datos en Amazon S3. Para obtener más información, consulte [Seguridad y control de acceso a metadatos y datos](#) en la Guía para desarrolladores de AWS Lake Formation.

Cada vez que una entidad principal de Athena (usuario, grupo o rol) ejecuta una consulta sobre los datos registrados mediante Lake Formation, Lake Formation verifica que la entidad principal tenga los permisos de Lake Formation adecuados para la base de datos, la tabla y la ubicación de Amazon S3, según corresponda para la consulta. Si la entidad principal tiene acceso, Lake Formation ofrece credenciales temporales a Athena y se ejecuta la consulta.

En el siguiente diagrama se ilustra el flujo descrito anteriormente.



En el siguiente diagrama, se muestra cómo funciona la venta de credenciales en Athena consulta por consulta para una consulta SELECT hipotética en una tabla con una ubicación de Amazon S3 registrada en Lake Formation:



1. Una entidad principal ejecuta una consulta SELECT en Athena.
2. Athena analiza la consulta y verifica los permisos de Lake Formation para ver si se ha concedido a la entidad principal acceso a la tabla y a las columnas de la tabla.



3. Si la entidad principal tiene acceso, Athena solicita credenciales de Lake Formation. Si la entidad principal no tiene acceso, Athena emite un error de acceso denegado.
4. Lake Formation emite credenciales a Athena para utilizarlas al leer datos de Amazon S3, junto con la lista de las columnas permitidas.
5. Athena utiliza las credenciales temporales de Lake Formation para consultar los datos de Amazon S3. Una vez completada la consulta, Athena descarta las credenciales.

## Consideraciones y limitaciones al momento de utilizar Athena para consultar datos registrados en Lake Formation

Tenga en cuenta lo siguiente al utilizar Athena para consultar los datos registrados en Lake Formation. Para obtener información adicional, consulte [Problemas conocidos de AWS Lake Formation](#) en la Guía para desarrolladores de AWS Lake Formation.

### Condiciones y limitaciones

- [Metadatos de columna visibles para usuarios no autorizados en algunas circunstancias con Avro y SerDe personalizado](#)
- [Uso de permisos de Lake Formation para vistas](#)
- [Control de acceso detallado de Lake Formation y grupos de trabajo de Athena](#)
- [Ubicación de los resultados de consulta de Athena en Simple Storage Service \(Amazon S3\) no registrados en Lake Formation](#)
- [Uso de grupos de trabajo de Athena para limitar el acceso al historial de consultas](#)
- [Acceso entre cuentas al catálogo de datos](#)
- [Ubicaciones de Amazon S3 cifradas con CSE-KMS registradas en Lake Formation](#)
- [Las ubicaciones de datos particionadas registradas en Lake Formation deben estar en subdirectorios de tablas](#)
- [Las consultas Create Table As Select \(CTAS\) requieren permisos de escritura de Amazon S3](#)
- [El permiso DESCRIBE es necesario en la base de datos predeterminada](#)

Metadatos de columna visibles para usuarios no autorizados en algunas circunstancias con Avro y SerDe personalizado

La autorización de nivel de columna de Lake Formation impide que los usuarios obtengan acceso a los datos de las columnas para las que el usuario no tiene permisos de Lake Formation. Sin

embargo, en determinadas situaciones, los usuarios pueden obtener acceso a los metadatos que describen todas las columnas de la tabla, incluidas las columnas para las que no tienen permisos sobre los datos.

Esto ocurre cuando los metadatos de columna se almacenan en propiedades de tabla para tablas que utilizan el formato de almacenamiento Apache Avro o un serializador/deserializador (SerDe) personalizado en el que el esquema de tabla se define en las propiedades de la tabla junto con la definición del SerDe. Cuando utilice Athena con Lake Formation, le recomendamos que revise el contenido de las propiedades de la tabla que registra con Lake Formation y, siempre que sea posible, limite la información almacenada en las propiedades de la tabla para evitar que los metadatos confidenciales sean visibles para los usuarios.

### Uso de permisos de Lake Formation para vistas

Para los datos registrados en Lake Formation, un usuario de Athena puede crear una VIEW solo si tiene permisos de Lake Formation para las tablas, las columnas y las ubicaciones de datos de Amazon S3 de origen en los que se basa la VIEW. Después de crear una VIEW en Athena, los permisos de Lake Formation se pueden aplicar a la VIEW. Los permisos de nivel de columna no están disponibles para un VIEW. Los usuarios que tienen permisos de Lake Formation para una VIEW pero no tienen permisos para la tabla y las columnas en las que se basó la vista no pueden utilizar la VIEW para consultar datos. Sin embargo, los usuarios con esta combinación de permisos pueden utilizar instrucciones como `DESCRIBE VIEW`, `SHOW CREATE VIEW`, y `SHOW COLUMNS` para ver metadatos de VIEW. Por este motivo, asegúrese de alinear los permisos de Lake Formation para cada VIEW con los permisos de tabla subyacentes. Los filtros de celda definidos en una tabla no se aplican a una VIEW para esa tabla. Los nombres de enlace de recursos deben tener el mismo nombre que el recurso de la cuenta de origen. Existen limitaciones adicionales al trabajar con vistas en una configuración multicuenta. Para obtener más información sobre cómo configurar los permisos para vistas compartidas entre cuentas, consulte [Acceso entre cuentas al catálogo de datos](#).

### Control de acceso detallado de Lake Formation y grupos de trabajo de Athena

Los usuarios del mismo grupo de trabajo de Athena pueden ver los datos que el control de acceso detallado de Lake Formation ha configurado para que el grupo de trabajo pueda acceder a ellos. Para obtener más información sobre el uso de un control de acceso detallado en Lake Formation, consulte [Manage fine-grained access control using AWS Lake Formation](#) en el Blog de macrodatos de AWS.

## Ubicación de los resultados de consulta de Athena en Simple Storage Service (Amazon S3) no registrados en Lake Formation

Las ubicaciones de los resultados de consulta en Amazon S3 para Athena no se pueden registrar en Lake Formation. Los permisos de Lake Formation no limitan el acceso a estas ubicaciones. A menos que limite el acceso, los usuarios de Athena pueden acceder a los archivos de resultados de consultas y metadatos cuando no tienen permisos de Lake Formation para los datos. Para evitar esto, le recomendamos que utilice grupos de trabajo para especificar la ubicación de los resultados de consulta y alinear la pertenencia a grupos de trabajo con permisos de Lake Formation. A continuación, puede utilizar políticas de permisos de IAM para limitar el acceso a ubicaciones de resultados de consulta. Para obtener más información sobre los resultados de consultas, consulte [Trabajo con resultados de las consultas, consultas recientes y archivos de salida](#).

## Uso de grupos de trabajo de Athena para limitar el acceso al historial de consultas

El historial de consultas de Athena expone una lista de consultas guardadas y las cadenas de consultas completas. A menos que utilice grupos de trabajo para separar el acceso a los historiales de consultas, los usuarios de Athena que no tengan autorización para consultar datos en Lake Formation pueden ver cadenas de consulta ejecutadas en esos datos, incluidos nombres de columna, criterios de selección, etc. Le recomendamos que utilice grupos de trabajo para separar los historiales de consultas y alinear la pertenencia al grupo de trabajo de Athena con permisos de Lake Formation para limitar el acceso. Para obtener más información, consulte [Uso de grupos de trabajo para controlar el acceso a las consultas y los costos](#).

## Acceso entre cuentas al catálogo de datos

Para acceder a un catálogo de datos en otra cuenta, puede utilizar la característica AWS Glue entre cuentas de Athena o configurar el acceso entre cuentas en Lake Formation.

## Acceso entre cuentas al catálogo de datos de Athena

Puede utilizar la característica de catálogo de AWS Glue entre cuentas de Athena para registrar el catálogo en su cuenta. Esta capacidad está disponible únicamente en la versión 2 o una versión posterior del motor Athena y está limitada al uso de la misma región entre cuentas. Para obtener más información, consulte [Registro de un AWS Glue Data Catalog de otra cuenta](#).

Si el catálogo de datos que se compartirá tiene una política de recursos configurada en AWS Glue, debe actualizarse para permitir el acceso a AWS Resource Access Manager y conceder permisos a la cuenta B para utilizar el catálogo de datos de la cuenta A, como en el siguiente ejemplo.

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Service": "ram.amazonaws.com"
    },
    "Action": "glue:ShareResource",
    "Resource": [
      "arn:aws:glue:<REGION>:<ACCOUNT-A>:table/*/*",
      "arn:aws:glue:<REGION>:<ACCOUNT-A>:database/*",
      "arn:aws:glue:<REGION>:<ACCOUNT-A>:catalog"
    ]
  },
  {
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::<ACCOUNT-B>:root"
    },
    "Action": "glue:*",
    "Resource": [
      "arn:aws:glue:<REGION>:<ACCOUNT-A>:table/*/*",
      "arn:aws:glue:<REGION>:<ACCOUNT-A>:database/*",
      "arn:aws:glue:<REGION>:<ACCOUNT-A>:catalog"
    ]
  }
]
}

```

Para obtener más información, consulte [Acceso entre cuentas a los catálogos de datos de AWS Glue](#).

## Configuración del acceso entre cuentas en Lake Formation

AWS Lake Formation le permite utilizar una sola cuenta para administrar un catálogo de datos central. Puede utilizar esta característica para implementar el [acceso entre cuentas](#) Acceder los metadatos del catálogo de datos y los datos subyacentes. Por ejemplo, una cuenta de propietario puede conceder permiso SELECT en una tabla a otra cuenta (destinatario).

Para que aparezca una base de datos compartida o una tabla en el Editor de consultas de Athena, debe [crear un enlace de recursos](#) en Lake Formation a la base de datos o tabla compartida. Cuando

la cuenta de destinatario en Lake Formation consulta la tabla del propietario, [CloudTrail](#) agrega el evento de acceso a datos a los registros de la cuenta de destinatario y de la cuenta de propietario.

Para las vistas compartidas, tenga en cuenta los siguientes puntos:

- Las consultas se ejecutan en enlaces de recursos de destino, no en la vista o la tabla de origen y luego, la salida se comparte en la cuenta de destino.
- No es suficiente compartir solo la vista. Todas las tablas que participan en la creación de la vista deben formar parte del recurso compartido entre cuentas.
- El nombre del enlace de recursos creado en los recursos compartidos debe coincidir con el nombre del recurso en la cuenta de propietario. Si el nombre no coincide, aparece un mensaje de error como el siguiente Failed analyzing stored view 'awsdatacatalog.my-lf-resource-link.my-lf-view': line 3:3: Schema *schema\_name* does not exist (Se produjo un error al momento de analizar la vista almacenada 'awsdatacatalog.my-lf-resource-link.my-lf-view': line 3:3: El esquema *schema\_name* no existe)

Para obtener más información sobre el acceso entre cuentas en Lake Formation, consulte los siguientes recursos en la Guía para desarrolladores de AWS Lake Formation:

[Acceso entre cuentas](#)

[Cómo funcionan los enlaces de recursos en Lake Formation](#)

[Registro entre cuentas de CloudTrail](#)

Ubicaciones de Amazon S3 cifradas con CSE-KMS registradas en Lake Formation

Las tablas de formato de tabla abierta (OTF), como Apache Iceberg, que tienen las siguientes características no se pueden consultar con Athena:

- Las tablas se basan en ubicaciones de datos de Amazon S3 registradas en Lake Formation.
- Los objetos de Amazon S3 se cifran mediante el cifrado del cliente (CSE).
- El cifrado utiliza claves administradas por el cliente de AWS KMS (CSE\_KMS).

Para consultar tablas que no son OTF (que están cifradas con una clave CSE\_KMS), añada el siguiente bloque a la política de la clave de AWS KMS que utiliza para el cifrado CSE.

*<KMS\_KEY\_ARN>* es el ARN de la clave de AWS KMS que cifra los datos. *<IAM-ROLE-ARN>* es el ARN del rol de IAM que registra la ubicación de Amazon S3 en Lake Formation.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "AWS": "*"
  },
  "Action": "kms:Decrypt",
  "Resource": "<KMS-KEY-ARN>",
  "Condition": {
    "ArnLike": {
      "aws:PrincipalArn": "<IAM-ROLE-ARN>"
    }
  }
}
```

Las ubicaciones de datos particionadas registradas en Lake Formation deben estar en subdirectorios de tablas

Las tablas particionadas registradas en Lake Formation deben tener datos particionados en directorios que sean subdirectorios de la tabla en Amazon S3. Por ejemplo, una tabla con la ubicación `s3://DOC-EXAMPLE-BUCKET/mytable` y las particiones `s3://DOC-EXAMPLE-BUCKET/mytable/dt=2019-07-11`, `s3://DOC-EXAMPLE-BUCKET/mytable/dt=2019-07-12`, etc. se puede registrar en Lake Formation y consultar mediante Athena. Por otra parte, una tabla con la ubicación `s3://DOC-EXAMPLE-BUCKET/mytable` y las particiones ubicadas en `s3://DOC-EXAMPLE-BUCKET/dt=2019-07-11`, `s3://DOC-EXAMPLE-BUCKET/dt=2019-07-12`, etc. no se puede registrar en Lake Formation. Debido a que estas particiones no son subdirectorios de `s3://DOC-EXAMPLE-BUCKET/mytable`, tampoco se pueden leer desde Athena.

Las consultas Create Table As Select (CTAS) requieren permisos de escritura de Amazon S3

Las consultas Create Table As Statements (CTAS) requieren acceso de escritura a la ubicación de las tablas de Amazon S3. Para ejecutar consultas CTAS sobre datos registrados en Lake Formation, los usuarios de Athena deben tener permisos de IAM para escribir en la tabla ubicaciones de Amazon S3, además de los permisos de Lake Formation correspondientes para leer las ubicaciones de datos. Para obtener más información, consulte [Creación de una tabla a partir de los resultados de una consulta \(CTAS\)](#).

El permiso DESCRIBE es necesario en la base de datos predeterminada

El permiso [DESCRIBE](#) de Lake Formation es necesario en la base de datos default. El siguiente ejemplo de comando AWS CLI otorga el permiso DESCRIBE en la base de datos default al usuario dataLake\_user1 en la cuenta 111122223333 de AWS.

```
aws lakeformation grant-permissions --principal
  DataLakePrincipalIdentifier=arn:aws:iam::111122223333:user/dataLake_user1 --
permissions "DESCRIBE" --resource '{ "Database": {"Name":"default"}}
```

Para obtener más información, consulte [Referencia de permisos de Lake Formation](#) en la Guía para desarrolladores de AWS Lake Formation.

## Administración de permisos de usuario de Lake Formation y Athena

Lake Formation ofrece credenciales para consultar almacenes de datos de Amazon S3 registrados en Lake Formation. Si anteriormente utilizó políticas de IAM para conceder o denegar permisos para leer ubicaciones de datos en Amazon S3, puede utilizar permisos de Lake Formation en su lugar. Sin embargo, otros permisos de IAM siguen siendo necesarios.

Siempre que utilice políticas de IAM, asegúrese de seguir las prácticas recomendadas de IAM. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM.

En las siguientes secciones, se resumen los permisos necesarios para utilizar Athena para consultar los datos registrados en Lake Formation. Para obtener más información, consulte [Seguridad en AWS Lake Formation](#) en la Guía para desarrolladores de AWS Lake Formation.

### Resumen de permisos

- [Permisos basados en identidad para Lake Formation y Athena](#)
- [Permisos de Amazon S3 para las ubicaciones de los resultados de las consultas de Athena](#)
- [Suscripciones a grupos de trabajo de Athena para el historial de consultas](#)
- [Permisos de Lake Formation para datos](#)
- [Permisos de IAM para escribir en ubicaciones de Amazon S3](#)
- [Permisos para datos cifrados, metadatos y resultados de consultas de Athena](#)
- [Permisos basados en recursos para buckets de Amazon S3 en cuentas externas \(opcional\)](#)

## Permisos basados en identidad para Lake Formation y Athena

Cualquier persona que utilice Athena para consultar datos registrados en Lake Formation debe tener una política de permisos de IAM que permita la acción `lakeformation:GetDataAccess`. La [Política administrada de AWS: AmazonAthenaFullAccess](#) permite esta acción. Si utiliza políticas insertadas, asegúrese de actualizar las políticas de permisos para permitir esta acción.

En Lake Formation, un administrador de lago de datos tiene permisos para crear objetos de metadatos tales como bases de datos y tablas, conceder permisos de Lake Formation a otros usuarios y registrar nuevas ubicaciones de Amazon S3. Para registrar nuevas ubicaciones, se necesitan permisos para el rol vinculado al servicio para Lake Formation. Para obtener más información, consulte [Creación de un administrador de lago de datos](#) y [Permisos de roles vinculados al servicio para Lake Formation](#) en la Guía para desarrolladores de AWS Lake Formation.

Un usuario de Lake Formation puede utilizar Athena para consultar bases de datos, tablas, columnas de tablas y almacenes de datos de Amazon S3 subyacentes en función de los permisos de Lake Formation que le hayan concedido los administradores del lago de datos. Los usuarios no pueden crear bases de datos ni tablas, ni registrar nuevas ubicaciones de Amazon S3 con Lake Formation. Para obtener más información, consulte [Creación de un usuario de lago de datos](#) en la Guía para desarrolladores de AWS Lake Formation.

En Athena, las políticas de permisos basadas en identidades, incluidas las de los grupos de trabajo de Athena, siguen controlando el acceso a las acciones de Athena para los usuarios de cuentas de Amazon Web Services. Además, el acceso federado podría proporcionarse a través de la autenticación basada en SAML disponible con controladores de Athena. Para obtener más información, consulte [Uso de grupos de trabajo para controlar el acceso a las consultas y los costos](#), [Políticas de IAM para acceder a los grupos de trabajo](#) y [Habilitación de acceso federado a la API de Athena](#).

Para obtener más información, consulte [Concesión de permisos de Lake Formation](#) en la Guía para desarrolladores de AWS Lake Formation.

## Permisos de Amazon S3 para las ubicaciones de los resultados de las consultas de Athena

Las ubicaciones de los resultados de consulta en Amazon S3 para Athena no se pueden registrar en Lake Formation. Los permisos de Lake Formation no limitan el acceso a estas ubicaciones. A menos que limite el acceso, los usuarios de Athena pueden acceder a los archivos de resultados de consultas y metadatos cuando no tienen permisos de Lake Formation para los datos. Para evitar esto, le recomendamos que utilice grupos de trabajo para especificar la ubicación de los resultados de consulta y alinear la pertenencia a grupos de trabajo con permisos de Lake Formation.



A continuación, puede utilizar políticas de permisos de IAM para limitar el acceso a ubicaciones de resultados de consulta. Para obtener más información sobre los resultados de consultas, consulte [Trabajo con resultados de las consultas, consultas recientes y archivos de salida](#).

### Suscripciones a grupos de trabajo de Athena para el historial de consultas

El historial de consultas de Athena expone una lista de consultas guardadas y las cadenas de consultas completas. A menos que utilice grupos de trabajo para separar el acceso a los historiales de consultas, los usuarios de Athena que no tengan autorización para consultar datos en Lake Formation pueden ver cadenas de consulta ejecutadas en esos datos, incluidos nombres de columna, criterios de selección, etc. Le recomendamos que utilice grupos de trabajo para separar los historiales de consultas y alinear la pertenencia al grupo de trabajo de Athena con permisos de Lake Formation para limitar el acceso. Para obtener más información, consulte [Uso de grupos de trabajo para controlar el acceso a las consultas y los costos](#).

### Permisos de Lake Formation para datos

Además del permiso de referencia para utilizar Lake Formation, los usuarios de Athena deben tener permisos de Lake Formation para obtener acceso a los recursos que consultan. Un administrador de Lake Formation concede y administra estos permisos. Para obtener más información, consulte [Seguridad y control de acceso a metadatos y datos](#) en la Guía para desarrolladores de AWS Lake Formation.

### Permisos de IAM para escribir en ubicaciones de Amazon S3

Los permisos de Lake Formation para Amazon S3 no incluyen la capacidad de escribir en Amazon S3. Las consultas Create Table As Statements (CTAS) requieren acceso de escritura a la ubicación de las tablas de Amazon S3. Para ejecutar consultas CTAS sobre datos registrados en Lake Formation, los usuarios de Athena deben tener permisos de IAM para escribir en la tabla ubicaciones de Amazon S3, además de los permisos de Lake Formation correspondientes para leer las ubicaciones de datos. Para obtener más información, consulte [Creación de una tabla a partir de los resultados de una consulta \(CTAS\)](#).

### Permisos para datos cifrados, metadatos y resultados de consultas de Athena

Los datos de origen subyacentes en Amazon S3 y los metadatos en el catálogo de datos que están registrados en Lake Formation se pueden cifrar. No hay ningún cambio en la forma en que Athena gestiona el cifrado de los resultados de las consultas cuando se utiliza Athena para consultar los datos registrados en Lake Formation. Para obtener más información, consulte [Cifrado de los resultados de las consultas de Athena en Amazon S3](#).

- Cifrado de datos de origen: se admite el cifrado de datos de origen de ubicaciones de datos de Amazon S3. Los usuarios de Athena que consultan ubicaciones cifradas de Amazon S3 registradas en Lake Formation necesitan permisos para cifrar y descifrar datos. Para obtener más información acerca de los requisitos, consulte [Opciones de cifrado de Simple Storage Service \(Amazon S3\) compatibles](#) y [Permisos para datos cifrados en Amazon S3](#).
- Cifrado de metadatos: se admite el cifrado de metadatos en el catálogo de datos. Para las entidades principales que utilizan Athena, las políticas basadas en identidad deben permitir las acciones "kms:GenerateDataKey", "kms:Decrypt" y "kms:Encrypt" para la clave utilizada para cifrar los metadatos. Para obtener más información, consulte [Cifrado de su catálogo de datos](#) en la Guía para desarrolladores de AWS Glue y en el [Acceso desde Athena a metadatos cifrados en AWS Glue Data Catalog](#).

Permisos basados en recursos para buckets de Amazon S3 en cuentas externas (opcional)

Para consultar una ubicación de datos de Amazon S3 en una cuenta diferente, debe permitir el acceso a la ubicación una política de IAM basada en recursos (política de bucket). Para obtener más información, consulte [Acceso entre cuentas en Athena a los buckets de Simple Storage Service \(Amazon S3\)](#).

Para obtener información sobre cómo acceder a un catálogo de datos en otra cuenta, consulte [Acceso entre cuentas al catálogo de datos de Athena](#).

## Aplicación de permisos de Lake Formation a bases de datos y tablas existentes

Si es la primera vez que utiliza Athena y usa Lake Formation para configurar el acceso a los datos de consulta, no es necesario configurar políticas de IAM para que los usuarios puedan leer datos de Amazon S3 y crear metadatos. Puede utilizar Lake Formation para administrar permisos.

El registro de datos con Lake Formation y la actualización de políticas de permisos de IAM no es un requisito. Si los datos no están registrados en Lake Formation, los usuarios de Athena que tienen los permisos adecuados en Amazon S3 (y en AWS Glue, si procede) pueden seguir consultando datos no registrados en Lake Formation.

Si ya tiene usuarios de Athena que consultan datos no registrados en Lake Formation, puede actualizar los permisos de IAM para Amazon S3 (y para AWS Glue Data Catalog, si procede) con el fin de utilizar los permisos de Lake Formation para administrar el acceso de los usuarios de forma centralizada. Para obtener permiso para leer ubicaciones de datos de Amazon S3, puede actualizar políticas basadas en recursos y políticas basadas en identidad para modificar permisos de Amazon

S3. Para obtener acceso a los metadatos, si ha configurado políticas de nivel de recursos para un control de acceso detallado con AWS Glue, puede utilizar permisos de Lake Formation para administrar el acceso en su lugar.

Para obtener más información, consulte [Acceso detallado a las bases de datos y las tablas en AWS Glue Data Catalog](#) y [Actualización de permisos de datos de AWS Glue para el modelo de AWS Lake Formation](#) en la Guía para desarrolladores de AWS Lake Formation.

## Uso de Lake Formation y los controladores JDBC y ODBC de Athena para el acceso federado a Athena

Los controladores JDBC y ODBC de Athena admiten la federación basada en SAML 2.0 con Athena mediante proveedores de identidad Okta y Microsoft Active Directory Federation Services (AD FS). Al integrar Amazon Athena con AWS Lake Formation, habilita la autenticación basada en SAML en Athena con credenciales corporativas. Con Lake Formation y AWS Identity and Access Management (IAM), puede mantener un control de acceso detallado a nivel de columna sobre los datos disponibles para el usuario SAML. Con los controladores JDBC y ODBC de Athena, el acceso federado está disponible para acceso mediante herramientas o programación.

Para utilizar Athena a fin de acceder a una fuente de datos controlada por Lake Formation, debe habilitar la federación basada en SAML 2.0 mediante la configuración de su proveedor de identidad (IdP) y AWS Identity and Access Management (IAM). Para conocer los pasos en detalle, consulte [Tutorial: Configuración de acceso federado a Athena para usuarios de Okta mediante Lake Formation y JDBC](#).

### Requisitos previos

Para utilizar Amazon Athena and Lake Formation para acceder federado, debe cumplir los siguientes requisitos:

- Administrar las identidades corporativas con un proveedor de identidad existente basado en SAML, como Okta o Active Directory Federation Services (AD FS).
- Utilizar el AWS Glue Data Catalog como almacén de metadatos.
- Definir y administrar los permisos en Lake Formation para acceder a las bases de datos, tablas y columnas en AWS Glue Data Catalog. Para obtener más información, consulte la [Guía para desarrolladores de AWS Lake Formation](#).
- Utilice la versión 2.0.14 o posterior del [controlador JDBC de Athena](#) o la versión 1.1.3 o posterior del [controlador ODBC de Athena](#).

## Consideraciones y limitaciones

Cuando utilice el controlador JDBC u ODBC de Athena junto con Lake Formation para configurar el acceso federado a Athena, tenga en cuenta los siguientes puntos:

- Actualmente, los controladores JDBC y ODBC de Athena admiten los proveedores de identidad de Okta, Servicios de federación de Active Directory (AD FS) y Azure AD. Si bien el controlador JDBC de Athena tiene una clase SAML genérica que se puede ampliar para utilizar otros proveedores de identidad, la compatibilidad con extensiones personalizadas que permiten que otros proveedores de identidad (IdP) se usen con Athena puede ser limitada.
- El acceso de federación mediante los controladores JDBC y ODBC no es compatible con la característica de propagación de identidades fiables de IAM Identity Center.
- Actualmente, no se puede utilizar la consola de Athena para configurar la compatibilidad con el uso de IdP y SAML con Athena. Para configurar esta compatibilidad, utilice el proveedor de identidad de terceros, las consolas de administración de Lake Formation e IAM y el cliente de controlador JDBC u ODBC.
- Debe comprender la [especificación SAML 2.0](#) y cómo funciona con su proveedor de identidad antes de configurar su proveedor de identidad y SAML para su uso con Lake Formation y Athena.
- Los proveedores SAML y los controladores JDBC y ODBC de Athena son proporcionados por terceros, por lo que la compatibilidad a través de AWS para los problemas relacionados con su uso puede ser limitada.

## Temas

- [Tutorial: Configuración de acceso federado a Athena para usuarios de Okta mediante Lake Formation y JDBC](#)

Tutorial: Configuración de acceso federado a Athena para usuarios de Okta mediante Lake Formation y JDBC

En este tutorial, se muestra cómo configurar Okta, AWS Lake Formation, permisos de AWS Identity and Access Management y el controlador JDBC de Athena para habilitar el uso federado basado en SAML de Athena. Lake Formation proporciona un control de acceso detallado sobre los datos disponibles en Athena para el usuario basado en SAML. Para establecer esta configuración, el tutorial utiliza la consola de desarrollador de Okta, las consolas de AWS IAM y Lake Formation, y la herramienta SQL Workbench/J.

## Requisitos previos

Este tutorial presupone que ha realizado las siguientes acciones:

- Crear una cuenta de Amazon Web Services. Para crear una cuenta, visite la [página de inicio de Amazon Web Services](#).
- [Configurar una ubicación de resultados de consulta](#) para Athena en Amazon S3.
- [Registrar una ubicación de bucket de datos de Amazon S3](#) con Lake Formation.
- Definir una [base de datos](#) y [tablas](#) en el [catálogo de datos de AWS Glue](#) que apunten a los datos en Amazon S3.
  - Si todavía no ha definido una tabla, [ejecute un rastreador AWS Glue](#) o [utilice Athena para definir una base de datos y una o más tablas](#) para ver los datos a los que desea obtener acceso.
  - En este tutorial, se utiliza una tabla basada en el [conjunto de datos de viajes de taxi de la ciudad de Nueva York](#) disponible en el [Registro de datos abiertos de AWS](#). El tutorial utiliza el nombre de la base de datos `tripdb` y el nombre de la tabla `nyctaxi`.

## Pasos del tutorial

- [Paso 1: Crear una cuenta de Okta](#)
- [Paso 2: Agregar usuarios y grupos a Okta](#)
- [Paso 3: Configurar una aplicación de Okta para la autenticación SAML](#)
- [Paso 4: Crear un proveedor de identidad SAML de AWS y un rol de IAM para acceso a Lake Formation](#)
- [Paso 5: Agregar el rol de IAM y el proveedor de identidad SAML a la aplicación de Okta](#)
- [Paso 6: Otorgar permisos de usuario y grupo a través de AWS Lake Formation](#)
- [Paso 7: Verificar el acceso a través del cliente JDBC de Athena](#)
- [Conclusión](#)
- [Recursos relacionados](#)

## Paso 1: Crear una cuenta de Okta

En este tutorial, se utiliza Okta como proveedor de identidad basado en SAML. Si no dispone de una cuenta de Okta, puede crear una gratuita. Es necesaria una cuenta de Okta para poder crear una aplicación de Okta para la autenticación SAML.

## Para crear una cuenta de Okta

1. Para utilizar Okta, vaya a la [página de registro para desarrolladores de Okta](#) y cree una cuenta de prueba gratuita de Okta. El servicio de edición de desarrollador es gratuito hasta los límites especificados por Okta en [developer.okta.com/pricing](https://developer.okta.com/pricing).
2. Cuando reciba el correo electrónico de activación, active la cuenta.

Se le asignará un nombre de dominio de Okta. Guarde el nombre de dominio como referencia. Más adelante, utilizará el nombre de dominio (`<okta-idp-domain>`) en la cadena JDBC que se conecta a Athena.

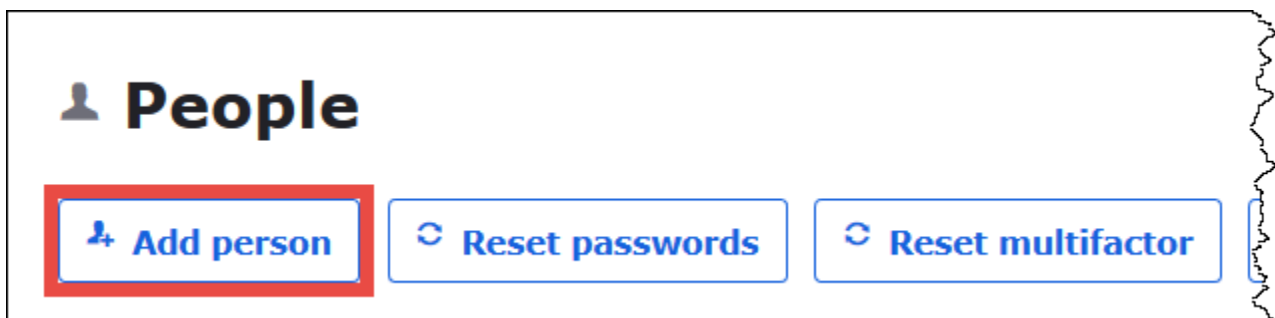
## Paso 2: Agregar usuarios y grupos a Okta

En este paso, se utiliza la consola de Okta para realizar las siguientes tareas:

- Crear dos usuarios de Okta.
- Cree dos grupos de Okta.
- Agregue un usuario de Okta a cada grupo de Okta.

### Para agregar usuarios a Okta

1. Después de activar la cuenta de Okta, inicie sesión como usuario administrativo en el dominio Okta asignado.
2. En el panel de navegación izquierdo, elija Directory (Directorio) y, a continuación, elija People (Personas).
3. Elija Add Person (Agregar persona) para agregar un usuario nuevo que accederá a Athena a través del controlador JDBC.



4. En el cuadro de diálogo Add person (Agregar persona), ingrese la información requerida.

- En First name (Nombre) y Last name (Apellido), ingrese valores. En este tutorial se utiliza *athena-okta-user*.
- Ingrese un Username (Nombre de usuario) y Primary email (Correo electrónico principal). En este tutorial se utiliza *athena-okta-user@anycompany.com*.
- En Password (Contraseña), elija Set by admin (Definida por administrador) y, a continuación, proporcione una contraseña. En este tutorial se desactiva la opción User must change password on first login (El usuario debe cambiar la contraseña en el primer inicio de sesión). Los requisitos de seguridad pueden variar.

## Add Person

User type <sup>?</sup>

User ▼

First name

athena-okta-user

Last name

athena-okta-user

Username

athena-okta-user@anycompany.com

Primary email

athena-okta-user@anycompany.com

Secondary email (optional)

Groups (optional)

Password <sup>?</sup>

Set by admin ▼

Enter password

User must change password on first login

Save

Save and Add Another

Cancel



5. Elija Save and Add Another (Guardar y agregar otro).
6. Ingrese la información de otro usuario. En este ejemplo, se agrega el usuario de analista empresarial *athena-ba-user@anycompany.com*.

## Add Person

User type <sup>?</sup>

User ▼

First name

athena-ba-user

Last name

athena-ba-user

Username

athena-ba-user@anycompany.com

Primary email

athena-ba-user@anycompany.com

Secondary email (optional)

Groups (optional)

Password <sup>?</sup>

Set by admin ▼

Enter password

User must change password on first login

Save

Save and Add Another

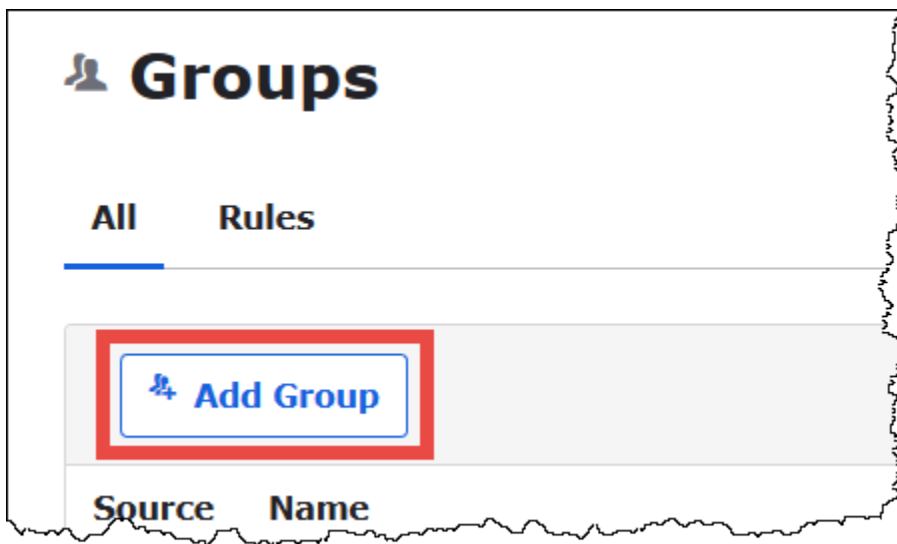
Cancel

## 7. Seleccione Guardar.

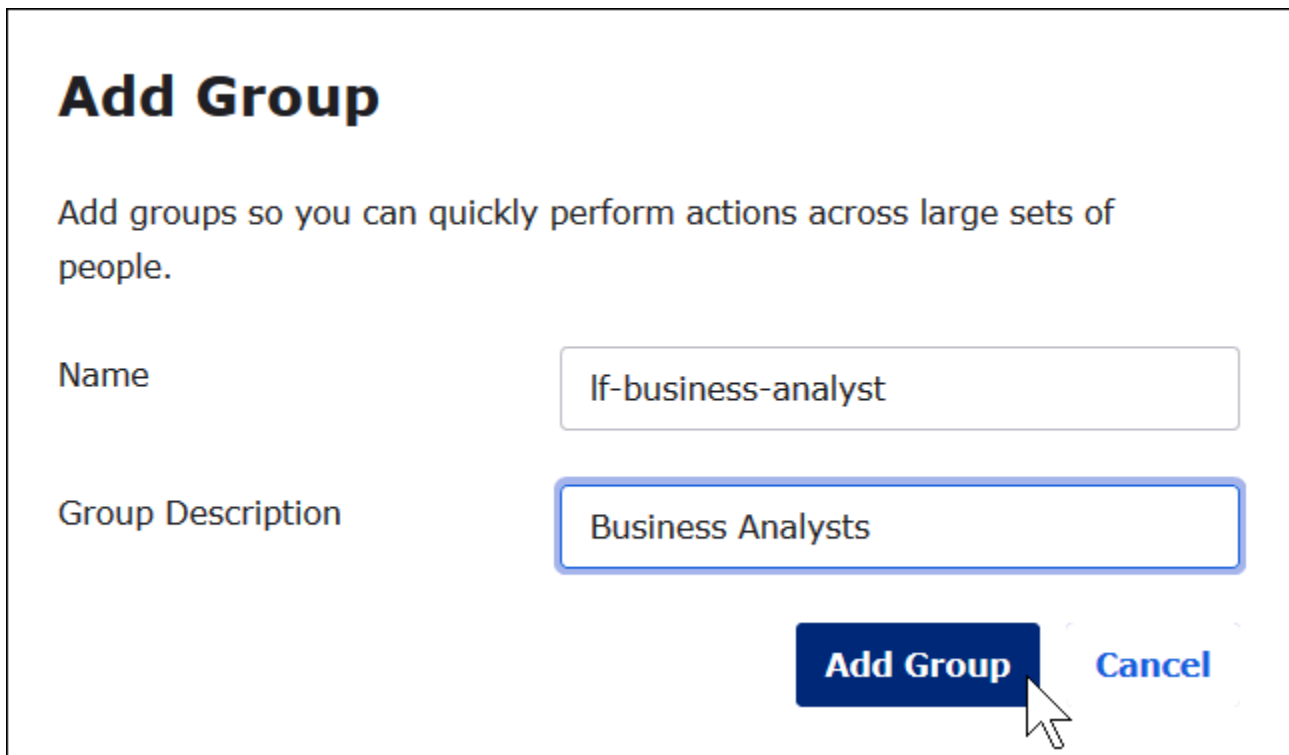
En el siguiente procedimiento, proporcionará acceso a dos grupos de Okta a través del controlador JDBC de Athena al agregar un grupo de “analista empresarial” y un grupo “Desarrollador”.

Para agregar grupos de Okta

1. En el panel de navegación de Okta, elija Directory (Directorio) y, a continuación, elija Groups (Grupos).
2. En la página Groups (Grupos), elija Add Group (Agregar grupo).



3. En el cuadro de diálogo Add Group (Agregar grupo), ingrese la información requerida.
  - En Name (Nombre), ingrese *lf-business-analyst*.
  - En Group Description (Descripción de grupo), ingrese *Business Analysts* (Analistas de negocio).



**Add Group**

Add groups so you can quickly perform actions across large sets of people.

Name

Group Description

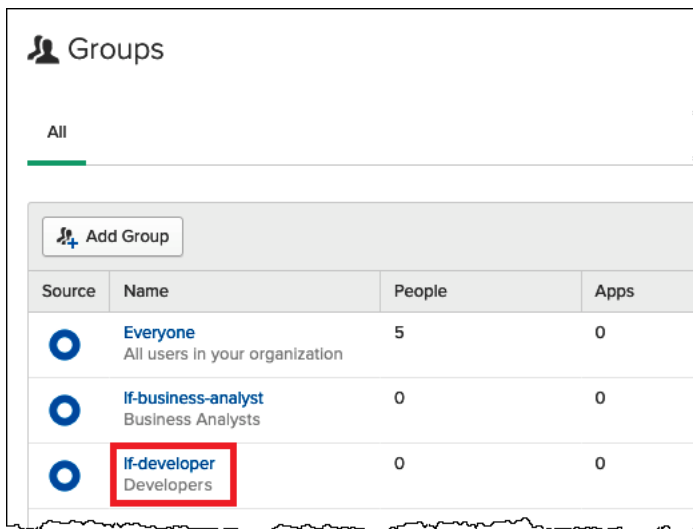
**Add Group** Cancel

4. Seleccione Añadir grupo.
5. En la página Groups (Grupos), vuelva a elegir Add Group (Agregar grupo). Esta vez ingresará información para el grupo Desarrollador.
6. Ingrese la información requerida.
  - En Name (Nombre), ingrese *lf-developer*.
  - En Group Description (Descripción de grupo), ingrese *Developers* (Desarrolladores).
7. Seleccione Añadir grupo.

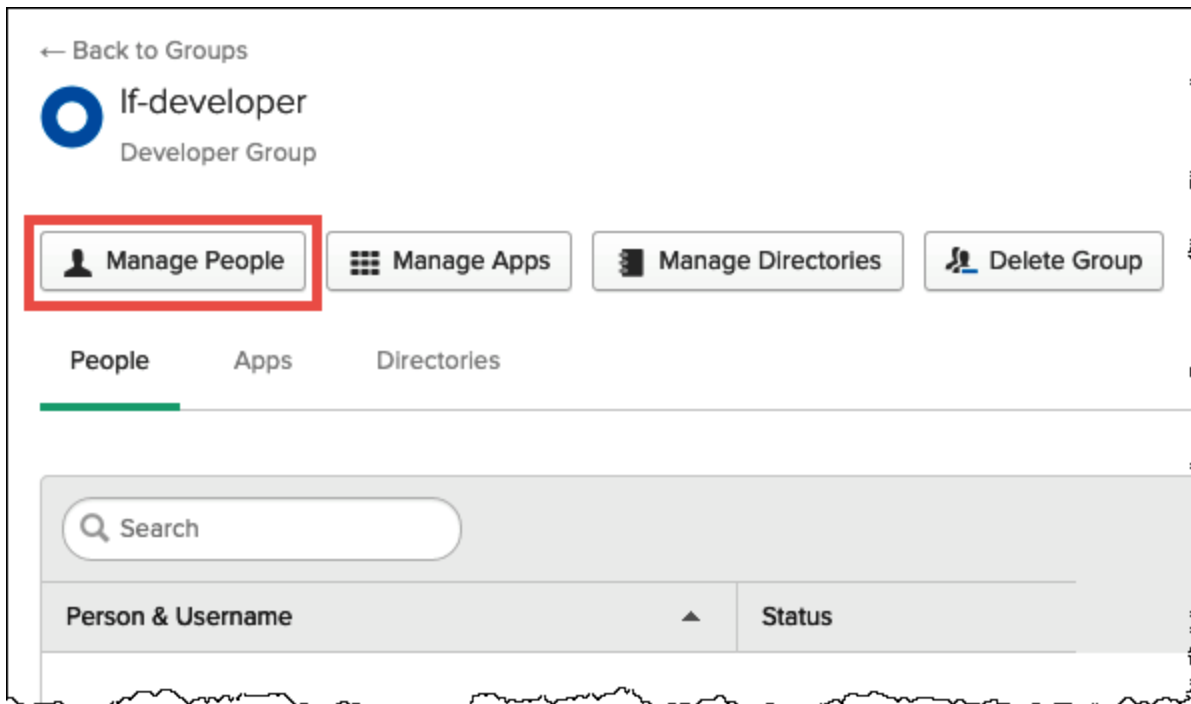
Ahora que tiene dos usuarios y dos grupos, puede agregar un usuario a cada grupo.

Para agregar usuarios a los grupos

1. En la página Groups (Grupos), elija el grupo lf-developer que acaba de crear. Agregará a este grupo uno de los usuarios de Okta que creó como desarrollador.




2. Elija Manage People (Administrar personas).




3. En la lista Not Members (No miembros), elija athena-okta-user.


← Back to Group

 **If-developer**  
Developers

---



### Add or remove people from the If-developer group

Search by person 

 **Not Members** Showing 1 - 4 of 4


Person & Username ▾

athena-ba-user athena-ba-user  
athena-ba-user@anycompany.com

**athena-okta-user athena-okta-user**  

athena-okta-user@anycompany.com

First Previous **1** Next Last

 **Members**

Person & Username ▲

First Previous Next Last

La entrada para el usuario pasa de la lista de No miembros de la izquierda a la lista de Miembros de la derecha.

← Back to Group

**If-developer**  
Developers

Add or remove people from the If-developer group

Cancel Save

Q ▼

+ Add All (3) - Remove All (1)

👤 **Not Members** Showing 1 - 3 of 3

Person & Username ▼

athena-ba-user athena-ba-user  
athena-ba-user@anycompany.com

First Previous **1** Next Last

👤 **Members** Showing 1 - 1 of 1

Person & Username ▲



athena-okta-user athena-okta-user  
athena-okta-user@anycompany.com

First Previous **1** Next Last

Cancel Save

4. Seleccione Guardar.
5. Elija Back to Group (Volver a grupo) o Directory (Directorio) y, a continuación, elija Groups (Grupos).
6. Elija el grupo If-business-analyst.
7. Elija Manage People (Administrar personas).
8. Agregue el usuario athena-ba-user a la lista de Miembros del grupo If-business-analyst y, a continuación, elija Save (Guardar).
9. Elija Back to Group (Volver a grupo) o elija Directory (Directorio), Groups (Grupos).

La página Groups (Grupos) muestra ahora que cada grupo tiene un usuario de Okta.

Source	Name	People	Apps
	If-business-analyst Business Analyst	1	0
	If-developer Developer Group	1	0

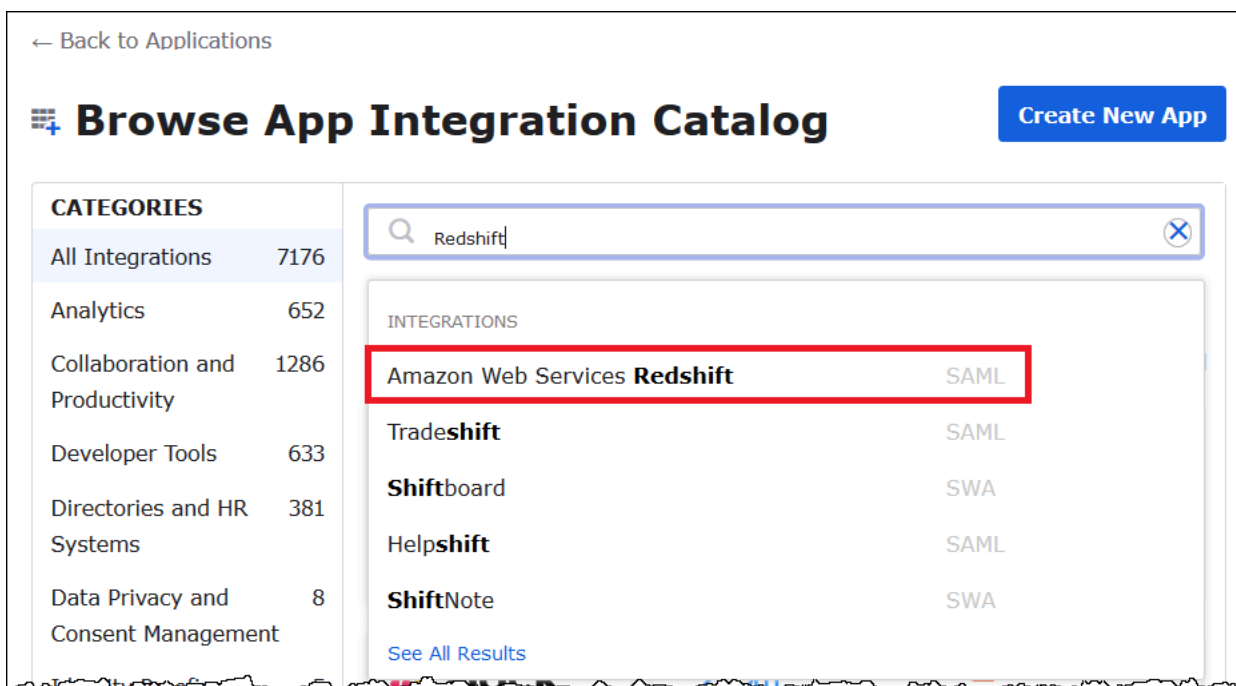
### Paso 3: Configurar una aplicación de Okta para la autenticación SAML

En este paso, se utiliza la consola de desarrollador de Okta para realizar las siguientes tareas:

- Agregar una aplicación SAML para su uso con AWS.
- Asignar la aplicación al usuario de Okta.
- Asignar la aplicación a un grupo de Okta.
- Descargue los metadatos resultantes del proveedor de identidad para su uso posterior con AWS.

Para agregar una aplicación para autenticación SAML

1. En el panel de navegación de Okta, elija Applications (Aplicaciones), Applications (Aplicaciones) para que pueda configurar una aplicación de Okta para la autenticación SAML en Athena.
2. Haga clic en Browse App Catalog (Explorar el catálogo de aplicaciones).
3. En el cuadro de búsqueda, escriba **Redshift**.
4. Elija Amazon Web Services Redshift. La aplicación de Okta de este tutorial utiliza la integración SAML existente para Amazon Redshift.




5. En la página Amazon Web Services Redshift, elija Add (Agregar) para crear una aplicación basada en SAML para Amazon Redshift.



← Back to Add Application

# Amazon Web Services Redshift

**Overview** Capabilities



**Add**

CATEGORIES

[Security](#)

## Overview

Okta's integration with Amazon Web Services (AWS) Redshift allows end users to authenticate to AWS Redshift accounts using single sign-on with SAML. Once SAML SSO is configured you can use SQL client tools such as SQL Workbench/J to connect to redshift directly from the application.

6. En Application label (Etiqueta de aplicación), ingrese Athena-LakeFormation-Okta y, a continuación, elija Done (Listo).

# Add Amazon Web Services Redshift

## **1** General Settings

### General Settings - Required

Application label

Athena-LakeFormation-Okta

This label displays under the app on your home page

Application Visibility

Do not display application icon to users

Do not display application icon in the Okta Mobile App

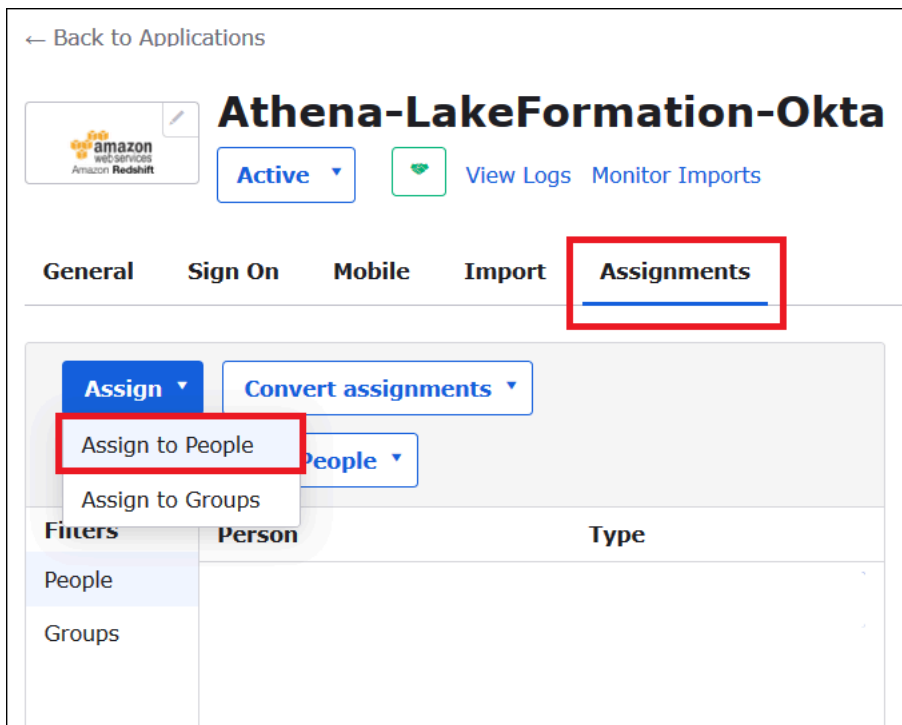
Cancel

Done

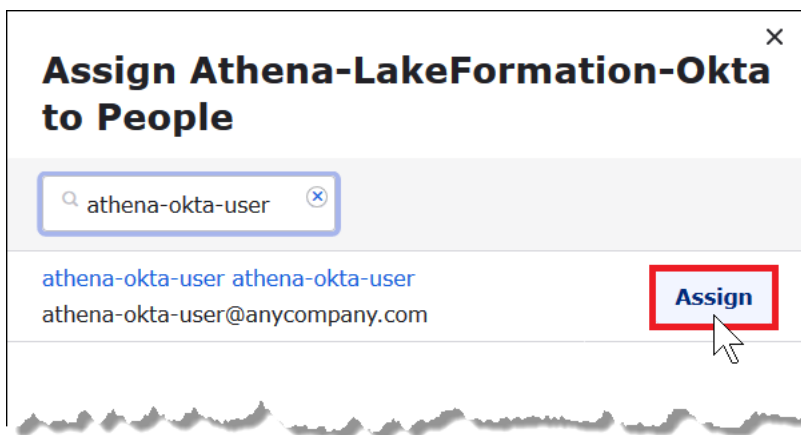
Ahora que ha creado una aplicación de Okta, puede asignarla a los usuarios y grupos que creó.

Para asignar la aplicación a usuarios y grupos

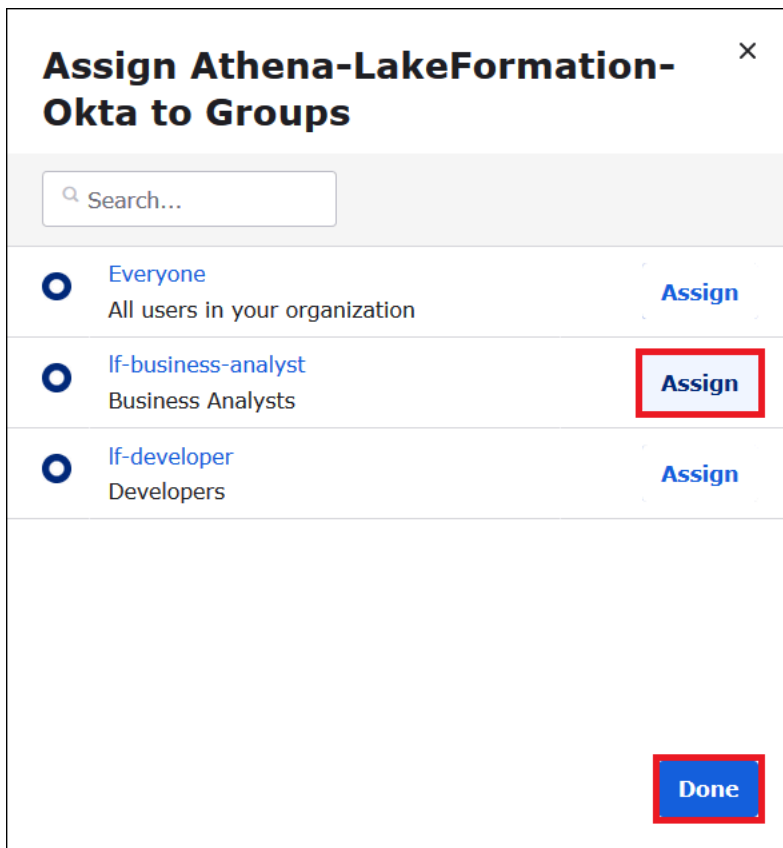
1. En la página Applications (Aplicaciones), elija la aplicación Athena-LakeFormation-Okta.
2. En la pestaña Assignments (Asignaciones), elija Assign (Asignar), Assign to People (Asignar a personas).



3. En el cuadro de diálogo Assign Athena-LakeFormation-Okta to People (Asignar Athena-LakeFormation-Okta a personas), busque el usuario athena-okta-user que creó anteriormente.
4. Elija Assign (Asignar) para asignar al usuario a la aplicación.



5. Elija Save and Go Back (Guardar y volver).
6. Seleccione Listo.
7. En la pestaña Assignments (Asignaciones), para la aplicación Athena-LakeFormation-Okta, elija Assign (Asignar), Assign to Groups (Asignar a grupos).
8. Para If-business-analyst, elija Assign (Asignar) para asignar la aplicación Athena-LakeFormation-Okta al grupo If-business-analyst y, a continuación, elija Done (Listo).



El grupo aparece en la lista de grupos de la aplicación.

The screenshot shows the AWS IAM console interface for an application named 'Athena-LakeFormation-Okta'. At the top, there is a navigation link '← Back to Applications'. Below it, the application logo (Amazon Redshift) and name are displayed. The application status is 'Active', and there are links for 'View Logs' and 'Monitor Imports'. A navigation bar includes tabs for 'General', 'Sign On', 'Mobile', 'Import', and 'Assignments', with 'Assignments' being the active tab. Below the navigation bar, there are buttons for 'Assign' and 'Convert assignments', a search box, and a 'Groups' dropdown. A table lists the assigned groups:

Filters	Priority	Assignment
People		
Groups	1	If-business-analyst Business Analysts

Ahora ya puede descargar los metadatos de la aplicación del proveedor de identidad para utilizarlos con AWS.

Para descargar los metadatos de la aplicación

1. Elija la pestaña Sign On (Inicio de sesión) de la aplicación de Okta y, a continuación, haga clic con el botón derecho en Identity Provider metadata (Metadatos del proveedor de identidad).

The screenshot shows the 'Sign On' configuration page for an application named 'Athena-LakeFormation-Okta'. The page is part of the Amazon IAM console. At the top, there's a status 'Active' and links for 'View Logs' and 'Monitor Imports'. The 'Sign On' tab is highlighted with a red box. Below the tabs, there's a 'Settings' section with an 'Edit' link. Under 'Sign on methods', there's a description of sign-on methods and a note about application username determination. The 'SAML 2.0' method is selected. A yellow warning banner indicates that SAML 2.0 is not configured until setup instructions are completed. A red box highlights the 'Identity Provider metadata' link in the banner.

**Athena-LakeFormation-Okta**

Active View Logs Monitor Imports

General **Sign On** Mobile Import Assignments

**Settings** Edit

**Sign on methods**

The sign-on method determines how a user signs into and manages their credentials for an application. Some sign-on methods require additional configuration in the 3<sup>rd</sup> party application.

Application username is determined by the user profile mapping.  
[Configure profile mapping](#)

SAML 2.0

Default Relay State

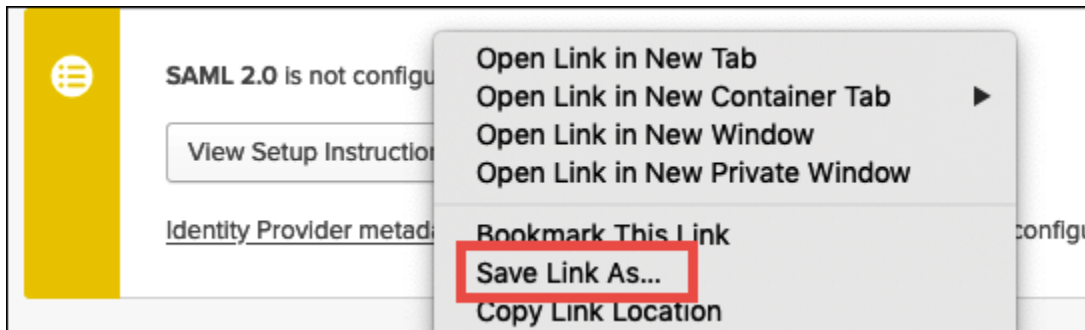
Attributes (Optional) [Learn More](#)

**SAML 2.0** is not configured until you complete the setup instructions.

[View Setup Instructions](#)

**Identity Provider metadata** is available if this application supports dynamic configuration.

2. Elija Save Link As (Guardar enlace como) para guardar los metadatos del proveedor de identidad, que están en formato XML, en un archivo. Dele un nombre que reconozca (por ejemplo, `Athena-LakeFormation-idp-metadata.xml`).



#### Paso 4: Crear un proveedor de identidad SAML de AWS y un rol de IAM para acceso a Lake Formation

En este paso, utiliza la consola de AWS Identity and Access Management (IAM) para realizar las siguientes tareas:

- Crear un proveedor de identidad para AWS.
- Crear un rol de IAM para acceso a Lake Formation.
- Agregar la política administrada AmazonAthenaFullAccess al rol.
- Agregar una política para Lake Formation y AWS Glue al rol.
- Agregar una política para los resultados de la consulta de Athena al rol.

#### Para crear un proveedor de identidad SAML de AWS

1. Inicie sesión en la consola de la cuenta de Amazon Web Services como administrador de cuenta de Amazon Web Services y vaya a la consola de IAM (<https://console.aws.amazon.com/iam/>).
2. En el panel de navegación, elija Identity providers (Proveedores de identidad) y, a continuación, haga clic en Add provider (Agregar proveedor).
3. En la pantalla Configurar proveedor, ingrese la siguiente información:
  - En Provider type (Tipo de proveedor), elija SAML.
  - En Provider name (Nombre de proveedor), ingrese AthenaLakeFormation0kta.
  - En Metadata document (Documento de metadatos), utilice la opción Choose file (Elegir archivo) para cargar el archivo XML de metadatos del proveedor de identidad (IdP) que descargó.
4. Elija Add Provider (Agregar proveedor).

A continuación, cree un rol de IAM para obtener acceso a AWS Lake Formation. Agregue dos políticas insertadas al rol. Una política proporciona permisos para acceder a Lake Formation y a la API de AWS Glue. La otra política proporciona acceso a Athena y a la ubicación de los resultados de consulta de Athena en Amazon S3.

Para crear un rol de IAM para obtener acceso a AWS Lake Formation

1. En el panel de navegación de la consola de IAM, elija Roles y, a continuación, elija Create role (Crear rol).
2. En la página Crear rol, siga los siguientes pasos:

**Create role** 1 2 3 4

**Select type of trusted entity**

**AWS service**  
 EC2, Lambda and others

**Another AWS account**  
 Belonging to you or 3rd party

**Web identity**  
 Cognito or any OpenID provider

**SAML 2.0 federation**  
 Your corporate directory

Allows users that are federated with SAML 2.0 to assume this role to perform actions in your account. [Learn more](#)

**Choose a SAML 2.0 provider**

If you're creating a role for API access, choose an Attribute and then type a Value to include in the role. This restricts access to users with the specified attributes.

**SAML provider** AthenaLakeFormationOkta

[Create new provider](#) [Refresh](#)

Allow programmatic access only

Allow programmatic and AWS Management Console access

**Attribute** SAML:aud

**Value\*** https://signin.aws.amazon.com/saml

**Condition** [+ Add condition \(optional\)](#)

\* Required Cancel **Next: Permissions**

- a. En Select type of trusted entity (Seleccionar tipo de entidad de confianza), elija SAML 2.0 Federation (Federación SAML 2.0).
- b. En SAML provider (Proveedor SAML), seleccione AthenaLakeFormationOkta.
- c. En SAML provider (Proveedor SAML), seleccione la opción Allow programmatic and AWS Management Console access (Permitir programación y acceso a ).



- d. Elija Siguiente: permisos.
3. En la página Attach Permissions policies (Adjuntar políticas de permisos), en Filter policies (Filtrar políticas), ingrese **Athena**.
4. Seleccione la política administrada AmazonAthenaFullAccess y, a continuación, elija Siguiente: etiquetas .

**Create role** 1 2 3 4

▼ Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy ↻

Filter policies  Showing 2 results

	Policy name	Used as
<input checked="" type="checkbox"/>	▶ AmazonAthenaFullAccess	Permissions policy (3)
<input type="checkbox"/>	▶ AWSQuicksightAthenaAccess	None

▶ Set permissions boundary

\* Required Cancel Previous Next: Tags

5. En la página Add tags (Añadir etiquetas), elija Next: Review (Siguiente: Revisión).
6. En la página Review (Revisar), en Role name (Nombre del rol), ingrese un nombre para el rol (por ejemplo, *Athena-LakeFormation-OktaRole*) y luego elija Create role (Crear rol).

## Create role

1 2 3 4



### Review

Provide the required information below and review this role before you create it.

**Role name\***  Use alphanumeric and '+=,@-\_' characters. Maximum 64 characters.

**Role description**  Maximum 1000 characters. Use alphanumeric and '+=,@-\_' characters.

**Trusted entities** The identity provider(s) `arn:aws:iam::[redacted]:saml-provider/AthenaLakeFormationOkta`

**Policies**  [AmazonAthenaFullAccess](#) 

**Permissions boundary** Permissions boundary is not set

*No tags were added.*

**\* Required** [Cancel](#) [Previous](#) Create role

A continuación, agregue políticas insertadas que permitan el acceso a Lake Formation, API de AWS Glue y resultados de consultas de Athena en Amazon S3.

Siempre que utilice políticas de IAM, asegúrese de seguir las prácticas recomendadas de IAM. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM.

Para agregar una política insertada al rol de Lake Formation y AWS Glue

1. En la lista de roles de la consola de IAM, elija el `Athena-LakeFormation-OktaRole` recién creado.
2. En la página Summary (Resumen) del rol, en la pestaña Permissions (Permisos), elija Add inline política (Agregar política insertada).
3. En la página Crear política, elija JSON.
4. Agregue una política insertada como la siguiente que proporcione acceso a Lake Formation y a las API de AWS Glue.

```
{
```

```

"Version": "2012-10-17",
"Statement": {
  "Effect": "Allow",
  "Action": [
    "lakeformation:GetDataAccess",
    "glue:GetTable",
    "glue:GetTables",
    "glue:GetDatabase",
    "glue:GetDatabases",
    "glue>CreateDatabase",
    "glue:GetUserDefinedFunction",
    "glue:GetUserDefinedFunctions"
  ],
  "Resource": "*"
}
}

```

5. Elija Revisar política.
6. En Name (Nombre), ingrese un nombre para la política (por ejemplo, **LakeFormationGlueInlinePolicy**).
7. Elija Crear política.

Para agregar una política insertada al rol para la ubicación de los resultados de consulta de Athena

1. En la página Summary (Resumen) del rol Athena-LakeFormation-OktaRole, en la pestaña Permissions (Permisos), elija Add inline policy (Agregar política insertada).
2. En la página Crear política, elija JSON.
3. Agregue una política insertada como la siguiente que permita el acceso del rol a la ubicación de los resultados de consulta de Athena. Reemplace los marcadores de posición *<athena-query-results-bucket>* en el ejemplo con el nombre del bucket de Amazon S3.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AthenaQueryResultsPermissionsForS3",
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:PutObject",

```

```
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::<athena-query-results-bucket>",
        "arn:aws:s3:::<athena-query-results-bucket>/*"
      ]
    }
  ]
}
```

4. Elija Revisar política.
5. En Name (Nombre), ingrese un nombre para la política (por ejemplo, **AthenaQueryResultsInlinePolicy**).
6. Elija Crear política.

A continuación, copie el ARN del rol de acceso a Lake Formation y el ARN del proveedor SAML creado. Estos son necesarios al configurar la aplicación SAML de Okta en la siguiente sección del tutorial.

Para copiar el ARN de rol y el ARN del proveedor de identidad SAML

1. En la consola de IAM, en la página Summary (Resumen) del rol Athena-LakeFormation-OktaRole, elija el icono Copy to clipboard (Copiar al portapapeles) que aparece junto a Role ARN (ARN de rol). El ARN tiene el siguiente formato:

```
arn:aws:iam::<account-id>:role/Athena-LakeFormation-OktaRole
```

2. Guarde el ARN completo de forma segura para su referencia posterior.
3. En el panel de navegación de la consola de IAM, elija Identity providers (Proveedores de identidad).
4. Elija el proveedor AthenaLakeFormationOkta.
5. En la página Summary (Resumen), elija el icono Copy to clipboard (Copiar al portapapeles) que aparece junto a Provider ARN (ARN del proveedor). El ARN debería tener el siguiente aspecto:

```
arn:aws:iam::<account-id>:saml-provider/AthenaLakeFormationOkta
```

6. Guarde el ARN completo de forma segura para su referencia posterior.

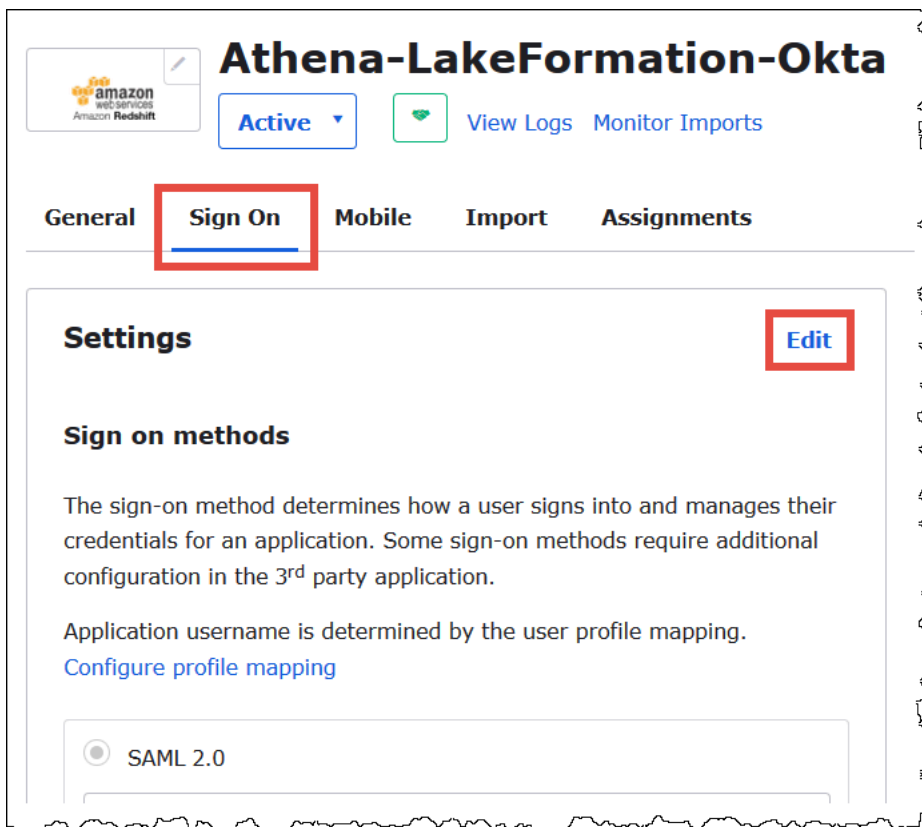
## Paso 5: Agregar el rol de IAM y el proveedor de identidad SAML a la aplicación de Okta

En este paso, vuelve a la consola de desarrollador de Okta para realizar las siguientes tareas:

- Agregar atributos URL de usuario y grupo de Lake Formation a la aplicación de Okta.
- Agregar el ARN para el proveedor de identidad y el ARN para el rol de IAM a la aplicación de Okta.
- Copiar el ID de la aplicación de Okta. El ID de la aplicación de Okta es necesario en el perfil del JDBC que se conecta a Athena.

Para agregar atributos URL de usuario y grupo de Lake Formation a la aplicación de Okta

1. Inicie sesión en la consola de desarrollador de Okta.
2. Elija la pestaña Applications (Aplicaciones), y, a continuación, elija la aplicación Athena-LakeFormation-Okta.
3. Elija la pestaña Sign On (Inicio de sesión) para la aplicación y, a continuación, elija Edit (Editar).



4. Elija Attributes (optional) (Atributos [opcional]) para expandirla.

**Athena-LakeFormation-Okta**

Active View Logs Monitor Imports

General **Sign On** Mobile Import Assignments

### Settings

[Cancel](#)

#### Sign on methods

The sign-on method determines how a user signs into and manages their credentials for an application. Some sign-on methods require additional configuration in the 3<sup>rd</sup> party application.

Application username is determined by the user profile mapping.  
[Configure profile mapping](#)

**1** SAML 2.0 is the only sign-on option currently supported for this application.

SAML 2.0

Default Relay State   
 All IDP-initiated requests will include this RelayState.

**Attributes (Optional)** [Learn More](#)

Name	Name format (optional)	Value
http:	Unspecified	user.login

[Add Another](#)

5. En Instrucciones de atributos (opcional), agregue el siguiente atributo:

- En Nombre, escriba **https://lakeformation.amazon.com/SAML/Attributes/Username**.

- En Value (Valor), ingrese **user.login**.
6. En Instrucciones de atributo de grupo (opcional), agregue el siguiente atributo:
- En Nombre, escriba **https://lakeformation.amazonaws.com/SAML/Attributes/Groups**.
  - En Name format (Formato de nombre), ingrese **Basic**.
  - En Filter (Filtro), elija Matches regex (Coincide con la expresión regular) y, a continuación, ingrese **.\*** en el cuadro de filtro.

The screenshot shows the SAML 2.0 configuration interface. It includes a section for "Attributes (Optional)" with a table for "Attribute Statements (optional)". The table has columns for "Name", "Name format", and "Value". One entry is shown with "http:" as the Name, "Unspecified" as the Name format, and "user.login" as the Value. Below this is an "Add Another" button. A second section, "Group Attribute Statements (optional)", is highlighted with a red box. It has columns for "Name", "Name format", and "Filter". One entry is shown with "https://la" as the Name, "Basic" as the Name format, and "Matches regex" as the Filter, with the value ".\*" entered in the filter field. Below this is another "Add Another" button. At the bottom of the configuration area is a "Preview SAML" button.

Name	Name format	Value
http:	Unspecified	user.login

Name	Name format	Filter
https://la	Basic	Matches regex .*

7. Desplácese hacia abajo hasta la sección Advanced Sign-On Settings (Configuración avanzada de inicio de sesión), donde agregará los ARN del proveedor de identidad y el rol de IAM a la aplicación de Okta.

Para agregar los ARN del proveedor de identidad y el rol de IAM a la aplicación de Okta

1. En Idp ARN and Role ARN (ARN de Idp y ARN de rol), ingrese el ARN de proveedor de identidad de AWS y el ARN de rol como valores separados por comas en el formato `<saml-arn>`, `<role-arn>`. La cadena combinada debe ser similar a la siguiente:

```
arn:aws:iam::<account-id>:saml-provider/  
AthenaLakeFormationOkta,arn:aws:iam::<account-id>:role/Athena-LakeFormation-  
OktaRole
```



## Advanced Sign-on Settings

These fields may be required for a Amazon Web Services Redshift proprietary sign-on option or general setting.

Idp ARN and Role ARN

Enter your AWS IDP ARN and Role ARN as comma separated values (e.g. "arn:aws:iam::1234567890:saml-provider/OKTA,arn:aws:iam::1234567890:role/SAML\_ROLE").

Session Duration

Get the user data from Lake Formation in the console



since this app is using SAML with no password.

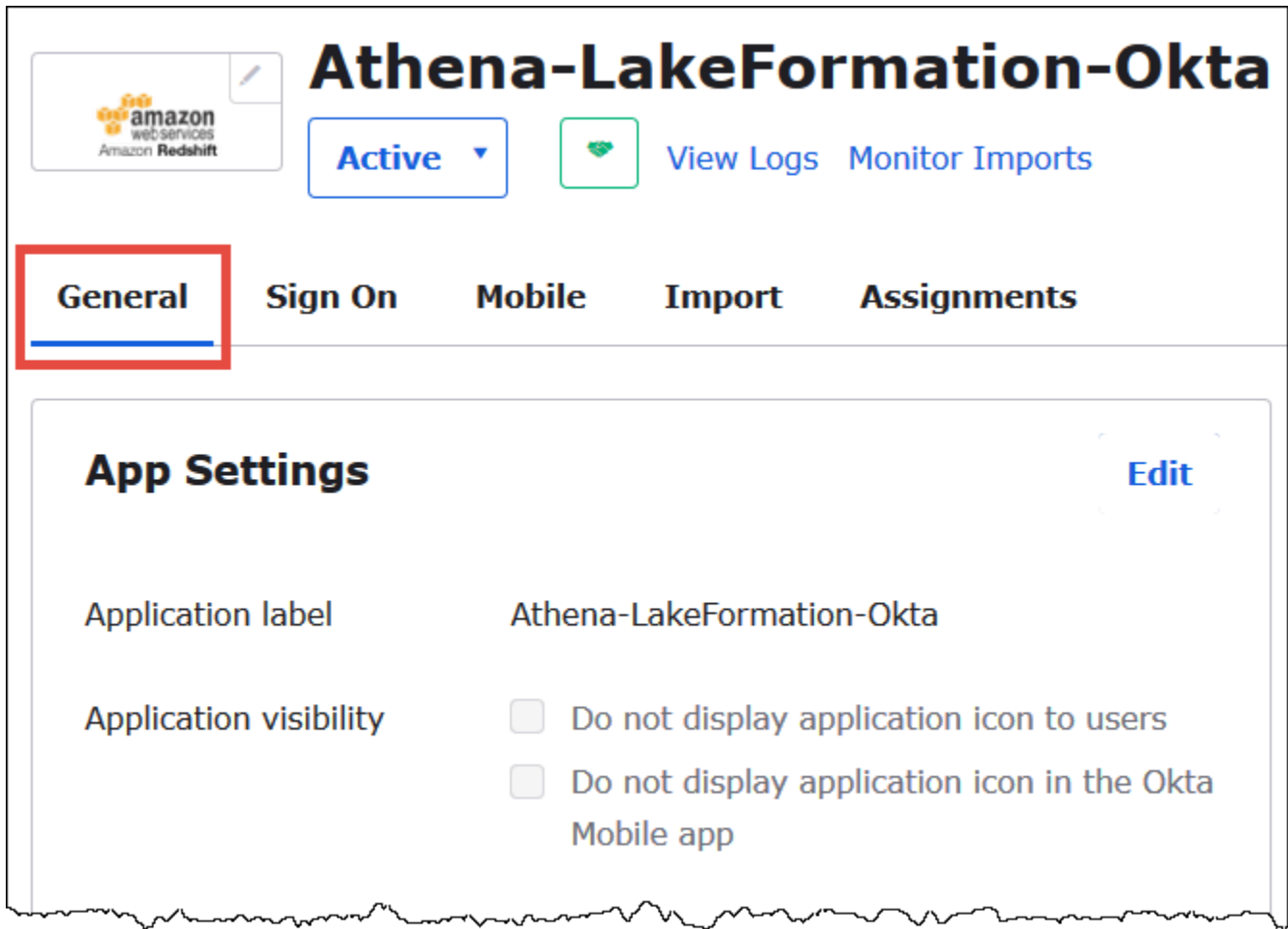
Save

2. Seleccione Guardar.

A continuación, copie el ID de la aplicación de Okta. Lo necesitará más adelante para la cadena JDBC que se conecta a Athena.

Para buscar y copiar el ID de la aplicación de Okta

1. Elija la pestaña General de la aplicación de Okta.



**Athena-LakeFormation-Okta**

amazon web services Amazon Redshift

Active

View Logs Monitor Imports

**General** Sign On Mobile Import Assignments

**App Settings** Edit

Application label Athena-LakeFormation-Okta

Application visibility

- Do not display application icon to users
- Do not display application icon in the Okta Mobile app

2. Desplácese hacia abajo hasta la sección App Embed Link (Enlace insertado de la aplicación).
3. En Embed Link (Insertar enlace), copie y guarde de forma segura la parte del ID de la aplicación de Okta de la URL. El ID de la aplicación de Okta es la parte de la URL después de `amazon_aws_redshift/` pero antes de la siguiente barra diagonal. Por ejemplo, si la dirección URL contiene `amazon_aws_redshift/aaa/bbb`, el ID de la aplicación es `aaa`.



**Note**

El enlace de inserción no se puede utilizar para registrarse directamente en la consola de Athena para ver las bases de datos. Los permisos de Lake Formation para usuarios y grupos de SAML solo se reconocen cuando se utiliza el controlador JDBC u ODBC para enviar consultas a Athena. Para ver las bases de datos, puede utilizar la herramienta SQL Workbench/J, que utiliza el controlador JDBC para conectarse a Athena. La herramienta SQL Workbench/J se aborda en [Paso 7: Verificar el acceso a través del cliente JDBC de Athena](#).

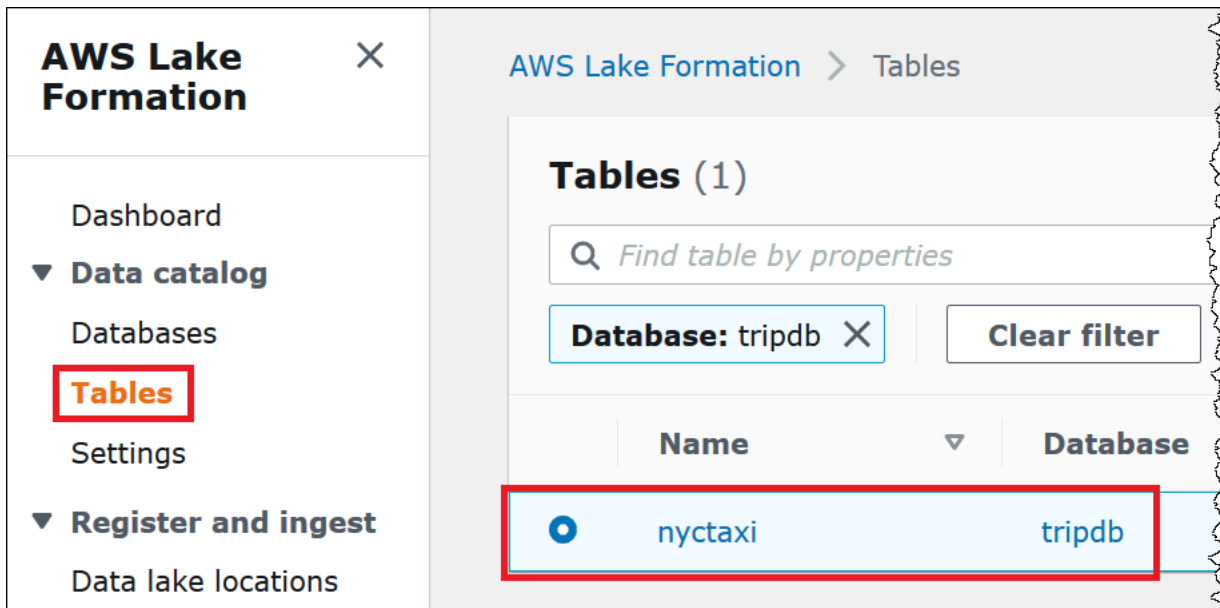
## Paso 6: Otorgar permisos de usuario y grupo a través de AWS Lake Formation

En este paso, se utiliza la consola de Lake Formation para conceder permisos en una tabla al usuario y grupo SAML. Puede realizar las siguientes tareas:

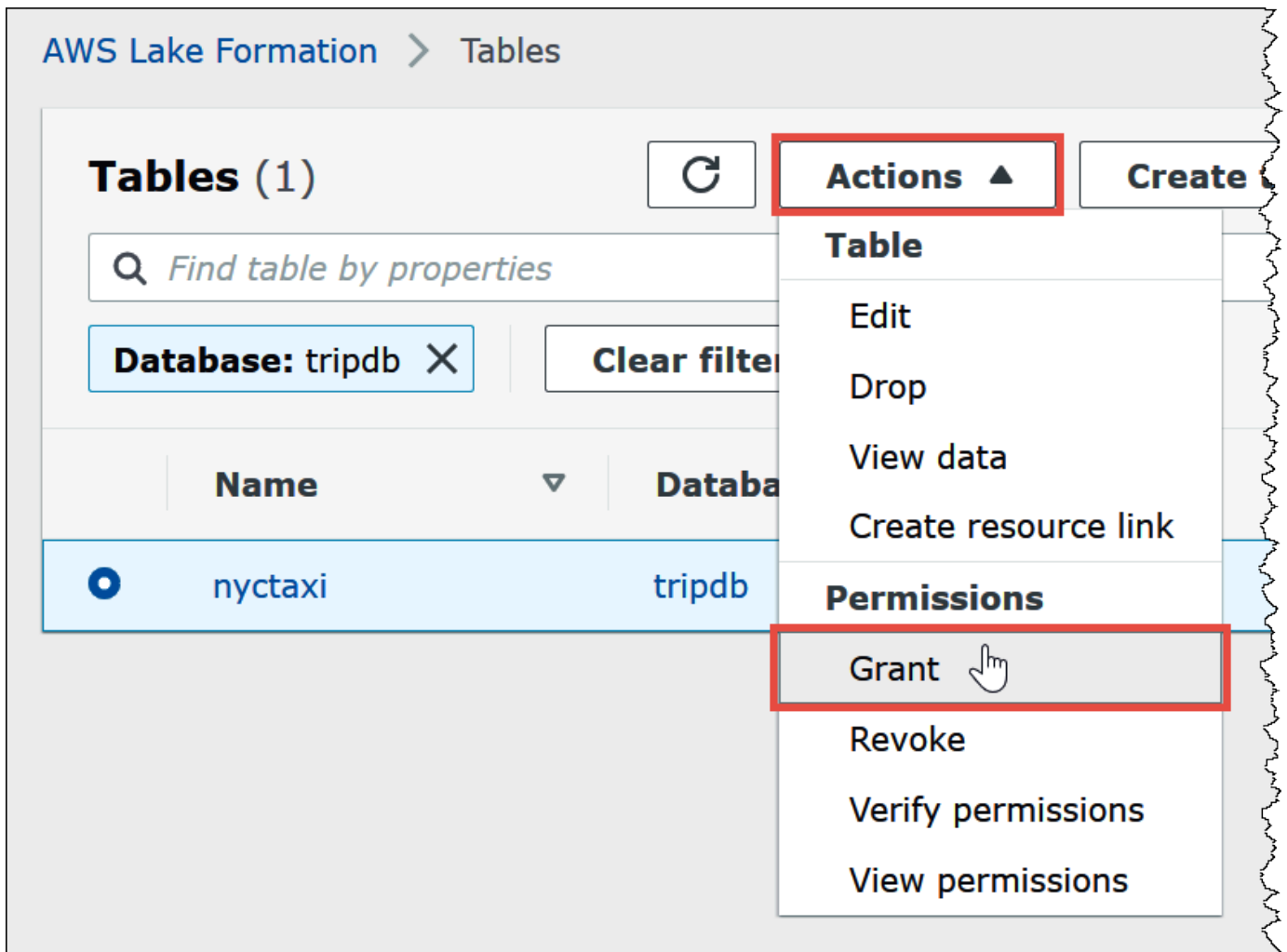
- Especificar el ARN del usuario SAML de Okta y los permisos de usuario asociados en la tabla.
- Especificar el ARN del grupo SAML de Okta y los permisos de usuario asociados en la tabla.
- Verificar los permisos que concedió.

Para conceder permisos en Lake Formation para el usuario de Okta

1. Inicie sesión como administrador del lago de datos en la AWS Management Console.
2. Abra la consola de Lake Formation en <https://console.aws.amazon.com/lakeformation/>.
3. En el panel de navegación, elija Tables (Tablas) y, a continuación, seleccione la tabla a la que desea conceder permisos. En este tutorial se utiliza la tabla `nyctaxi` de la base de datos `tripdb`.



4. En Actions (Acciones), elija Grant (Conceder).



5. En el cuadro de diálogo Conceder permisos, ingrese la siguiente información:
  - a. En SAML and Amazon QuickSight users and groups (Usuarios y grupos de SAML y Amazon QuickSight), ingrese el ARN de usuario SAML de Okta con el siguiente formato:

```
arn:aws:iam::<account-id>:saml-provider/AthenaLakeFormationOkta:user/<athena-okta-user>@<anycompany.com>
```
  - b. En Columns (Columnas), en Choose filter type (Elegir tipo de filtro), elija Include columns (Incluir columnas) o Exclude columns (Excluir columnas).
  - c. Utilice el menú desplegable Choose one or more columns (Elegir una o varias columnas) debajo del filtro para especificar las columnas que desea incluir o excluir del usuario o para él.
  - d. En Table permissions (Permisos de tabla), elija Select. En este tutorial solo se concede el permiso SELECT. Los requisitos pueden variar.

**Grant permissions: nyctaxi**

Choose the access permissions to grant.

My account  
User or role from this AWS account.

External account  
AWS account or AWS organization outside of my account.

**IAM users and roles**  
Add one or more IAM users or roles.  
Choose IAM principals to add

**SAML and Amazon QuickSight users and groups**  
Enter a SAML user or group ARN or Amazon QuickSight ARN. Press Enter to add additional ARNs.  
saml-provider/AthenaLakeFormationOkta:user/athena-okta-user@anycompany.com

**Columns - optional**  
Choose filter type  
None

**Table permissions**  
Choose the specific access permissions to grant.  
 Alter  Insert  Drop  Delete  Select  Describe

Super  
This permission is the union of the individual permissions above and supersedes them. [See here](#)

**Grantable permissions**  
Choose the permissions that may be granted to others.  
 Alter  Insert  Drop  Delete  Select  Describe

Super  
This permission allows the principal to grant any of the above permissions and supersedes those grantable permissions.

Cancel **Grant**

## 6. Elija Grant (Conceder).

Ahora realice pasos similares para el grupo de Okta.

Para conceder permisos en Lake Formation para el grupo de Okta

1. En la página Tables (Tablas) de la consola de Lake Formation, asegúrese de que la tabla nyctaxi sigue seleccionada.
2. En Actions (Acciones), elija Grant (Conceder).
3. En el cuadro de diálogo Grant permissions (Conceder permisos), ingrese la siguiente información:
  - a. En SAML and Amazon QuickSight users and groups (Usuarios y grupos de SAML y Amazon QuickSight), ingrese el ARN de grupo SAML de Okta con el siguiente formato:

```
arn:aws:iam::<account-id>:saml-provider/AthenaLakeFormationOkta:group/lf-business-analyst
```

- b. En Columns (Columnas), Choose filter type (Elegir tipo de filtro), elija Include Columns (Incluir columnas).
- c. En Choose one or more columns (Elegir una o varias columnas), elija las tres primeras columnas de la tabla.
- d. En Table permissions (Permisos de tabla), elija los permisos de acceso específicos que desea conceder. En este tutorial solo se concede el permiso SELECT. Los requisitos pueden variar.

**My account**  
User or role from this AWS account.

**External account**  
AWS account or AWS organization outside of my account.

**IAM users and roles**  
Add one or more IAM users or roles.  
Choose IAM principals to add

**SAML and Amazon QuickSight users and groups**  
Enter a SAML user or group ARN or Amazon QuickSight ARN. Press Enter to add additional ARNs.  
:saml-provider/AthenaLakeFormationOkta:group/lf-business-analyst

**Columns - optional**  
Choose filter type  
Include columns

**Include columns**  
Grant permissions to access the selected columns.  
Choose one or more columns

vendorid  X  
bigint

lpep\_pickup\_datetime  X  
string

lpep\_dropoff\_datetime  X  
string

**Table permissions**  
Choose the specific access permissions to grant.

Alter  Insert  Drop  Delete  **Select**

**Super**  
This permission is the union of the individual permissions above and supersedes them. [See here](#)

**Grantable permissions**  
Choose the permissions that may be granted to others.

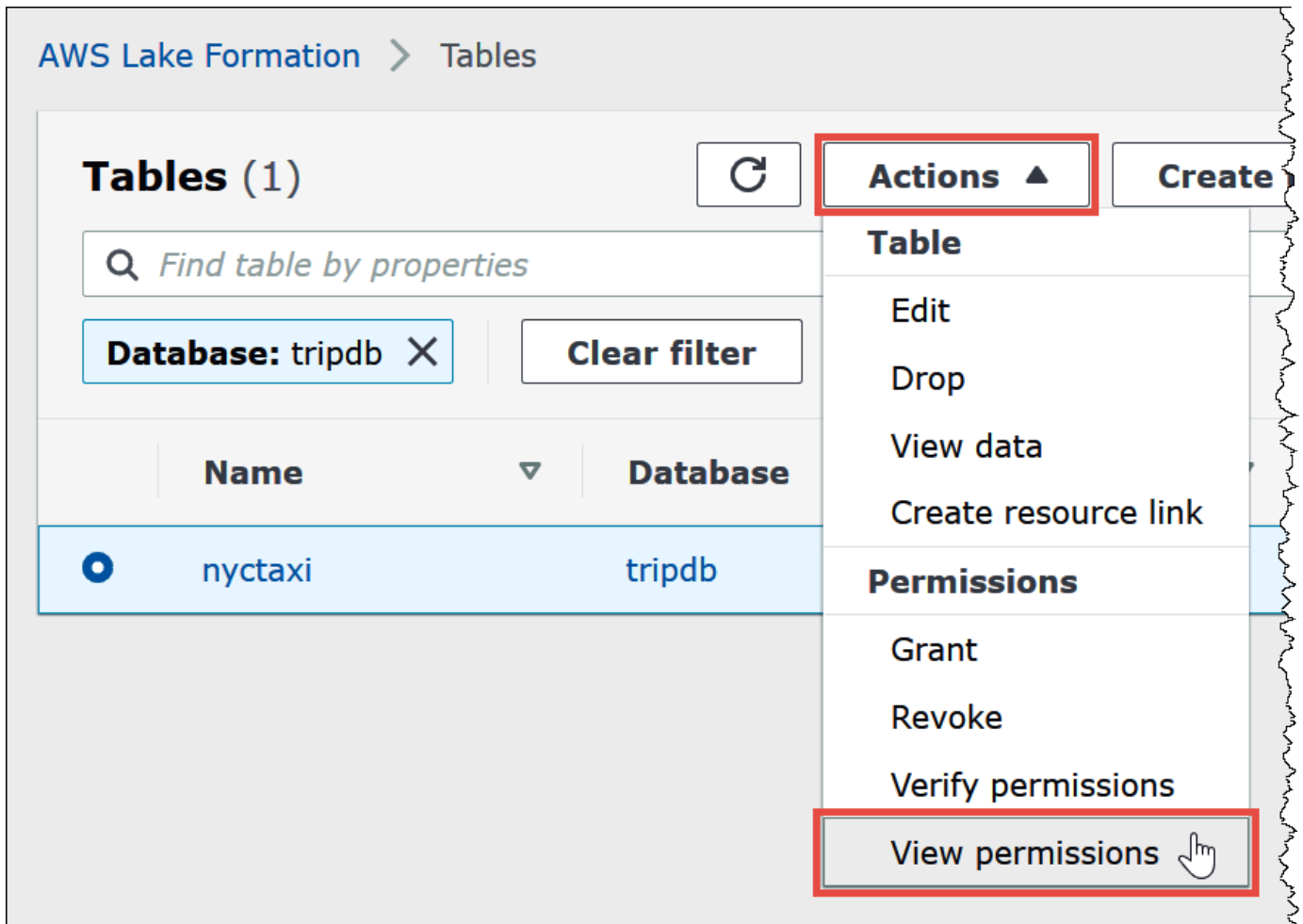
Alter  Insert  Drop  Delete  Select

**Super**  
This permission allows the principal to grant any of the above permissions and supersedes those grantable permissions.

Cancel **Grant**

#### 4. Elija Grant (Conceder).

5. Para verificar los permisos que concedió, elija Actions (Acciones), View permissions (Ver permisos).



En la página Data permissions (Permisos de datos) de la tabla `nyctaxi` se muestran los permisos para el usuario `athena-okta-user` y el grupo `lf-business-analyst`.



**Data permissions (10)**  
Choose a database or table for which to review, grant or revoke user permissions.

Find by properties

Database: tripdb X Table: nyctaxi X Clear filter

	Principal	Principal type	Resource type	Resource	Permissions
<input type="radio"/>	lf-business-analyst	AD group	Column	Include: tripdb.nyctaxi. [lpep_dropoff_dateti me, lpep_pickup_datetim e, vendorid]	Select
<input type="radio"/>	athena-okta- user@anycompany .com	AD user	Column	tripdb.nyctaxi.*	Select

### Paso 7: Verificar el acceso a través del cliente JDBC de Athena

Ahora puede utilizar un cliente JDBC para realizar una conexión de prueba a Athena como usuario SAML de Okta.

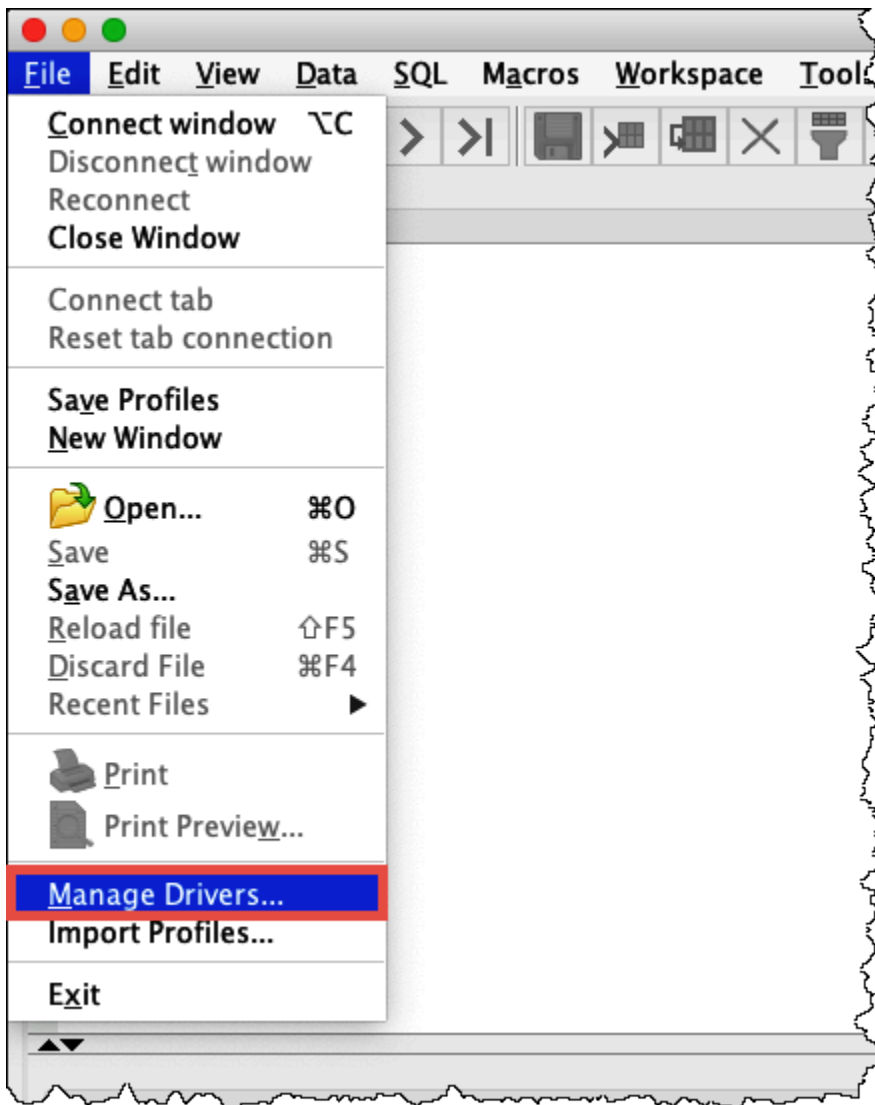
En esta sección, realiza las siguientes tareas:

- Preparar el cliente de prueba: descargue el controlador JDBC de Athena, instale SQL Workbench y agregue el controlador a Workbench. En este tutorial, se utiliza SQL Workbench para acceder a Athena a través de la autenticación de Okta y verificar los permisos de Lake Formation.
- En SQL Workbench:
  - Cree una conexión para el usuario Athena Okta.
  - Ejecute consultas de prueba como usuario Athena Okta.
  - Cree y pruebe una conexión para el usuario de analista empresarial.
- En la consola de Okta, agregue el usuario de analista empresarial al grupo de desarrolladores.
- En la consola Lake Formation, configure los permisos de tabla para el grupo de desarrolladores.
- En SQL Workbench, ejecute consultas de prueba como usuario de analista empresarial y verifique cómo afecta el cambio en los permisos a los resultados.

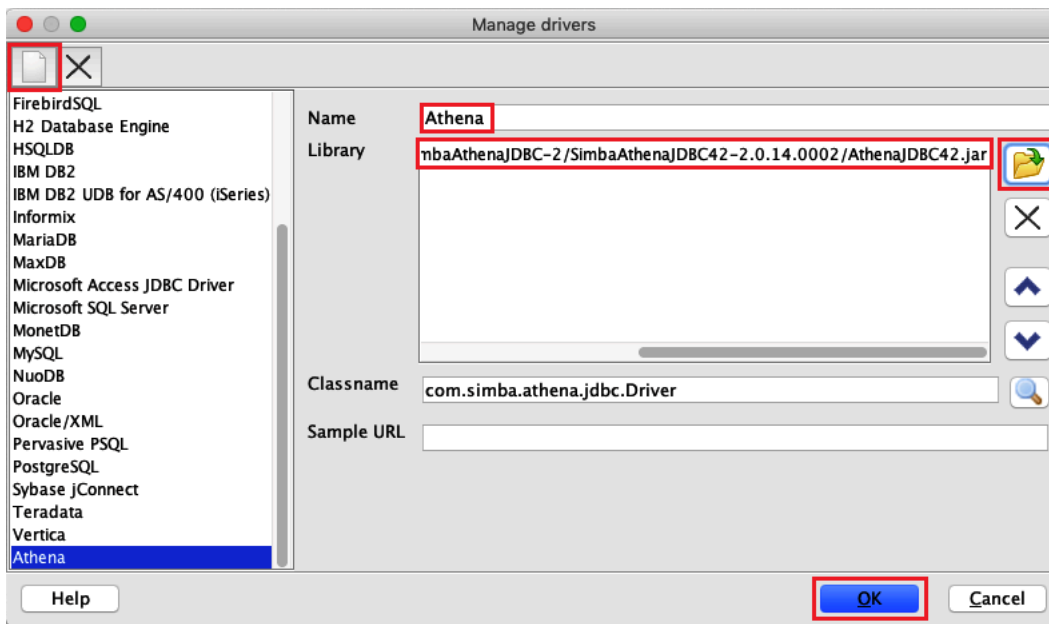
Para preparar el cliente de prueba

1. Descargue y extraiga el controlador JDBC de Athena compatible con Lake Formation (versión 2.0.14 o posterior) de [Conexión a Amazon Athena con JDBC](#).

2. Descargue e instale la herramienta de consulta SQL [SQL Workbench/J](#), disponible con una licencia modificada de Apache 2.0.
3. En SQL Workbench, elija File (Archivo) y, a continuación, elija Manage Drivers (Administrar controladores).



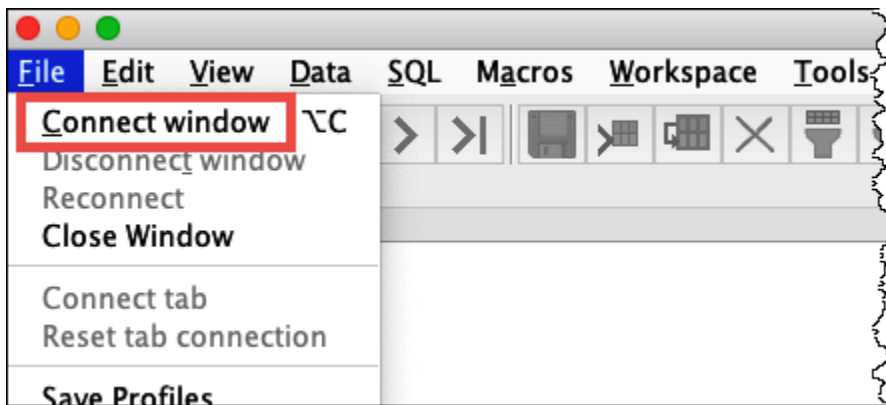
4. En el cuadro de diálogo Administrar controladores, siga los siguientes pasos:
  - a. Elija el icono de nuevo controlador.
  - b. En Nombre, escriba **Athena**.
  - c. En Library (Biblioteca), busque y elija el archivo `.jar` del controlador Simba JDBC de Athena que acaba de descargar.
  - d. Seleccione Aceptar.



Ahora puede crear y probar una conexión para el usuario de Athena Okta.

Para crear una conexión para el usuario de Okta

1. Elija File (Archivo), Connect window (Conectar ventana).



2. En el cuadro de diálogo Perfil de conexión, cree una conexión al ingresar la siguiente información:

- En el cuadro de nombre, ingrese **Athena\_Okta\_User\_Connection**.
- En Driver (Controlador), elija el controlador Simba JDBC de Athena.
- En URL, realice una de las siguientes operaciones:
  - Para utilizar una URL de conexión, ingrese una cadena de conexión de una sola línea. En el siguiente ejemplo se agregan saltos de línea por motivos de legibilidad.

```

jdbc:awsathena://AwsRegion=region-id;
S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/athena_results;
AwsCredentialsProviderClass=com.simba.athena.iamsupport.plugin.OktaCredentialsProvider;
user=athena-okta-user@anycompany.com;
password=password;
idp_host=okta-idp-domain;
App_ID=okta-app-id;
SSL_Insecure=true;
LakeFormationEnabled=true;

```

- Para utilizar una URL basada en perfiles de AWS, siga estos pasos:
  1. Configure un [perfil de AWS](#) que tenga un archivo de credenciales de AWS, como en el siguiente ejemplo.

```

[athena_lf_dev]
plugin_name=com.simba.athena.iamsupport.plugin.OktaCredentialsProvider
idp_host=okta-idp-domain
app_id=okta-app-id
uid=athena-okta-user@anycompany.com
pwd=password

```

2. En URL, ingrese una cadena de conexión de una sola línea, como el siguiente ejemplo. En el ejemplo se agregan saltos de línea por motivos de legibilidad.

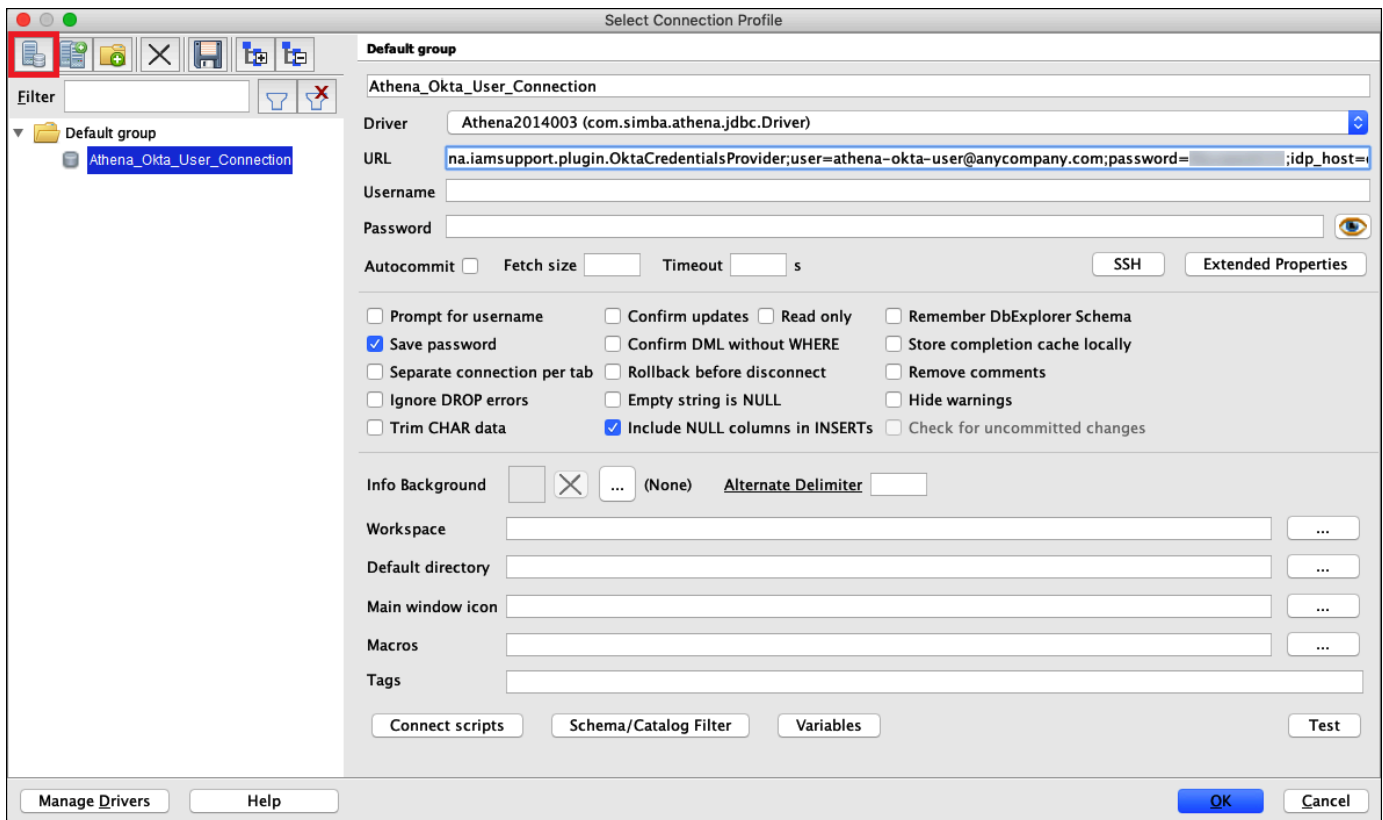
```

jdbc:awsathena://AwsRegion=region-id;
S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/athena_results;
profile=athena_lf_dev;
SSL_Insecure=true;
LakeFormationEnabled=true;

```

Tenga en cuenta que estos ejemplos son representaciones básicas de la URL necesaria para conectarse a Athena. Para ver una lista completa de los parámetros admitidos en la URL, consulte la [documentación de JDBC](#).

En la siguiente imagen se muestra un perfil de conexión de SQL Workbench que utiliza una dirección URL de conexión.



Ahora que ha establecido una conexión para el usuario de Okta, puede probarla recuperando algunos datos.

Para probar la conexión para el usuario de Okta

1. Elija Test (Probar) y, a continuación, verifique que la conexión se realiza correctamente.
2. Desde la ventana Instrucción de SQL Workbench, ejecute el siguiente comando SQL DESCRIBE. Verifique que se muestren todas las columnas.

```
DESCRIBE "tripdb"."nyctaxi"
```

The screenshot shows the SQL Workbench interface with the following details:

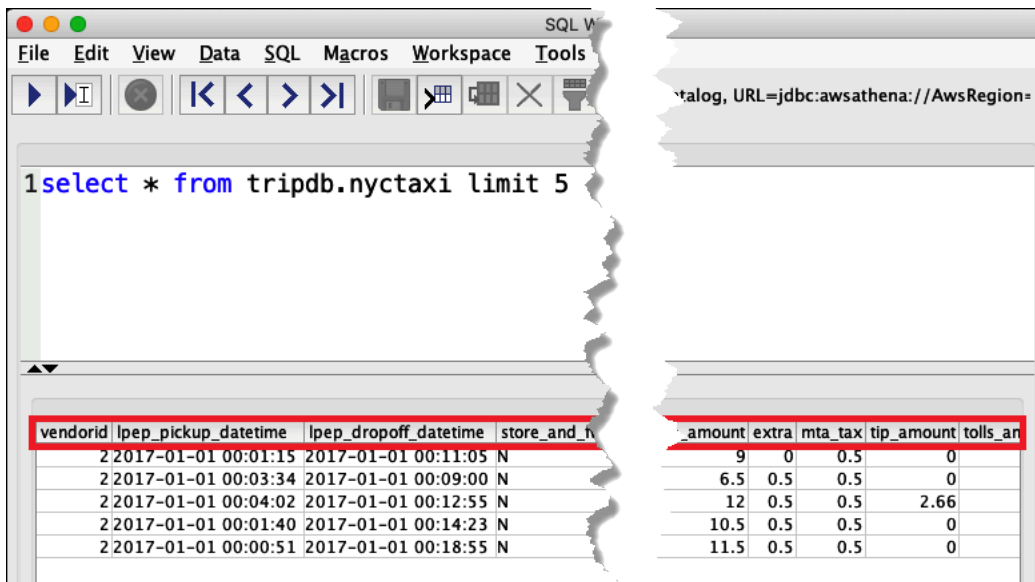
- Window title: SQL Workbench/J Athena\_AD\_User\_Connection - Default.wksp
- Menu bar: File, Edit, View, Data, SQL, Macros, Workspace, Tools, Help
- Toolbar: Navigation and execution icons.
- Statement 1: `1 describe "tripdb"."nyctaxi" |`
- Database Explorer 2: `tripdb.nyctaxi (EXTERNAL_TABLE)`
- Messages: Empty.
- Table Schema:

COLUMN_NAME	DATA_TYPE	PK	NULLABLE	DEFAULT	AUTOINCREMENT	COMPUTED	REMARKS	POSITION
vendorid	bigint	NO	YES		NO	NO		1
lpep_pickup_datetime	string(255)	NO	YES		NO	NO		2
lpep_dropoff_datetime	string(255)	NO	YES		NO	NO		3
store_and_fwd_flag	string(255)	NO	YES		NO	NO		4
ratecodeid	bigint	NO	YES		NO	NO		5
pulocationid	bigint	NO	YES		NO	NO		6
dolocationid	bigint	NO	YES		NO	NO		7
passenger_count	bigint	NO	YES		NO	NO		8
trip_distance	double	NO	YES		NO	NO		9
fare_amount	double	NO	YES		NO	NO		10
extra	double	NO	YES		NO	NO		11
mta_tax	double	NO	YES		NO	NO		12
tip_amount	double	NO	YES		NO	NO		13
tolls_amount	double	NO	YES		NO	NO		14
ehail_fee	string(255)	NO	YES		NO	NO		15
improvement_surcharge	double	NO	YES		NO	NO		16
total_amount	double	NO	YES		NO	NO		17
payment_type	bigint	NO	YES		NO	NO		18
trip_type	bigint	NO	YES		NO	NO		19

Bottom right corner: | L:1 C:29 |

- Desde la ventana Instrucción de SQL Workbench, ejecute el siguiente comando SQL SELECT. Verifique que se muestren todas las columnas.

```
SELECT * FROM tripdb.nyctaxi LIMIT 5
```



A continuación, debe verificar que el usuario `athena-ba-user`, como miembro del grupo `lf-business-analyst`, solo tiene acceso a las tres primeras columnas de la tabla que especificó anteriormente en Lake Formation.

Para verificar el acceso de `athena-ba-user`

1. En SQL Workbench, en el cuadro de diálogo Connection profile (Perfil de conexión), cree otro perfil de conexión.
  - En el nombre de perfil de conexión, ingrese **Athena\_Okta\_Group\_Connection**.
  - En Driver (Controlador), elija el controlador Simba JDBC de Athena.
  - En URL, realice una de las siguientes operaciones:
    - Para utilizar una URL de conexión, ingrese una cadena de conexión de una sola línea. En el siguiente ejemplo se agregan saltos de línea por motivos de legibilidad.

```
jdbc:awsathena://AwsRegion=region-id;  
S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/athena_results;  
AwsCredentialsProviderClass=com.simba.athena.iamsupport.plugin.OktaCredentialsProvider;  
user=athena-ba-user@anycompany.com;  
password=password;  
idp_host=okta-idp-domain;  
App_ID=okta-application-id;  
SSL_Insecure=true;
```

```
LakeFormationEnabled=true;
```

- Para utilizar una URL basada en perfiles de AWS, siga estos pasos:
  1. Configure un perfil de AWS que tenga un archivo de credenciales, como en el siguiente ejemplo.

```
[athena_lf_ba]
plugin_name=com.simba.athena.iamsupport.plugin.OktaCredentialsProvider
idp_host=okta-idp-domain
app_id=okta-application-id
uid=athena-ba-user@anycompany.com
pwd=password
```

2. En URL, ingrese una cadena de conexión de una sola línea, como la siguiente. En el ejemplo se agregan saltos de línea por motivos de legibilidad.

```
jdbc:awsathena://AwsRegion=region-id;  
S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/athena_results;  
profile=athena_lf_ba;  
SSL_Insecure=true;  
LakeFormationEnabled=true;
```

2. Elija Test (Probar) para confirmar que la conexión se realiza correctamente.
3. Desde la ventana SQL Statement (Instrucción SQL), ejecute los mismos comandos SQL DESCRIBE y SELECT que ejecutó antes y examine los resultados.

Como athena-ba-user es miembro del grupo lf-business-analyst, solo se devuelven las tres primeras columnas que especificó en la consola de Lake Formation.

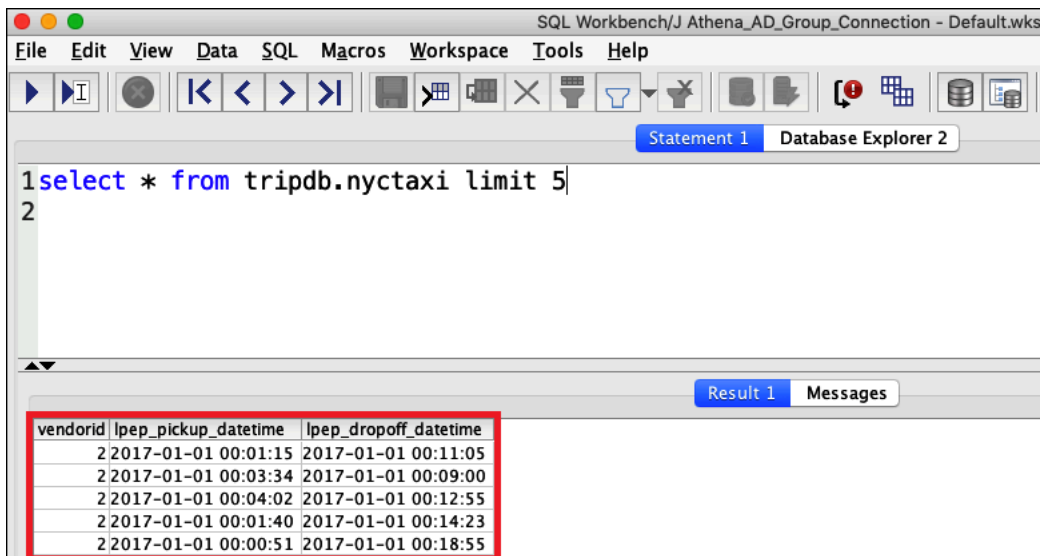
The screenshot shows the SQL Workbench/J interface. The SQL Statement window contains the following text:

```
1 describe tripdb.nyctaxi |
2
```

The Results window shows the output for the query:

COLUMN_NAME	DATA_TYPE	PK	NULLABLE	DEFAULT	AUTOINCREMENT	COMPUTED	REMARKS	POSITION
vendorid	bigint	NO	YES		NO	NO		1
lpep_pickup_datetime	string(255)	NO	YES		NO	NO		2
lpep_dropoff_datetime	string(255)	NO	YES		NO	NO		3

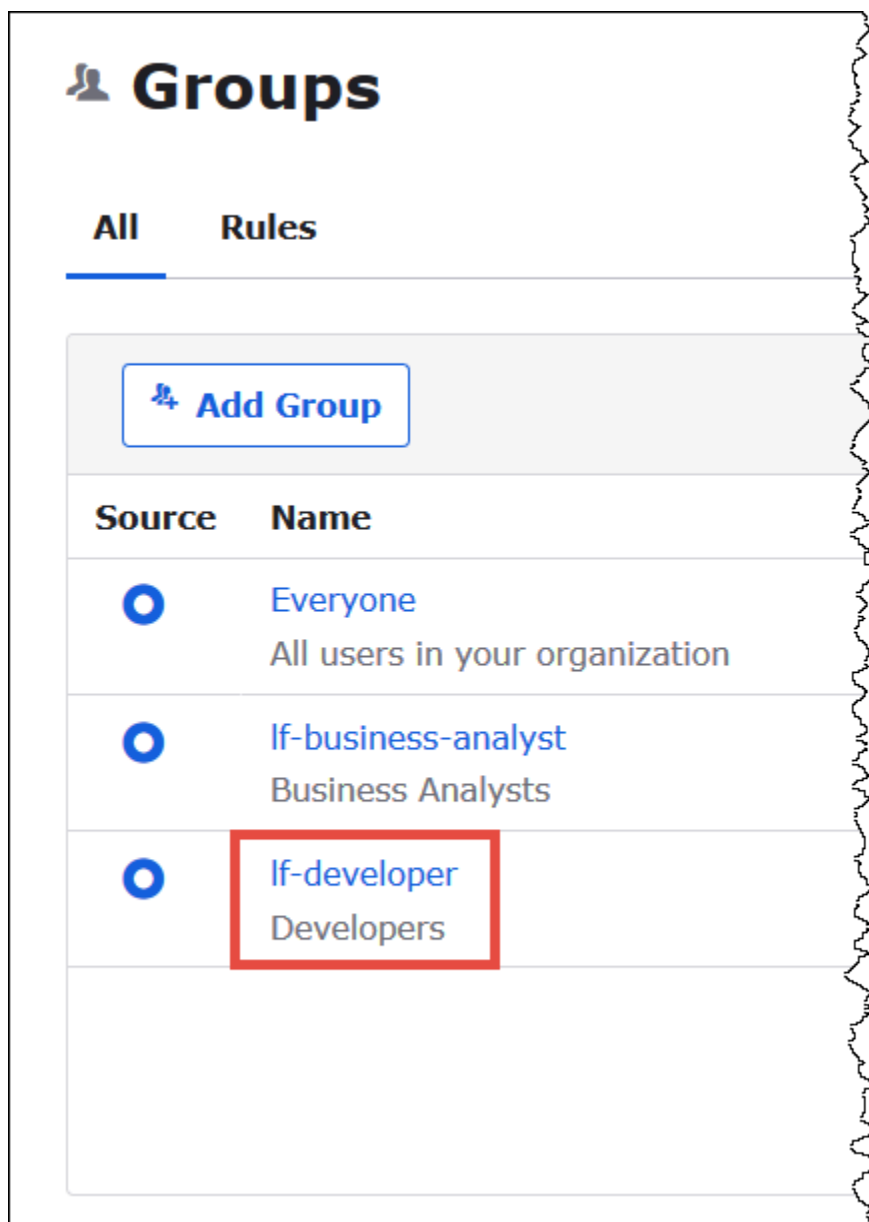




A continuación, vuelva a la consola de Okta para agregar el usuario `athena-ba-user` al grupo de Okta `lf-developer`.

Para agregar el usuario `athena-ba-user` al grupo `lf-developer`

1. Inicie sesión en la consola de Okta como usuario administrativo del dominio Okta asignado.
2. Elija Directory y, a continuación, elija Groups (Grupos).
3. En la página Groups (Grupos), elija el grupo `lf-developer`.



4. Elija Manage People (Administrar personas).
5. En la lista Not Members (No miembros), elija athena-ba-user para agregarlo al grupo If-developer.
6. Seleccione Guardar.

Ahora regrese a la consola de Lake Formation para configurar los permisos de tabla para el grupo If-developer.

## Para configurar los permisos de tabla para lf-developer-group

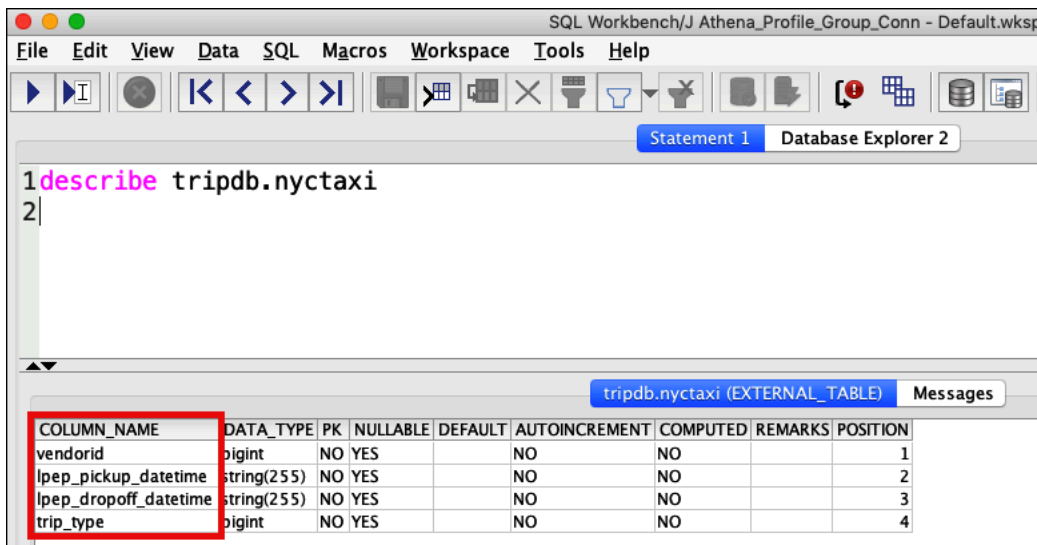
1. Inicie sesión en la consola de Lake Formation como administrador del lago de datos.
2. En el panel de navegación, elija Tablas.
3. Seleccione la tabla nyctaxi.
4. Elija Actions (Acciones), Grant (Conceder).
5. En el cuadro de diálogo Conceder permisos, ingrese la siguiente información:
  - En SAML and Amazon QuickSight users and groups (Usuarios y grupos de SAML y Amazon QuickSight), ingrese el ARN de grupo lf-developer SAML de Okta con el siguiente formato:
  - En Columns (Columnas), Choose filter type (Elegir tipo de filtro), elija Include Columns (Incluir columnas).
  - Elija la columna trip\_type.
  - En Table permissions (Permisos de tabla), elija SELECT.
6. Elija Grant (Conceder).

Ahora puede utilizar SQL Workbench para verificar el cambio en los permisos para el grupo lf-developer. El cambio debe reflejarse en los datos disponibles para el usuario athena-ba-user, que ahora es miembro del grupo lf-developer.

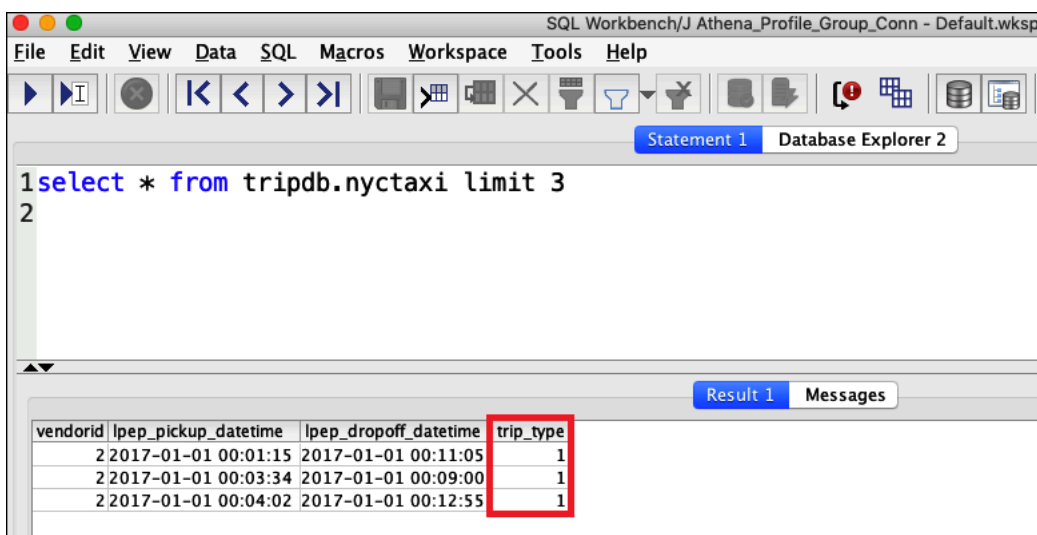
## Para verificar el cambio en los permisos de athena-ba-user

1. Cierre el programa SQL Workbench y, a continuación, vuelva a abrirlo.
2. Conéctese al perfil de athena-ba-user.
3. En la ventana Statement (Instrucción), ejecute las mismas instrucciones SQL que ejecutó anteriormente:

Esta vez, se muestra la columna trip\_type.



Como athena-ba-user ahora es miembro de los grupos lf-developer e lf-business-analyst, la combinación de permisos de Lake Formation para esos grupos determina las columnas que se devuelven.



## Conclusión

En este tutorial configuró la integración de Athena con AWS Lake Formation utilizando Okta como proveedor SAML. Utilizó Lake Formation e IAM para controlar los recursos que están disponibles para el usuario SAML en el catálogo de datos AWS Glue del lago de datos.

## Recursos relacionados

Para obtener información relacionada, consulte los siguientes recursos.

- [Conexión a Amazon Athena con JDBC](#)
- [Habilitación de acceso federado a la API de Athena](#)
- [Guía para desarrolladores de AWS Lake Formation](#)
- [Concesión y revocación de permisos de catálogo de datos](#) en la Guía para desarrolladores de AWS Lake Formation.
- [Federación y proveedores de identidad](#) en la Guía del usuario de IAM.
- [Creación de proveedores de identidad SAML de IAM](#) en la Guía del usuario de IAM.
- [Enabling federation to AWS using Windows Active Directory, ADFS, and SAML 2.0](#) (Habilitación de la federación de AWS por medio de Windows Active Directory, ADFS y SAML 2.0) en el blog de seguridad de .

## Administración de la carga de trabajo

Puede utilizar las características de grupo de trabajo, administración de capacidad, ajuste del rendimiento, soporte de compresión, etiquetas y Service Quotas de Athena para administrar su carga de trabajo.

### Temas

- [Uso de grupos de trabajo para controlar el acceso a las consultas y los costos](#)
- [Administración de la capacidad de procesamiento de consultas](#)
- [Ajuste del rendimiento en Athena](#)
- [Compatibilidad con la compresión de Athena](#)
- [Etiquetado de recursos de Athena](#)
- [Service Quotas](#)

## Uso de grupos de trabajo para controlar el acceso a las consultas y los costos

Utilice grupos de trabajo para separar usuarios, equipos, aplicaciones o cargas de trabajo con el fin de establecer límites en la cantidad de datos que puede procesar cada consulta o todo el grupo de trabajo, así como para supervisar los costos. Como los grupos de trabajo funcionan como recursos, para controlar el acceso a ellos puede usar las políticas basadas en identidades de nivel de recursos. También puede ver métricas relacionadas con las consultas en Amazon CloudWatch, controlar

los costos mediante la configuración de los límites de la cantidad de datos escaneados, crear los umbrales y desencadenar acciones, como Amazon SNS, cuando se superan estos umbrales.

Para controlar aún más los costos, puede crear reservas de capacidad con el número de unidades de procesamiento de datos que especifique y agregar uno o más grupos de trabajo a la reserva. Para obtener más información, consulte [Administración de la capacidad de procesamiento de consultas](#).

Los grupos de trabajo se integran con IAM, CloudWatch y Amazon Simple Notification Service y los [informes de costos y uso de AWS](#) de la siguiente manera:

- Las políticas basadas en identidades de IAM con permisos de nivel de recursos controlan quién ejecuta consultas en un grupo de trabajo.
- Si habilita las métricas de consulta, Athena publica las métricas de consulta del grupo de trabajo en CloudWatch.
- En Amazon SNS, puede crear temas de Amazon SNS que emiten alarmas para los usuarios de grupos de trabajo especificados cuando los controles de uso de datos de las consultas en un grupo de trabajo superan los umbrales establecidos.
- Cuando etiqueta un grupo de trabajo con una etiqueta configurada como etiqueta de asignación de costos en la consola de Administración de facturación y costos, los costos asociados con la ejecución de consultas en ese grupo de trabajo aparecen en los informes de costos y uso con esa etiqueta de asignación de costos.

## Temas

- [Uso de grupos de trabajo para la ejecución de consultas](#)
- [Control de costos y supervisión de consultas con métricas y eventos de CloudWatch](#)

Consulte también la publicación del Blog de macrodatos de AWS [Consultas separadas y administración de costos mediante grupos de trabajo de Amazon Athena](#), que muestra cómo utilizar grupos de trabajo para separar cargas de trabajo, controlar el acceso de usuarios y administrar el uso y los costos de las consultas.

## Uso de grupos de trabajo para la ejecución de consultas

Le recomendamos que utilice grupos de trabajo para aislar consultas para equipos, aplicaciones o diferentes cargas de trabajo. Por ejemplo, puede crear grupos de trabajo independientes para dos equipos diferentes en su organización. También puede separar cargas de trabajo. Por ejemplo, puede crear dos grupos de trabajo independientes, uno para aplicaciones programadas

automatizadas, como, por ejemplo, la generación de informes y otro para uso ad-hoc por parte de analistas. Puede cambiar entre grupos de trabajo.

## Temas

- [Beneficios de utilizar grupos de trabajo](#)
- [Cómo funcionan los grupos de trabajo](#)
- [Configuración de los grupos de trabajo](#)
- [Políticas de IAM para acceder a los grupos de trabajo](#)
- [Configuración del grupo de trabajo](#)
- [Administración de los grupos de trabajo](#)
- [Uso de grupos de trabajo de Athena habilitados para IAM Identity Center](#)
- [API de grupos de trabajo de Athena](#)
- [Solución de problemas de grupos de trabajo](#)

## Beneficios de utilizar grupos de trabajo

Los grupos de trabajo le permiten:

Aislar a usuarios, equipos, aplicaciones o cargas de trabajo en grupos.

Cada grupo de trabajo tiene su propio historial de consultas diferenciado y una lista de las consultas guardadas. Para obtener más información, consulte [Cómo funcionan los grupos de trabajo](#).

Para todas las consultas en el grupo de trabajo, puede elegir configurar la configuración del grupo de trabajo. Incluyen una ubicación de Simple Storage Service (Amazon S3) para almacenar los resultados de las consultas, el propietario del bucket esperado, el cifrado y el control de los objetos escritos en el bucket de resultados de las consultas. También puede aplicar la configuración del grupo de trabajo. Para obtener más información, consulte [Configuración del grupo de trabajo](#).

Implemente las restricciones de costos.

Puede configurar dos tipos de restricciones de costos para las consultas en un grupo de trabajo:

- El Límite por consulta consiste en un umbral para la cantidad de datos escaneados para cada consulta. Athena cancela las

consultas cuando superan el umbral especificado. El límite se aplica a cada consulta en ejecución dentro de un grupo de trabajo. Puede establecer solo un límite por consulta y actualizarlo si es necesario.

- Límite por grupo de trabajo es un umbral que puede establecer para cada grupo de trabajo para la cantidad de datos escaneados por consultas en el grupo de trabajo. Cuando se supera un umbral, se activa una alarma de Amazon SNS que desencadena una acción de su elección, como, por ejemplo, el envío de un correo electrónico a un usuario especificado. Puede configurar varios límites por grupo de trabajo para cada grupo de trabajo.

Para conocer los pasos en detalle, consulte [Configuración de los límites de control del uso de datos](#).

Hacer un seguimiento de las métricas relacionadas con las consultas para todas las consultas de grupos de trabajo en CloudWatch.

Si configura el grupo de trabajo para publicar métricas, Athena publica en CloudWatch toda consulta que se ejecuta en un grupo de trabajo. Puede [ver las métricas de consultas](#) para cada uno de sus grupos de trabajo dentro de la consola de Athena. En CloudWatch, puede crear paneles personalizados y establecer umbrales y alarmas en estas métricas.

## Cómo funcionan los grupos de trabajo

En Athena, los grupos de trabajo tienen las siguientes características:

- De forma predeterminada, cada cuenta tiene un grupo de trabajo principal y los permisos predeterminados permiten que todos los usuarios autenticados accedan a este grupo de trabajo. No se puede eliminar el grupo de trabajo principal.
- Cada grupo de trabajo que cree muestra las consultas guardadas y el historial de consultas solo para las consultas ejecutadas en él, y no para todas las consultas de la cuenta. Esto separa sus consultas de otras consultas dentro de una cuenta y mejora la eficiencia a la hora de localizar sus propias consultas guardadas y las consultas en el historial.
- La deshabilitación de un grupo de trabajo evita que se ejecuten consultas en él, hasta que vuelve a habilitarlo. Las consultas enviadas a un grupo de trabajo deshabilitado dan error, hasta que vuelve a habilitarlo.



- Si tiene los permisos adecuados, puede eliminar un grupo de trabajo vacío, así como un grupo de trabajo que contiene consultas guardadas. En este caso, antes de eliminar un grupo de trabajo, Athena le avisa que se eliminarán las consultas guardadas. Antes de eliminar un grupo de trabajo al que otros usuarios tienen acceso, asegúrese de que sus usuarios tengan acceso a otros grupos de trabajo en los que puedan seguir ejecutando consultas.
- Puede configurar la configuración de todo el grupo de trabajo y aplicar su uso por parte de todas las consultas que se ejecutan en un grupo de trabajo. La configuración incluye una ubicación en Simple Storage Service (Amazon S3) para almacenar los resultados de las consultas, el propietario del bucket esperado, el cifrado y el control de los objetos escritos en el bucket de resultados de las consultas.

#### Important

Cuando aplica la configuración en todo el grupo de trabajo, todas las consultas que se ejecutan en este grupo de trabajo utilizan la configuración del grupo de trabajo. Esto ocurre incluso si su configuración del cliente es diferente de la configuración del grupo de trabajo. Para obtener más información, consulte [La configuración del grupo de trabajo invalida la configuración del cliente](#).

### Limitaciones de los grupos de trabajo

- Puede crear hasta 1000 grupos de trabajo por región en su cuenta.
- No se puede eliminar el grupo de trabajo principal.
- Puede abrir hasta diez pestañas de consulta dentro de cada grupo de trabajo. Al cambiar entre grupos de trabajo, la pestaña de la consulta permanece abierta para hasta tres grupos de trabajo.

### Configuración de los grupos de trabajo

La configuración de grupos de trabajo implica su creación y el establecimiento de permisos para su uso. En primer lugar, decida qué grupos de trabajo necesita su organización y créelos. A continuación, configure las políticas de grupos de trabajo de IAM que controlan el acceso de los usuarios y acciones en un recurso de `workgroup`. Los usuarios con acceso a estos grupos de trabajo pueden ejecutar ahora consultas en ellos.

**Note**

Utilice estas tareas para configurar grupos de trabajo cuando comienza a utilizarlos por primera vez. Si la cuenta de Athena ya utiliza grupos de trabajo, cada uno de los usuarios de la cuenta necesita permisos para ejecutar consultas en uno o varios grupos de trabajo de la cuenta. Antes de ejecutar consultas, verifique su política de IAM para ver a qué grupos de trabajo puede acceder, ajustar su política si es necesario y [cambiar](#) al grupo de trabajo que va a utilizar.

De forma predeterminada, si no ha creado ningún grupo de trabajo, todas las consultas en su cuenta se ejecutan en el grupo de trabajo principal.

Athena muestra el grupo de trabajo actual en la opción Grupo de trabajo en la parte superior derecha de la consola. Puede utilizar esta opción para cambiar de grupo de trabajo. Cuando ejecuta consultas, se ejecutan en el grupo de trabajo actual. Puede ejecutar consultas en el contexto de un grupo de trabajo en la consola, mediante las operaciones de la API, la interfaz de línea de comandos o una aplicación cliente con el controlador JDBC u ODBC. Cuando tiene acceso a un grupo de trabajo, puede ver la configuración del grupo de trabajo, la métrica y los límites de control de uso de datos. Con permisos adicionales, puede editar la configuración y los límites de control de uso de datos.

Para configurar grupos de trabajo

1. Decida qué grupos de trabajo va a crear. Por ejemplo, puede decidir lo siguiente:
  - Quién puede ejecutar consultas en cada grupo de trabajo y quién es el propietario de la configuración del grupo de trabajo. Esto determina las políticas de IAM que crea. Para obtener más información, consulte [Políticas de IAM para acceder a los grupos de trabajo](#).
  - Qué ubicaciones de Amazon S3 va a utilizar para los resultados de la consulta para las consultas que se ejecutan en cada grupo de trabajo. La ubicación debe existir en Simple Storage Service (Amazon S3) para poder especificarla para los resultados de las consultas del grupo de trabajo. Todos los usuarios que utilizan un grupo de trabajo deben tener acceso a esta ubicación. Para obtener más información, consulte [Configuración del grupo de trabajo](#).
  - Si el propietario del bucket de resultados de las consultas de Simple Storage Service (Amazon S3) tiene control total sobre los nuevos objetos que se escriben en el bucket o no. Por ejemplo, si la ubicación del resultado de las consultas es propiedad de otra cuenta, puede

conceder la propiedad y el control total sobre los resultados de las consultas a la otra cuenta. Para obtener más información, consulte [AclConfiguration](#).

- Especifique el ID de la Cuenta de AWS que espera sea el propietario del bucket de ubicación de salida. Se trata de una medida de seguridad añadida opcional. Si el ID de cuenta del propietario del bucket no coincide con el ID que se especifique aquí, los intentos de generar el bucket fallarán. Para obtener más información, consulte [Verificación de la propiedad del bucket con la condición de propietario del bucket](#) en la Guía del usuario de Amazon S3. Esta configuración no se aplica a las instrucciones CTAS, INSERT INTO o UNLOAD.
  - Qué configuración de cifrado es necesaria y qué grupos de trabajo tienen consultas que deben cifrarse. Le recomendamos que cree grupos de trabajo independientes para consultas cifradas y no cifradas. De esta forma, puede forzar el cifrado de un grupo de trabajo que se aplica a todas las consultas que se ejecutan en ella. Para obtener más información, consulte [Cifrado de los resultados de las consultas de Athena en Amazon S3](#).
2. Cree los grupos de trabajo necesarios y añádales etiquetas. Para ver los pasos, consulte [Creación de un grupo de trabajo](#).
  3. Cree políticas de IAM para sus usuarios, grupos o roles para habilitar el acceso de estos a grupos de trabajo. Las políticas establecen la pertenencia a grupos de trabajo y el acceso a acciones en un recurso de workgroup. Para conocer los pasos en detalle, consulte [Políticas de IAM para acceder a los grupos de trabajo](#). Para ver ejemplos de políticas de JSON, consulte [Acceso a grupos de trabajo y etiquetas](#).
  4. Defina la configuración del grupo de trabajo. Especifique una ubicación en Simple Storage Service (Amazon S3) para almacenar los resultados de las consultas, y especifique opcionalmente el propietario del bucket esperado, la configuración de cifrado y el control de los objetos escritos en el bucket de resultados de las consultas. Puede aplicar la configuración del grupo de trabajo. Para obtener más información, consulte [configuración del grupo de trabajo](#).

#### Important

Si [invalida la configuración del cliente](#), Athena utilizará la configuración del grupo de trabajo. Esto afecta a consultas que ejecuta en la consola, mediante los controladores, la interfaz de línea de comandos o las operaciones de la API.

Si bien las consultas se siguen ejecutando, la automatización integrada en función de la disponibilidad de resultados en un determinado bucket de Amazon S3 podría interrumpirse. Le recomendamos que informe a los usuarios antes de la invalidación. Una vez que se establece la configuración del grupo de trabajo para la invalidación,

puede omitir la especificación de la configuración del cliente en los controladores o en la API.

5. Notifique a los usuarios qué grupos de trabajo se deben utilizar para ejecutar consultas. Envíe un correo electrónico para informar a los usuarios de las cuentas acerca de qué nombres de grupo de trabajo pueden utilizar, las políticas de IAM necesarias y la configuración del grupo de trabajo.
6. Configure los límites de control de costos, también conocidos como límites de control de uso de datos para las consultas y grupos de trabajo. Para que se le notifique cuando se supera un umbral, cree un tema de Amazon SNS y configure suscripciones. Para conocer los pasos en detalle, consulte [Configuración de los límites de control del uso de datos](#) e [Introducción a Amazon SNS](#) en la Guía para desarrolladores de Amazon Simple Notification Service.
7. Cambie al grupo de trabajo para poder ejecutar consultas. Para ejecutar consultas, cambie al grupo de trabajo adecuado. Para conocer los pasos en detalle, consulte [the section called “Especificar un grupo de trabajo en el que se ejecutarán consultas”](#).

## Políticas de IAM para acceder a los grupos de trabajo

Para controlar el acceso a los grupos de trabajo, utilice permisos de IAM de nivel de recursos o políticas de IAM basadas en identidad. Siempre que utilice políticas de IAM, asegúrese de seguir las prácticas recomendadas de IAM. Para más información, consulte [Prácticas recomendadas de seguridad en IAM](#) en la Guía del usuario de IAM.

### Note

Para acceder a los grupos de trabajo habilitados para la propagación de identidad de confianza, los usuarios del IAM Identity Center deben estar asignados al `IdentityCenterApplicationArn` que se devuelve en la respuesta de la acción de la API [GetWorkGroup](#) de Athena.

El siguiente procedimiento es específico de Athena.

Para obtener información específica sobre IAM, consulte los enlaces que se enumeran al final de esta sección. Para obtener información sobre políticas de grupos de trabajo JSON de ejemplo, consulte [Ejemplos de políticas de grupos de trabajo](#).

Para utilizar el editor visual de la consola de IAM para crear una política de grupo de trabajo

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación de la izquierda, elija Políticas y, a continuación, elija Crear política.
3. En la pestaña Editor visual, elija Elegir un servicio. A continuación, elija un servicio de Athena para agregar a la política.
4. Elija Seleccionar acciones y, a continuación, elija las acciones que desea añadir a la política. El editor visual muestra las acciones disponibles en Athena. Para obtener más información, consulte [Acciones, recursos y claves de condición de Amazon Athena](#) en la Referencia de autorizaciones de servicio.
5. Elija Añadir acciones para escribir una acción específica o utilice comodines (\*) para especificar varias acciones.

De forma predeterminada, la política que está creando permite las acciones que usted elija. Si eligió una o más acciones que admiten permisos en el nivel de recursos para el recurso `workgroup` en Athena, el editor visual enumera el recurso `workgroup`.

6. Elija Recursos para especificar grupos de trabajo específicos para su política. Para ver políticas de JSON grupos de trabajo de ejemplo, consulte [Ejemplos de políticas de grupos de trabajo](#).
7. Especifique el recurso `workgroup` como se indica a continuación:

```
arn:aws:athena:<region>:<user-account>:workgroup/<workgroup-name>
```

8. Elija Revisar la política y, a continuación, escriba un Nombre y una Descripción (opcional) para la política que está creando. Revise el resumen de la política para asegurarse de que ha concedido los permisos deseados.
9. Elija Crear política para guardar la nueva política.
10. Adjunte esta política basada en identidad a un usuario, grupo o rol.

Para obtener más información, consulte los siguientes temas en la Referencia de autorizaciones de servicio y la Guía del usuario de IAM:

- [Acciones, recursos y claves de condición de Amazon Athena](#)
- [Creación de políticas con el editor visual](#)
- [Adición y eliminación de políticas de IAM](#)

- [Control del acceso a los recursos](#)

Para ver políticas de JSON grupos de trabajo de ejemplo, consulte [Ejemplos de políticas de grupos de trabajo](#).

Para obtener una lista completa de las acciones de Amazon Athena, consulte los nombres de acciones de la API en la [Referencia de API de Amazon Athena](#).

### Ejemplos de políticas de grupos de trabajo

En esta sección se incluyen ejemplos de políticas que puede utilizar para habilitar varias acciones en grupos de trabajo. Siempre que utilice políticas de IAM, asegúrese de seguir las prácticas recomendadas de IAM. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM.

Un grupo de trabajo es un recurso de IAM administrado por Athena. Por tanto, si su política de grupos de trabajo utiliza acciones que toman `workgroup` como entrada, deberá especificar el ARN del grupo de trabajo de la siguiente manera:

```
"Resource": [arn:aws:athena:<region>:<user-account>:workgroup/<workgroup-name>]
```

Donde `<workgroup-name>` es el nombre de su grupo de trabajo. Por ejemplo, para el grupo de trabajo denominado `test_workgroup`, especifíquelo como un recurso tal y como se indica a continuación:

```
"Resource": ["arn:aws:athena:us-east-1:123456789012:workgroup/test_workgroup"]
```

Para obtener una lista completa de las acciones de Amazon Athena, consulte los nombres de acciones de la API en la [Referencia de API de Amazon Athena](#). Para obtener más información sobre las políticas de IAM, consulte [Creación de políticas con el editor visual](#) en la Guía del usuario de IAM. Para obtener más información acerca de la creación de políticas de IAM para grupos de trabajo, consulte [Políticas de IAM para acceder a los grupos de trabajo](#).

- [Example policy for full access to all workgroups](#)
- [Example policy for full access to a specified workgroup](#)
- [Example policy for running queries in a specified workgroup](#)
- [Example policy for running queries in the primary workgroup](#)
- [Example policy for management operations on a specified workgroup](#)

- [Example policy for listing workgroups](#)
- [Example policy for running and stopping queries in a specific workgroup](#)
- [Example policy for working with named queries in a specific workgroup](#)
- [Example policy for working with Spark notebooks](#)

### Example Ejemplo de política para el acceso completo a todos los grupos de trabajo

La siguiente política permite acceso completo a todos los recursos de grupos de trabajo que podrían existir en la cuenta. Le recomendamos que utilice esta política para aquellos usuarios en su cuenta que deben administrar grupos de trabajo para los demás usuarios.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:*"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

### Example Ejemplo de política para el acceso completo a un grupo de trabajo especificado

La siguiente política permite acceso completo al recurso del grupo de trabajo específico individual, denominado `workgroupA`. Puede utilizar esta política para los usuarios con control total sobre un grupo de trabajo en particular.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:ListEngineVersions",
        "athena:ListWorkGroups",

```

```

        "athena:ListDataCatalogs",
        "athena:ListDatabases",
        "athena:GetDatabase",
        "athena:ListTableMetadata",
        "athena:GetTableMetadata"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "athena:BatchGetQueryExecution",
        "athena:GetQueryExecution",
        "athena:ListQueryExecutions",
        "athena:StartQueryExecution",
        "athena:StopQueryExecution",
        "athena:GetQueryResults",
        "athena:GetQueryResultsStream",
        "athena:CreateNamedQuery",
        "athena:GetNamedQuery",
        "athena:BatchGetNamedQuery",
        "athena:ListNamedQueries",
        "athena>DeleteNamedQuery",
        "athena:CreatePreparedStatement",
        "athena:GetPreparedStatement",
        "athena:ListPreparedStatements",
        "athena:UpdatePreparedStatement",
        "athena>DeletePreparedStatement"
    ],
    "Resource": [
        "arn:aws:athena:us-east-1:123456789012:workgroup/workgroupA"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "athena>DeleteWorkGroup",
        "athena:UpdateWorkGroup",
        "athena:GetWorkGroup",
        "athena:CreateWorkGroup"
    ],
    "Resource": [
        "arn:aws:athena:us-east-1:123456789012:workgroup/workgroupA"
    ]
}

```



```

    }
  ]
}

```

## Example Ejemplo de política para la ejecución de consultas en un grupo de trabajo especificado

En la siguiente política, un usuario puede ejecutar consultas en el `workgroupA` especificado y verlas. El usuario no tiene permiso para realizar las tareas de administración para el grupo de trabajo en sí, como, por ejemplo, actualizar o eliminar.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:ListEngineVersions",
        "athena:ListWorkGroups",
        "athena:ListDataCatalogs",
        "athena:ListDatabases",
        "athena:GetDatabase",
        "athena:ListTableMetadata",
        "athena:GetTableMetadata"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "athena:GetWorkGroup",
        "athena:BatchGetQueryExecution",
        "athena:GetQueryExecution",
        "athena:ListQueryExecutions",
        "athena:StartQueryExecution",
        "athena:StopQueryExecution",
        "athena:GetQueryResults",
        "athena:GetQueryResultsStream",
        "athena:CreateNamedQuery",
        "athena:GetNamedQuery",
        "athena:BatchGetNamedQuery",
        "athena:ListNamedQueries",
        "athena>DeleteNamedQuery",
        "athena>CreatePreparedStatement",

```

```

        "athena:GetPreparedStatement",
        "athena:ListPreparedStatements",
        "athena:UpdatePreparedStatement",
        "athena>DeletePreparedStatement"
    ],
    "Resource": [
        "arn:aws:athena:us-east-1:123456789012:workgroup/workgroupA"
    ]
}
]
}

```

Example Ejemplo de política para la ejecución de consultas en el grupo de trabajo principal

Puede modificar el ejemplo anterior para permitir que un usuario determinado también ejecute consultas en el grupo de trabajo principal.

#### Note

Le recomendamos que agregue el recurso de grupo de trabajo principal para todos los usuarios que, de lo contrario, se configuran para ejecutar consultas en los grupos de trabajo designados. Agregar este recurso a las políticas de usuario del grupo de trabajo es útil en caso de eliminación o deshabilitación del grupo de trabajo designado. En este caso, pueden seguir ejecutando consultas en el grupo de trabajo principal.

Para permitir que los usuarios de su cuenta ejecuten consultas en el grupo de trabajo principal, agregue una línea que contenga el ARN del grupo de trabajo principal a la sección de recursos de [Example policy for running queries in a specified workgroup](#), como en el siguiente ejemplo.

```
arn:aws:athena:us-east-1:123456789012:workgroup/primary"
```

Example Ejemplo de política para la realización de operaciones de administración en un grupo de trabajo especificado

En la siguiente política, se permite a un usuario crear, eliminar, obtener detalles y actualizar un grupo de trabajo `test_workgroup`.

```

{
  "Version": "2012-10-17",
  "Statement": [

```

```

    {
      "Effect": "Allow",
      "Action": [
        "athena:ListEngineVersions"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "athena:CreateWorkGroup",
        "athena:GetWorkGroup",
        "athena>DeleteWorkGroup",
        "athena:UpdateWorkGroup"
      ],
      "Resource": [
        "arn:aws:athena:us-east-1:123456789012:workgroup/test_workgroup"
      ]
    }
  ]
}

```

### Example Ejemplo de política para crear una lista de los grupos de trabajo

La siguiente política permite a todos los usuarios crear una lista de todos los grupos de trabajo:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:ListWorkGroups"
      ],
      "Resource": "*"
    }
  ]
}

```

### Example Ejemplo de política para la ejecución y la detención de consultas en un grupo de trabajo específico

En esta política, se permite a un usuario ejecutar consultas en el grupo de trabajo:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:StartQueryExecution",
        "athena:StopQueryExecution"
      ],
      "Resource": [
        "arn:aws:athena:us-east-1:123456789012:workgroup/test_workgroup"
      ]
    }
  ]
}
```

Example Ejemplo de política para trabajar con consultas con nombre en un grupo de trabajo específico

En la siguiente política, un usuario tiene permisos para crear, eliminar y obtener información acerca de consultas con nombre en el grupo de trabajo especificado:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:CreateNamedQuery",
        "athena:GetNamedQuery",
        "athena>DeleteNamedQuery"
      ],
      "Resource": [
        "arn:aws:athena:us-east-1:123456789012:workgroup/test_workgroup"
      ]
    }
  ]
}
```

Example Ejemplo de política para trabajar con cuadernos de Spark en Athena

Use una política como la siguiente para trabajar con los cuadernos de Spark en Athena.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreatingWorkGroupWithDefaults",
      "Action": [
        "athena:CreateWorkGroup",
        "s3:CreateBucket",
        "iam:CreateRole",
        "iam:CreatePolicy",
        "iam:AttachRolePolicy",
        "s3:GetBucketLocation",
        "athena:ImportNotebook"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:athena:us-east-1:123456789012:workgroup/Demo*",
        "arn:aws:s3:::123456789012-us-east-1-athena-results-bucket-*",
        "arn:aws:iam::123456789012:role/service-role/
AWSAthenaSparkExecutionRole-*",
        "arn:aws:iam::123456789012:policy/service-role/
AWSAthenaSparkRolePolicy-*"
      ]
    },
    {
      "Sid": "AllowRunningCalculations",
      "Action": [
        "athena:ListWorkGroups",
        "athena:GetWorkGroup",
        "athena:StartSession",
        "athena:CreateNotebook",
        "athena:ListNotebookMetadata",
        "athena:ListNotebookSessions",
        "athena:GetSessionStatus",
        "athena:GetSession",
        "athena:GetNotebookMetadata",
        "athena:CreatePresignedNotebookUrl"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:athena:us-east-1:123456789012:workgroup/Demo*"
    },
    {
      "Sid": "AllowListWorkGroupAndEngineVersions",

```

```
    "Action": [
      "athena:ListWorkGroups",
      "athena:ListEngineVersions"
    ],
    "Effect": "Allow",
    "Resource": "*"
  }
]
```

## Configuración del grupo de trabajo

Cada grupo de trabajo tiene la siguiente configuración:

- Un nombre único. Puede contener de 1 a 128 caracteres, incluidos los caracteres alfanuméricos, guiones y guiones bajos. Después de crear un grupo de trabajo, no podrá cambiar el nombre. Sin embargo, puede crear un nuevo grupo de trabajo con la misma configuración y un nombre diferente.
- Configuración que se aplica a todas las consultas que se ejecutan en el grupo de trabajo. Incluyen:
  - Una ubicación en Amazon S3 para almacenar los resultados de las consultas para todas las consultas que se ejecutan en este grupo de trabajo. Esta ubicación debe existir previamente para poder especificarla para el grupo de trabajo cuando lo crea. Para obtener información sobre la creación de un bucket de Amazon S3, consulte [Creación de un bucket](#).
  - Control del propietario del bucket de los resultados de las consultas: si el propietario del bucket de resultados de Simple Storage Service (Amazon S3) tiene control total sobre los nuevos objetos que se escriben en el bucket o no. Por ejemplo, si la ubicación del resultado de las consultas es propiedad de otra cuenta, puede conceder la propiedad y el control total sobre los resultados de las consultas a la otra cuenta.
  - Propietario esperado del bucket: el ID de la Cuenta de AWS que se espera que sea el propietario del bucket de resultados de la consulta. Se trata de una medida de seguridad añadida. Si el ID de cuenta del propietario del bucket no coincide con el ID que se especifique aquí, los intentos de generar el bucket fallarán. Para obtener información detallada, consulte [Verificación de la propiedad del bucket con la condición de propietario del bucket](#) en la Guía del usuario de Amazon S3.

**Note**

La configuración del propietario esperado del bucket se aplica solo a la ubicación de salida de Amazon S3 especificada para los resultados de las consultas de Athena. No se aplica a otras ubicaciones de Amazon S3, como las ubicaciones de origen de datos en buckets externos de Amazon S3, ubicaciones de tablas de destino de CTAS y INSERT INTO, ubicaciones de salida de instrucciones UNLOAD, operaciones para desbordar buckets para consultas federadas, o consultas SELECT ejecutadas en una tabla de otra cuenta.

- Una configuración de cifrado, si utiliza cifrado para todas las consultas de grupo de trabajo. Puede cifrar solo todas las consultas en un grupo de trabajo, no solo algunos de ellos. Es mejor crear grupos de trabajo independientes que contengan las consultas cifradas o sin cifrar.

Además, su grupo de trabajo puede [invalidar la configuración del cliente](#). Antes del lanzamiento de grupos de trabajo, puede especificar la ubicación de los resultados y las opciones de cifrado como parámetros en el controlador JDBC u ODBC, o en la pestaña Propiedades en la consola de Athena. Esta configuración también podría especificarse directamente a través de las operaciones de la API. Esta configuración se conoce como “configuración del cliente”. Con los grupos de trabajo, puede ajustar estas configuraciones a nivel de grupo de trabajo para controlar las opciones disponibles a nivel de cliente. La aplicación de las configuraciones a nivel de grupo de trabajo también evita que los usuarios tengan que ajustar sus configuraciones del cliente de forma individual. Si selecciona la opción Invalidar la configuración del cliente para el grupo de trabajo, las consultas utilizan la configuración del grupo de trabajo y hacen caso omiso a la configuración del cliente.

Si se selecciona Invalidación de configuración del cliente, se notifica al usuario en la consola de que su configuración ha cambiado. Si la configuración del grupo de trabajo se aplica de esta forma, los usuarios pueden omitir la configuración del cliente correspondiente. Luego, las consultas que se ejecutan en la consola utilizan la configuración del grupo de trabajo aunque se encuentre presente la configuración del cliente. Además, cuando las consultas del grupo de trabajo se ejecutan a través de la AWS CLI, las operaciones de la API o los controladores JDBC u ODBC, la configuración del cliente (como la ubicación y el cifrado de los resultados de las consultas) queda anulada por la configuración del grupo de trabajo. Para ver las configuraciones del grupo de trabajo, [consulte los detalles del grupo de trabajo](#).

También puede [establecer los límites de consulta](#) de las consultas en grupos de trabajo.

## La configuración del grupo de trabajo invalida la configuración del cliente

Los diálogos Crear grupo de trabajo y Editar grupo de trabajo tienen un campo titulado Invalidar la configuración del cliente. Este campo se deshabilita de forma predeterminada. Dependiendo de si lo selecciona, Athena hace lo siguiente:

- Si no se selecciona Invalidar la configuración del cliente, no se aplica la configuración del grupo de trabajo al nivel del cliente. Cuando la opción de invalidar la configuración del cliente no se encuentra seleccionada para el grupo de trabajo, Athena utiliza la configuración del cliente para todas las consultas que se ejecutan en el grupo de trabajo, incluida la configuración de la ubicación de los resultados de las consultas, el propietario esperado del bucket, el cifrado y el control de los objetos escritos en el bucket de resultados de las consultas. Cada usuario puede especificar su propia configuración en el menú de Configuración en la consola. Si no se establece la configuración del cliente, se aplica la configuración del grupo de trabajo. Si utiliza la AWS CLI, las acciones de la API o los controladores JDBC y ODBC para ejecutar consultas en un grupo de trabajo que no invalida la configuración del cliente, las consultas utilizarán la configuración que especifique en las consultas.
- Si se selecciona Invalidar la configuración del cliente, se aplica la configuración del grupo de trabajo al nivel del cliente. Cuando la opción de invalidar la configuración del cliente se encuentra seleccionada para el grupo de trabajo, Athena utiliza la configuración del grupo de trabajo para todas las consultas que se ejecutan en el grupo de trabajo, incluida la configuración de la ubicación de los resultados de las consultas, el propietario esperado del bucket, el cifrado y el control de los objetos escritos en el bucket de resultados de las consultas. La configuración del grupo de trabajo invalida cualquier configuración del cliente que especifique para una consulta cuando utilice la consola, las acciones de la API o los controladores JDBC u ODBC.

Si invalida la configuración del cliente, la próxima vez que usted o cualquier usuario del grupo de trabajo abra la consola de Athena, Athena le notificará que las consultas en el grupo de trabajo utilizan la configuración del grupo de trabajo y le preguntará si acepta este cambio.

### Important

Si utiliza las acciones de la API, la AWS CLI o los controladores JDBC y ODBC para ejecutar consultas en un grupo de trabajo y anular la configuración del cliente, asegúrese de omitir la configuración del cliente en las consultas o de actualizarla a fin de que coincida con la configuración del grupo de trabajo. Si especifica la configuración del cliente en las consultas, pero la ejecuta en un grupo de trabajo que anula la configuración, las consultas se ejecutarán, pero se utilizará la configuración del grupo de trabajo. Para



obtener información sobre la visualización de la configuración de un grupo de trabajo, consulte [Ver los detalles del grupo de trabajo](#).

## Administración de los grupos de trabajo

En el navegador <https://console.aws.amazon.com/athena/>, puede llevar a cabo las tareas siguientes:

Instrucción	Descripción
<a href="#">Creación de un grupo de trabajo</a>	Crear un grupo de trabajo nuevo.
<a href="#">Editar un grupo de trabajo</a>	Editar un grupo de trabajo y cambiar su configuración. No puede cambiar un nombre de grupo de trabajo, pero puede crear un nuevo grupo de trabajo con la misma configuración y un nombre diferente.
<a href="#">Ver los detalles del grupo de trabajo</a>	Vea los detalles del grupo de trabajo, por ejemplo, su nombre, descripción, límites de uso de datos, ubicación de los resultados de las consultas, propietario del bucket esperado de los resultados de las consultas, cifrado y control de los objetos escritos en el bucket de los resultados de las consultas. También puede verificar si este grupo de trabajo aplica su configuración, si se ha seleccionado Invalidar la configuración del cliente.
<a href="#">Eliminar un grupo de trabajo</a>	<p>Elimine un grupo de trabajo. Si elimina un grupo de trabajo, se eliminan el historial de consultas, las consultas guardadas, la configuración del grupo de trabajo y los controles de límites de datos por consulta. Los controles del límite de datos del grupo de trabajo permanecen en CloudWatch y puede eliminarlos de manera individual.</p> <p>No se puede eliminar el grupo de trabajo principal.</p>
<a href="#">Cambiar los grupos de trabajo</a>	Cambie entre los grupos de trabajo a los que tiene acceso.
<a href="#">Copiar una consulta guardada entre grupos de trabajo</a>	Copie una consulta guardada entre grupos de trabajo. Es posible que desee hacerlo si, por ejemplo, creó una consulta en un grupo de

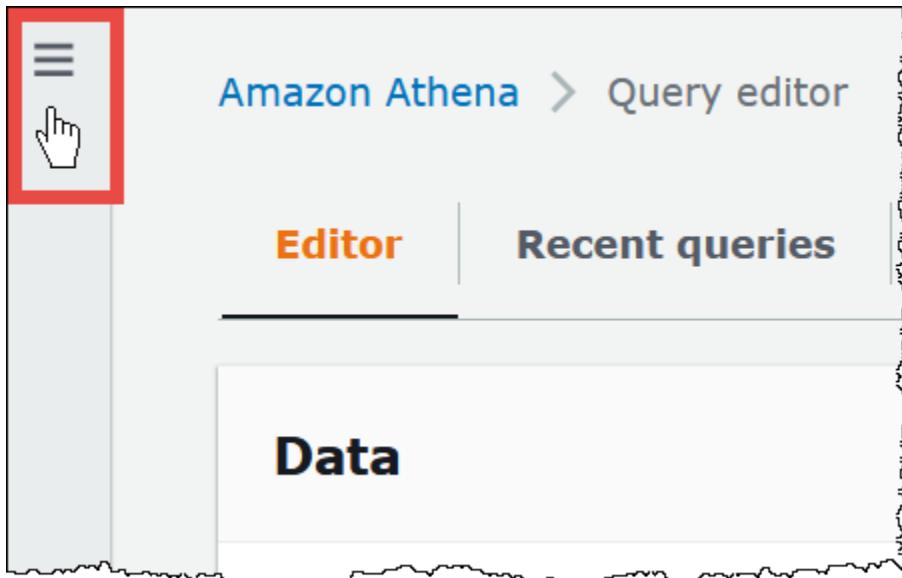
Instrucción	Descripción
	trabajo de vista previa y desea que esté disponible en un grupo de trabajo que no sea de vista previa.
<a href="#">Habilitar o deshabilitar un grupo de trabajo</a>	Habilite o deshabilite un grupo de trabajo. Cuando un grupo de trabajo está deshabilitado, los usuarios no pueden ejecutar consultas o crear consultas con nombre nuevas. Si tiene acceso a él, puede seguir viendo las métricas, los controles de límites de uso de datos, la configuración de los grupos de trabajo, el historial de consultas y las consultas guardadas.
<a href="#">Especificar un grupo de trabajo en el que se ejecutarán consultas</a>	Antes de ejecutar consultas, debe especificar a Athena qué grupo de trabajo se debe utilizar. Debe tener permisos en el grupo de trabajo.
<a href="#">Creación de un grupo de trabajo de Athena que utilice la autenticación del IAM Identity Center</a>	Para utilizar las identidades del IAM Identity Center con Athena, debe crear un grupo de trabajo habilitado para IAM Identity Center. Después de crear el grupo de trabajo, puede utilizar la consola o la API del IAM Identity Center para asignar usuarios o grupos del IAM Identity Center al grupo de trabajo.

## Creación de un grupo de trabajo

Para crear un grupo de trabajo se necesitan permisos para las acciones de la API `CreateWorkgroup`. Consulte [Acceso a grupos de trabajo y etiquetas](#) y [Políticas de IAM para acceder a los grupos de trabajo](#). Si está añadiendo etiquetas, también tiene que añadir permisos a `TagResource`. Consulte [Ejemplos de política de etiquetas para grupos de trabajo](#).

Para crear un grupo de trabajo en la consola


1. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.




2. En el panel de navegación de la consola de Athena, elija Grupos de trabajo.
3. En el panel Grupos de trabajo, elija Crear grupo de trabajo.
4. En la página Crear grupo de trabajo, rellene los campos tal y como se indica a continuación:

Campo	Descripción
Nombre del grupo de trabajo	Obligatorio. Escriba un nombre único para el grupo de trabajo. Utilice entre 1 y 128 caracteres. (A-Z,a-z,0-9,_,-,.). Este nombre no se puede cambiar.
Descripción	Opcional. Escriba una descripción para el grupo de trabajo. Puede contener un máximo de 1024 caracteres.
Elección del tipo de motor	<p>Elija Athena SQL si desea ejecutar consultas SQL ad hoc sobre los <a href="#">datos de Amazon S3</a> o utilizar un <a href="#">conector de origen de datos compilado previamente</a> para ejecutar <a href="#">consultas federadas</a> en una variedad de orígenes de datos externos a Amazon S3. Puede ejecutar consultas mediante el editor de consultas de Athena, <a href="#">AWS CLI</a> o las <a href="#">API de Athena</a>.</p> <p>Elija Apache Spark si desea crear, editar y ejecutar aplicaciones del cuaderno de Jupyter con Python y Apache Spark. Los cuadernos de Jupyter contienen una lista de celdas que pueden incluir código, texto, Markdown, cálculos, gráficos y contenido</p>

Campo	Descripción
	<p>multimedia enriquecido. Las celdas se ejecutan en orden como cálculos en una sesión de cuaderno interactiva en Athena. Para obtener información sobre cómo crear y configurar un grupo de trabajo habilitado para Spark, consulte <a href="#">Creación de un grupo de trabajo habilitado para Spark en Athena</a>.</p> <p>Tras crear un grupo de trabajo, se puede actualizar su motor de análisis (por ejemplo, de la versión 2 del motor de Athena a la versión 3 del motor de Athena), pero no se puede cambiar su tipo de motor. Por ejemplo, un grupo de trabajo de la versión 3 del motor de Athena no se puede cambiar a un grupo de trabajo de la versión 3 del motor de PySpark.</p>
Actualizar el motor de consultas	Elija cómo desea actualizar el grupo de trabajo cuando se publique una nueva versión del motor Athena. Puede dejar que Athena decida cuándo actualizar el grupo de trabajo o puede especificar manualmente una versión del motor. Para obtener más información, consulte <a href="#">Control de versiones del motor Athena</a> .
Modo de autenticación	Elija AWS Identity and Access Management (IAM) para utilizar la autenticación o la federación de IAM para el grupo de trabajo. Elija IAM Identity Center si desea admitir identidades de personal, como usuarios y grupos de proveedores de identidad de SAML 2.0, como Microsoft Active Directory. Para obtener más información, consulte <a href="#">Trusted identity propagation across applications</a> en la Guía del usuario del AWS IAM Identity Center.
Rol de servicio para el acceso al IAM Identity Center	Athena necesita permisos de IAM para acceder al IAM Identity Center en su nombre. Para obtener más información sobre los roles de servicio de IAM, consulte <a href="#">Creación de un rol para delegar permisos a un servicio de AWS</a> en la Guía del usuario de IAM.

Campo	Descripción
Ubicación del resultado de las consultas	<p>Opcional. Ingrese una ruta a un bucket o un prefijo de Amazon S3. El bucket y el prefijo deben existir para poder especificarlos.</p> <div data-bbox="548 352 1507 903" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>Si ejecuta consultas en la consola, especificar la ubicación de los resultados de la consulta es opcional. Si no lo especifica para el grupo de trabajo o en Configuración, Athena utiliza la ubicación predeterminada para los resultados de la consulta. Si ejecuta consulta con la API o los controladores, debe especificar la ubicación de los resultados de la consulta en al menos uno de los siguientes dos sitios: para consultas individuales con <a href="#">OutputLocation</a> o para el grupo de trabajo con <a href="#">WorkGroupConfiguration</a>.</p></div>

Campo	Descripción
Propietario esperado del bucket	<p>Opcional. Ingrese el ID de la Cuenta de AWS que espera que sea el propietario del bucket de ubicación de salida. Se trata de una medida de seguridad añadida. Si el ID de cuenta del propietario del bucket no coincide con el ID que se especifique aquí, los intentos de generar el bucket fallarán. Para obtener información detallada, consulte <a href="#">Verificación de la propiedad del bucket con la condición de propietario del bucket</a> en la Guía del usuario de Amazon S3.</p> <div data-bbox="548 590 1507 1192"><p> <b>Note</b></p><p>La configuración del propietario esperado del bucket se aplica solo a la ubicación de salida de Amazon S3 especificada para los resultados de las consultas de Athena. No se aplica a otras ubicaciones de Amazon S3, como las ubicaciones de origen de datos en buckets externos de Amazon S3, ubicaciones de tablas de destino de CTAS y INSERT INTO, ubicaciones de salida de instrucciones UNLOAD, operaciones para desbordar buckets para consultas federadas, o consultas SELECT ejecutadas en una tabla de otra cuenta.</p></div>

Campo	Descripción
<p>Asignar control total sobre los resultados de las consultas al propietario del bucket</p>	<p>Este campo se deshabilita de forma predeterminada. Si lo selecciona y <a href="#">están habilitadas las ACL</a> para el bucket de ubicación de resultados de las consultas, se concede un acceso de control total sobre los resultados de las consultas al propietario del bucket. Por ejemplo, si la ubicación del resultado de las consultas es propiedad de otra cuenta, puede utilizar esta opción para conceder la propiedad y el control total sobre los resultados de las consultas a la otra cuenta.</p> <p>Si la configuración de propiedad de objetos de S3 del bucket es Propietario del bucket preferido, el propietario del bucket también es propietario de todos los objetos de resultados de las consultas escritos desde este grupo de trabajo. Por ejemplo, si el grupo de trabajo de una cuenta externa habilita esta opción y establece la ubicación del resultado de la consulta en el bucket de Simple Storage Service (Amazon S3) de la cuenta que tiene Bucket owner preferred (Propietario del bucket preferido) como configuración de propiedad de objetos de S3, usted es propietario de los resultados de las consultas del grupo de trabajo externo y tiene control total sobre ellos.</p> <p>Seleccionar esta opción cuando la configuración de la propiedad de objetos S3 del bucket de resultados de las consultas es Propietario del bucket obligatorio no surte ningún efecto. Para obtener más información, consulte <a href="#">Configuración de la propiedad de objetos</a> en la Guía del usuario de Amazon S3.</p>
<p>Cifrar resultados de la consulta</p>	<p>Opcional. Cifrar resultados almacenados en Amazon S3. Si se selecciona, se cifran todas las consultas en el grupo de trabajo.</p> <p>Si se selecciona, puede seleccionar el Tipo de cifrado, la Clave de cifrado y escribir el ARN de la clave de KMS.</p> <p>Si no tiene la clave, abra la <a href="#">consola de AWS KMS</a> para crearla. Para obtener más información, consulte <a href="#">Creación de claves</a> en la Guía para desarrolladores de AWS Key Management Service.</p>


Campo	Descripción
Establecer <b><i>encryption_type</i></b> como cifrado mínimo	<p>Opcional. Seleccione esta opción para aplicar un tipo mínimo de cifrado en los resultados de las consultas a todos los usuarios del grupo de trabajo. Al seleccionar esta opción, se muestra una tabla con la jerarquía de los tipos de cifrado. En la tabla también se muestran los tipos de cifrado que los usuarios del grupo de trabajo podrán utilizar cuando especifique un tipo de cifrado en particular como mínimo. Para utilizar esta opción, no se debe seleccionar Invalidar la configuración del cliente.</p> <p>Para obtener más información, consulte <a href="#">Configuración del cifrado mínimo para un grupo de trabajo</a>.</p>
Habilitación de permisos de acceso a S3	<p>Este campo se selecciona de forma predeterminada al elegir IAM Identity Center como modo de autenticación. Si se selecciona, con esta opción se aplican los permisos basados en usuarios o grupos del IAM Identity Center en las ubicaciones de Amazon S3.</p>
Creación de un prefijo S3 basado en la identidad del usuario	<p>Cuando se selecciona esta opción, Athena crea un prefijo de Amazon S3 al almacenar los resultados de las consultas. El prefijo se basa en la identidad del usuario del IAM Identity Center.</p>
Publicación de métricas de consultas en CloudWatch	<p>Este campo se selecciona de forma predeterminada. Publicar métricas de consultas en CloudWatch. Consulte <a href="#">Supervisión de las consultas de Athena con métricas de CloudWatch</a>.</p>
Invalidación de la configuración del cliente	<p>Este campo se deshabilita de forma predeterminada. Si se selecciona, la configuración del grupo de trabajo se aplica a todas las consultas en el grupo de trabajo y se invalida la configuración del lado del cliente. Para obtener más información, consulte <a href="#">La configuración del grupo de trabajo invalida la configuración del cliente</a>.</p>



Campo	Descripción
Buckets de S3 de pago por solicitante	Opcional. Elija Habilitar consultas en buckets de pago por solicitante en Amazon S3 si los usuarios del grupo de trabajo van a ejecutar consultas en datos almacenados en buckets de Amazon S3 que están configurados como pago por solicitante. A la cuenta del usuario que ejecuta la consulta se le cargan las tarifas de acceso a los datos aplicables y de transferencia de datos asociados a la consulta. Para obtener más información, consulte <a href="#">Buckets de pago por solicitante</a> en la Guía del usuario de Amazon Simple Storage Service.
Administración del control del uso de datos por consulta	Opcional. Establece el límite de la cantidad máxima de datos puede analizar una consulta. Puede definir solo un límite por consulta para un grupo de trabajo. El límite se aplica a todas las consultas del grupo de trabajo y si la consulta supera el límite, se cancelará. Para obtener más información, consulte <a href="#">Configuración de los límites de control del uso de datos</a> .
Alertas del uso de datos de grupos de trabajo	Opcional. Establezca varios umbrales de alerta cuando las consultas que se ejecutan en este grupo de trabajo analizan una cantidad específica de datos en un periodo específico. Las alertas se implementan mediante las alarmas de Amazon CloudWatch y se aplican a todas las consultas del grupo de trabajo. Para obtener más información, consulte <a href="#">Uso de las alarmas de Amazon CloudWatch</a> en la Guía del usuario de Amazon CloudWatch.
Etiquetas	Opcional. Añada una o más etiquetas a un grupo de trabajo. Una etiqueta es una marca que se asigna a un recurso de grupo de trabajo de Athena. Consta de una clave y un valor. Utilice las <a href="#">prácticas recomendadas para etiquetado</a> de AWS para crear un conjunto coherente de etiquetas y categorizar los grupos de trabajo según su finalidad, propietario o entorno. También puede usar etiquetas en las políticas de IAM y para controlar los costos de facturación. No utilice claves de etiquetas duplicadas en el mismo grupo de trabajo. Para obtener más información, consulte <a href="#">the section called “Etiquetado de recursos”</a> .

5. Elija Crear grupo de trabajo. El grupo de trabajo aparece en la lista en la página Grupos de trabajo.

También puede utilizar la operación de la API [CreateWorkGroup](#) para crear un grupo de trabajo.

 Important

Después de crear grupos de trabajo, cree IAM para [Políticas de IAM para acceder a los grupos de trabajo](#) que le permiten ejecutar acciones relacionadas con el grupo de trabajo.

### Editar un grupo de trabajo

Para editar un grupo de trabajo se necesitan permisos para las operaciones de la API UpdateWorkgroup. Consulte [Acceso a grupos de trabajo y etiquetas](#) y [Políticas de IAM para acceder a los grupos de trabajo](#). Si está añadiendo o editando etiquetas, también tiene que tener permisos para TagResource. Consulte [Ejemplos de política de etiquetas para grupos de trabajo](#).

Para editar un grupo de trabajo en la consola

1. En el panel de navegación de la consola de Athena, elija Grupos de trabajo.
2. En la página Grupos de trabajo, seleccione el botón del grupo de trabajo que quiere editar.
3. Seleccione Acciones, Editar.
4. Cambie los campos según sea necesario. Para la lista de campos, consulte [Crear grupo de trabajo](#). Puede cambiar todos los campos, excepto el nombre del grupo de trabajo. Si necesita cambiar el nombre, cree otro grupo de trabajo con el nuevo nombre y la misma configuración.
5. Elija Guardar cambios. El grupo de trabajo actualizado aparece en la lista de la página Grupos de trabajo.

### Ver los detalles del grupo de trabajo

Para cada grupo de trabajo, puede ver sus detalles. Los detalles incluyen el nombre del grupo de trabajo, la descripción, si está habilitado o desactivado, y la configuración que se utiliza para las consultas que se ejecutan en el grupo de trabajo, que incluyen la ubicación de los resultados de las consultas, el propietario del bucket esperado, el cifrado y el control de objetos escritos en el bucket de resultados de las consultas. Si un grupo de trabajo tiene límites de uso de datos, también se muestran.

## Para ver los detalles del grupo de trabajo

1. En el panel de navegación de la consola de Athena, elija Grupos de trabajo.
2. En la página Grupos de trabajo, elija el enlace del grupo de trabajo que quiere ver. La página Información general sobre los detalles muestra la página del grupo de trabajo.

## Eliminar un grupo de trabajo

Puede eliminar un grupo de trabajo si tiene permiso para hacerlo. No se puede eliminar el grupo de trabajo principal.

Si tiene los permisos adecuados, puede eliminar un grupo de trabajo en cualquier momento. También puede eliminar un grupo de trabajo que contiene consultas guardadas. En este caso, antes de proceder a la eliminación de un grupo de trabajo, Athena le avisa de que se eliminarán las consultas guardadas.

Si elimina un grupo de trabajo mientras se encuentra en él, la consola cambia el foco al grupo de trabajo principal. Si tiene acceso a él, puede ejecutar consultas y ver su configuración.

Si elimina un grupo de trabajo, se eliminan su configuración y los controles del límite de datos por consulta. Los controles del límite de datos del grupo de trabajo permanecen en CloudWatch y puede eliminarlos allí, si es necesario.

### Important

Antes de eliminar un grupo de trabajo, asegúrese de que sus usuarios también pertenecen a otros grupos de trabajo donde puedan seguir ejecutando consultas. Si las políticas de IAM de los usuarios les permitían ejecutar consultas solo en este grupo de trabajo y lo elimina, ya no tienen permisos para ejecutar consultas. Para obtener más información, consulte [Example policy for running queries in the primary workgroup](#).

## Para eliminar un grupo de trabajo en la consola

1. En el panel de navegación de la consola de Athena, elija Grupos de trabajo.
2. En la página Grupos de trabajo, seleccione el botón del grupo de trabajo que quiere eliminar.
3. Elija Acciones, Eliminar.

4. En la solicitud de confirmación Eliminar grupo de trabajo ingrese el nombre del grupo de trabajo y, a continuación, elija Eliminar.

Para eliminar un grupo de trabajo con la operación de la API, utilice la acción `DeleteWorkGroup`.

### Cambiar los grupos de trabajo

Puede cambiar de un grupo de trabajo a otro si tiene permisos para ambos.

Puede abrir hasta diez pestañas de consulta dentro de cada grupo de trabajo. Al cambiar entre grupos de trabajo, la pestaña de la consulta permanece abierta para hasta tres grupos de trabajo.

### Para cambiar grupos de trabajo

1. En la consola de Athena, utilice la opción Grupo de trabajo en la parte superior derecha para elegir uno.
2. Si aparece el cuadro de diálogo de configuración del grupo de trabajo *workgroup-name*, elija Confirmación.

La opción Grupo de trabajo muestra el nombre del grupo de trabajo al que se ha cambiado. Ahora ya puede ejecutar consultas en este grupo de trabajo.

### Copiar una consulta guardada entre grupos de trabajo

Actualmente, la consola de Athena no tiene la opción de copiar una consulta guardada de un grupo de trabajo a otro de forma directa, pero puede realizar la misma tarea manualmente mediante el siguiente procedimiento.

### Para copiar una consulta guardada entre grupos de trabajo

1. En la consola de Athena, en el grupo de trabajo desde el que desea copiar la consulta, elija la pestaña Consultas guardadas.
2. Elija el enlace de la consulta guardada que quiere copiar. Athena abre la consulta en el editor de consultas.
3. En el editor de consultas, seleccione el texto de la consulta y, a continuación, presione **Ctrl+C** para copiarlo.
4. [Cambie](#) al grupo de trabajo de destino o [Cree un grupo de trabajo](#) y, a continuación, cambie a él.

5. Abra una nueva pestaña en el editor de consultas y, luego, presione **Ctrl+V** para pegar el texto en la pestaña nueva.
6. En el editor de consultas, elija Guardar como para guardar la consulta en el grupo de trabajo de destino.
7. En el cuadro de diálogo Elegir un nombre, ingrese un nombre para la consulta y una descripción opcional.
8. Seleccione Guardar.

### Habilitar o deshabilitar un grupo de trabajo

Si tiene permisos para hacerlo, puede habilitar o deshabilitar grupos de trabajo en la consola, mediante las operaciones de la API o con los controladores JDBC y ODBC.

#### Para habilitar o deshabilitar un grupo de trabajo

1. En el panel de navegación de la consola de Athena, elija Grupos de trabajo.
2. En la página Grupos de trabajo, elija el enlace del grupo de trabajo.
3. En la esquina superior derecha, elija Habilitar grupo de trabajo o Desactivar grupo de trabajo.
4. En la solicitud de confirmación, elija Habilitar o Desactivar. Si deshabilita un grupo de trabajo, los usuarios no pueden ejecutar consultas en él ni consultas con nombre nuevas. Si habilita un grupo de trabajo, los usuarios pueden utilizarlo para ejecutar consultas.

### Especificar un grupo de trabajo en el que se ejecutarán consultas

Para especificar un grupo de trabajo que se va a utilizar, debe tener permisos del grupo de trabajo.

#### Para especificar el grupo de trabajo que se va a utilizar

1. Asegúrese de que sus permisos le permiten ejecutar consultas en un grupo de trabajo que pretende utilizar. Para obtener más información, consulte [the section called “Políticas de IAM para acceder a los grupos de trabajo”](#).
2. Para especificar el grupo de trabajo, utilice una de estas opciones:
  - Si va a usar la consola de Athena, establezca el grupo de trabajo [mediante el cambio de grupos de trabajo](#).

- Si utiliza las operaciones de la API de Athena, especifique el nombre de grupo de trabajo de la acción de la API. Por ejemplo, puede definir el nombre de grupo de trabajo en [StartQueryExecution](#), tal y como se indica a continuación:

```
StartQueryExecutionRequest startQueryExecutionRequest = new
    StartQueryExecutionRequest()
        .withQueryString(ExampleConstants.ATHENA_SAMPLE_QUERY)
        .withQueryExecutionContext(queryExecutionContext)
        .withWorkGroup(WorkgroupName)
```

- Si utiliza el controlador JDBC u ODBC, establezca el nombre del grupo de trabajo en la cadena de conexión mediante el parámetro de configuración `Workgroup`. El controlador pasa el nombre del grupo de trabajo a Athena. Especifique el parámetro de grupo de trabajo en la cadena de conexión tal y como se muestra en el siguiente ejemplo:

```
jdbc:awsathena://AwsRegion=<AWSREGION>;UID=<ACCESSKEY>;
PWD=<SECRETKEY>;S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/<athena-
output>-<AWSREGION>;
Workgroup=<WORKGROUPNAME>;
```

## Configuración del cifrado mínimo para un grupo de trabajo

Como administrador de un grupo de trabajo de Athena SQL, puede aplicar un nivel mínimo de cifrado en Amazon S3 para todos los resultados de las consultas del grupo de trabajo. Puede utilizar esta característica para asegurarse de que los resultados de las consultas nunca se almacenen en un bucket de Amazon S3 sin cifrar.

Cuando los usuarios de un grupo de trabajo con el cifrado mínimo habilitado envían una consulta, solo pueden establecer el cifrado en el nivel mínimo que configure o en un nivel superior si hay alguno disponible. Athena cifra los resultados de la consulta en el nivel especificado cuando el usuario ejecuta la consulta o en el nivel establecido en el grupo de trabajo.

Se encuentran disponibles los siguientes niveles:

- Básico: cifrado del servidor de Amazon S3 con claves administradas por Amazon S3 (SSE\_S3).
- Intermedio: cifrado del servidor con claves administradas por KMS (SSE\_KMS).
- Avanzado: cifrado del cliente con claves administradas por KMS (CSE\_KMS).

## Consideraciones y limitaciones

- La característica de cifrado mínimo no se encuentra disponible para los grupos de trabajo habilitados para Apache Spark.
- La característica de cifrado mínimo solo funciona cuando el grupo de trabajo no habilita la opción [Invalidar la configuración del cliente](#).
- Si el grupo de trabajo tiene habilitada la opción Invalidar la configuración del cliente, prevalece la configuración de cifrado del grupo de trabajo y la configuración de cifrado mínimo no tiene efecto.
- La habilitación de esta característica no tiene ningún costo.

## Habilitación del cifrado mínimo para un grupo de trabajo

Puede habilitar un nivel de cifrado mínimo para los resultados de las consultas de su grupo de trabajo de Athena SQL al crear o actualizar el grupo de trabajo. Para ello, puede utilizar la consola, la API o la AWS CLI de Athena.

### Uso de la consola de Athena para habilitar el cifrado mínimo

Para comenzar a crear o editar un grupo de trabajo mediante la consola de Athena, consulte [Crear un grupo de trabajo](#) o [Editar un grupo de trabajo](#). Al configurar el grupo de trabajo, proceda con los siguientes pasos para habilitar el cifrado mínimo.

A fin de configurar el nivel de cifrado mínimo para los resultados de las consultas del grupo de trabajo

1. En la sección de Configuraciones adicionales, amplíe Configuración.
2. Desactive la opción Invalidar la configuración del cliente o compruebe que no se encuentre seleccionada.
3. En la sección de Configuraciones adicionales, amplíe Configuración de los resultados de la consulta.
4. Seleccione la opción Cifrado de los resultados de la consulta.
5. En Tipo de cifrado, seleccione el método de cifrado que desee que Athena utilice para los resultados de las consultas de su grupo de trabajo (SSE\_S3, SSE\_KMS o CSE\_KMS). Estos tipos de cifrado corresponden a los niveles de seguridad básico, intermedio y avanzado.
6. A fin de aplicar el método de cifrado que eligió como nivel mínimo de cifrado para todos los usuarios, seleccione Establecer ***encryption\_method*** como cifrado mínimo.

Al seleccionar esta opción, en una tabla se muestra la jerarquía de cifrado y los niveles de cifrado que se permitirán a los usuarios cuando el tipo de cifrado que elija pase a ser el mínimo.

7. Después de crear el grupo de trabajo o actualizar la configuración del grupo de trabajo, elija Crear grupo de trabajo o Guardar cambios.

### Uso de la API o la AWS CLI de Athena para habilitar el cifrado mínimo

Cuando utilice la API [CreateWorkGroup](#) o [UpdateWorkGroup](#) para crear o actualizar un grupo de trabajo de Athena SQL, establezca [EnforceWorkGroupConfiguration](#) en `false`, [EnableMinimumEncryptionConfiguration](#) en `true`, y utilice [EncryptionOption](#) a fin de especificar el tipo de cifrado.

En la AWS CLI, utilice el comando [create-work-group](#) o [update-work-group](#) con los parámetros `--configuration` o `--configuration-updates` y especifique las opciones correspondientes a las de la API.

### Uso de grupos de trabajo de Athena habilitados para IAM Identity Center

La característica de propagación de identidad de confianza del AWS IAM Identity Center permiten utilizar las identidades de sus empleados en los servicios de análisis de AWS. La propagación de identidad de confianza le ahorra tener que realizar configuraciones de proveedores de identidad específicos del servicio o configuraciones de rol de IAM.

Con IAM Identity Center, puede administrar la seguridad de inicio de sesión de las identidades de sus empleados, también conocidos como usuarios de los empleados. El IAM Identity Center ofrece un lugar en el que puede crear o conectar a los usuarios de los empleados y administrar de forma centralizada su acceso a todas sus cuentas y aplicaciones de AWS. Puede utilizar permisos de varias cuentas para asignar a estos usuarios acceso a las Cuentas de AWS. Puede utilizar las asignaciones de aplicaciones para asignar a sus usuarios el acceso a las aplicaciones habilitadas para el IAM Identity Center, a las aplicaciones en la nube y a las aplicaciones del lenguaje de marcado para confirmaciones de seguridad (SAML 2.0) del cliente. Para obtener más información, consulte [Propagación de identidad de confianza en aplicaciones](#) en la Guía del usuario del AWS IAM Identity Center.

Actualmente, el soporte de Athena SQL para la propagación de identidades de confianza le permite usar la misma identidad para Amazon EMR Studio y la interfaz de Athena SQL en EMR Studio. Para utilizar las identidades del IAM Identity Center con SQL de Athena en EMR Studio, debe crear grupos de trabajo habilitados para IAM Identity Center en Athena. Luego puede utilizar la consola



del IAM Identity Center o la API para asignar usuarios o grupos del IAM Identity Center a los grupos de trabajo de Athena habilitados para IAM Identity Center. Las consultas de un grupo de trabajo de Athena que utilice la propagación de identidades de confianza deben ejecutarse desde la interfaz SQL de Athena en un EMR Studio que tenga habilitado el IAM Identity Center.

## Consideraciones y limitaciones

Cuando utilice la propagación de identidades de confianza con Amazon Athena, tenga en cuenta los siguientes puntos:

- No puede cambiar el método de autenticación del grupo de trabajo una vez creado el grupo de trabajo.
  - No se pueden modificar los grupos de trabajo existentes de Athena SQL para admitir grupos de trabajo habilitados para IAM Identity Center.
  - No se pueden modificar los grupos de trabajo habilitados para IAM Identity Center para admitir permisos de IAM de nivel de recursos o políticas de IAM basadas en identidades.
- Para acceder a los grupos de trabajo habilitados para la propagación de identidad de confianza, los usuarios del IAM Identity Center deben estar asignados al `IdentityCenterApplicationArn` que se devuelve en la respuesta de la acción de la API [GetWorkGroup](#) de Athena.
- Los permisos de acceso de Amazon S3 deben configurarse para utilizar identidades de propagación de identidad de confianza. Para obtener más información, consulte [S3 Access Grants and corporate directory identities](#) en la Guía del usuario de Amazon S3.
- Los grupos de trabajo de Athena habilitados para IAM Identity Center requieren que Lake Formation esté configurado para usar las identidades del IAM Identity Center. Para obtener más información sobre la configuración, consulte [Integración del IAM Identity Center](#) en la Guía para desarrolladores de AWS Lake Formation.
- De forma predeterminada, las consultas caducan después de 30 minutos en los grupos de trabajo que utilizan una propagación de identidad de confianza. Puede solicitar un aumento del tiempo de espera de una consulta, pero el tiempo máximo que puede ejecutar una consulta en grupos de trabajo de propagación de identidades de confianza es de una hora.
- Los cambios en los derechos de los usuarios o grupos en los grupos de trabajo de propagación de identidades de confianza pueden tardar hasta una hora en aplicarse.
- Las consultas de un grupo de trabajo de Athena que utiliza la propagación de identidad de confianza no se pueden ejecutar directamente desde la consola de Athena. Deben ejecutarse desde la interfaz de Athena en un EMR Studio que tenga habilitado el IAM Identity Center. Para

obtener más información sobre el uso de Athena en EMR Studio, consulte [Uso del editor de SQL de Amazon Athena en EMR Studio](#) en la Guía de administración de Amazon EMR.

- La propagación de identidades de confianza no es compatible con las siguientes características de Athena.
  - Claves de contexto `aws:CalledVia`.
  - Athena para grupos de trabajo de Spark.
  - Acceso federado a la API de Athena.
  - Acceso federado a Athena mediante Lake Formation y los controladores JDBC y ODBC de Athena.
- Puede utilizar la propagación de identidad de confianza con Athena solo en las siguientes Regiones de AWS:
  - `us-east-2`: Este de EE. UU. (Ohio)
  - `us-east-1`: Este de EE. UU. (Norte de Virginia)
  - `us-west-1`: Oeste de EE. UU. (Norte de California)
  - `us-west-2`: Oeste de EE. UU. (Oregón)
  - `af-south-1`: África (Ciudad del Cabo)
  - `ap-east-1`: Asia-Pacífico (Hong Kong)
  - `ap-southeast-3`: Asia-Pacífico (Yakarta)
  - `ap-south-1`: Asia-Pacífico (Bombay)
  - `ap-northeast-3`: Asia-Pacífico (Osaka)
  - `ap-northeast-2`: Asia-Pacífico (Seúl)
  - `ap-southeast-1`: Asia Pacífico (Singapur)
  - `ap-southeast-2`: Asia Pacífico (Sídney)
  - `ap-northeast-1`: Asia Pacífico (Tokio)
  - `ca-central-1`: Canadá (centro)
  - `eu-central-1`: Europa (Fráncfort)
  - `eu-west-1`: Europa (Irlanda)
  - `eu-west-2`: Europa (Londres)
  - `eu-south-1`: Europa (Milán)
  - ~~`eu-west-3`: Europa (París)~~
  - `eu-north-1`: Europa (Estocolmo)

- `me-south-1`: Medio Oriente (Baréin)
- `sa-east-1`: América del Sur (São Paulo)

## Permisos necesarios

El usuario de IAM del administrador que crea el grupo de trabajo habilitado para IAM Identity Center en la consola de Athena debe tener adjuntas las siguientes políticas.

- La política administrada de `AmazonAthenaFullAccess`. Para obtener más información, consulte [Política administrada de AWS: AmazonAthenaFullAccess](#).
- La siguiente política insertada que permite las acciones de IAM y del IAM Identity Center:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:createRole",
        "iam:CreatePolicy",
        "iam:AttachRolePolicy",
        "iam:ListRoles",
        "iam:PassRole",
        "identitystore:ListUsers",
        "identitystore:ListGroups",
        "identitystore:CreateUser",
        "identitystore:CreateGroup",
        "sso:ListInstances",
        "sso:CreateInstance",
        "sso>DeleteInstance",
        "sso:DescribeUser",
        "sso:DescribeGroup",
        "sso:ListTrustedTokenIssuers",
        "sso:DescribeTrustedTokenIssuer",
        "sso:ListApplicationAssignments",
        "sso:DescribeRegisteredRegions",
        "sso:GetManagedApplicationInstance",
        "sso:GetSharedSsoConfiguration",
        "sso:PutApplicationAssignmentConfiguration",
        "sso:CreateApplication",
        "sso>DeleteApplication",
        "sso:PutApplicationGrant",
```

```

        "sso:PutApplicationAuthenticationMethod",
        "sso:PutApplicationAccessScope",
        "sso:ListDirectoryAssociations",
        "sso:CreateApplicationAssignment",
        "sso>DeleteApplicationAssignment",
        "organizations:ListDelegatedAdministrators",
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization",
        "organizations:CreateOrganization",
        "sso-directory:SearchUsers",
        "sso-directory:SearchGroups",
        "sso-directory:CreateUser"
    ],
    "Effect": "Allow",
    "Resource": [
        "*"
    ]
}
]
}

```

## Creación de un grupo de trabajo de Athena habilitado para IAM Identity Center

El siguiente procedimiento muestra los pasos y las opciones relacionadas con la creación de un grupo de trabajo de Athena habilitado para IAM Identity Center. Para obtener una descripción de las demás opciones de configuración disponibles para los grupos de trabajo de Athena, consulte [Creación de un grupo de trabajo](#).

Cómo crear un grupo de trabajo habilitado para SSO en la consola de Athena

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. En el panel de navegación de la consola de Athena, elija Grupos de trabajo.
3. En el panel Grupos de trabajo, elija Crear grupo de trabajo.
4. En la página Crear grupo de trabajo, para Nombre del grupo de trabajo, ingrese un nombre para el grupo de trabajo.
5. En Motor de análisis, utilice el Athena SQL predeterminado.
6. En Autenticación, elija IAM Identity Center.
7. En Rol de servicio para el acceso al IAM Identity Center, elija un rol de servicio existente o cree uno nuevo.

Athena requiere permisos para acceder al IAM Identity Center en su nombre. Para ello, Athena necesita un rol de servicio. Un rol de servicio es un rol de IAM que usted administra y que autoriza a un servicio de AWS a acceder a otros servicios de AWS en su nombre. Para obtener más información, consulte [Creación de un rol para delegar permisos a un servicio de AWS](#) en la Guía del usuario de IAM.

8. Expanda Configuración de resultados de la consulta y, a continuación, introduzca o elija una ruta de Amazon S3 en Ubicación del resultado de las consultas.
9. (Opcional) Elija Cifrar resultados de la consulta.
10. (Opcional) Elija Crear un prefijo S3 basado en la identidad del usuario.

Al crear un grupo de trabajo habilitado para IAM Identity Center, la opción Habilitar los permisos de acceso a S3 aparece seleccionada de forma predeterminada. Puede utilizar los permisos de acceso de Amazon S3 para controlar el acceso a las ubicaciones (prefijos) de los resultados de las consultas de Athena en Amazon S3. Para obtener más información sobre los permisos de acceso de Amazon S3, consulte [Administración del acceso con los permisos de acceso de Amazon S3](#).

En grupos de trabajo de Athena que utilizan la autenticación de IAM Identity Center, puede habilitar la creación de ubicaciones de resultados de consultas basadas en la identidad que se rigen por los permisos de acceso de Amazon S3. Estos prefijos de Amazon S3 basados en la identidad del usuario permiten que los usuarios en un grupo de trabajo de Athena mantengan sus resultados de consulta aislados de otros usuarios en el mismo grupo de trabajo.

Al activar la opción de prefijo de usuario, Athena añade el ID de usuario como prefijo de ruta de Amazon S3 a la ubicación de salida del resultado de la consulta para el grupo de trabajo (por ejemplo, `s3://DOC-EXAMPLE-BUCKET/${user_id}`). Para utilizar esta característica, debe configurar los permisos de acceso para permitir que solo el usuario acceda a la ubicación que tiene el prefijo de `user_id`. Para ver un ejemplo de política de rol de ubicación de permisos de acceso de Amazon S3 que restringe el acceso a los resultados de las consultas de Athena, consulte [Ejemplo de política de rol](#).

#### Note

Al seleccionar la opción de prefijo S3 de identidad de usuario, se habilita automáticamente la opción de anular la configuración del cliente para el grupo de

trabajo, tal y como se describe en el siguiente paso. La opción de anular la configuración del cliente es un requisito para la característica de prefijo de identidad del usuario.

11. Expanda Configuración y, a continuación, confirme que esté seleccionada la opción Invalidar la configuración del lado del cliente.

Cuando selecciona Invalidar la configuración del cliente, se aplica la configuración del grupo de trabajo al nivel del grupo de trabajo para todos los clientes. Para obtener más información, consulte [La configuración del grupo de trabajo invalida la configuración del cliente](#).

12. (Opcional) Realice cualquier otro ajuste de configuración que necesite tal y como se describe en [Creación de un grupo de trabajo](#).
13. Elija Crear grupo de trabajo.
14. Utilice la sección Grupos de trabajo de la consola de Athena para asignar usuarios o grupos del directorio del IAM Identity Center a su grupo de trabajo de Athena habilitado para el IAM Identity Center.

### Ejemplo de política de rol

El siguiente ejemplo muestra una política para que un rol se asocie a una ubicación de permiso de acceso de Amazon S3 que restringe el acceso a los resultados de consulta de Athena.

```
{
  "Statement": [{
    "Action": ["s3:*"],
    "Condition": {
      "ArnNotEquals": {
        "s3:AccessGrantsInstanceArn": "arn:aws:s3:${region}:${account}:access-grants/default"
      },
      "StringNotEquals": {
        "aws:ResourceAccount": "${account}"
      }
    },
    "Effect": "Deny",
    "Resource": "*",
    "Sid": "ExplicitDenyS3"
  }, {
    "Action": ["kms:*"],
    "Effect": "Deny",
    "NotResource": "arn:aws:kms:${region}:${account}:key/${keyid}",
```

```

    "Sid": "ExplicitDenyKMS"
  }, {
    "Action": ["s3:ListMultipartUploadParts", "s3:GetObject"],
    "Condition": {
      "ArnEquals": {
        "s3:AccessGrantsInstanceArn": "arn:aws:s3:${region}:${account}:access-
grants/default"
      },
      "StringEquals": {
        "aws:ResourceAccount": "${account}"
      }
    },
    "Effect": "Allow",
    "Resource": "arn:aws:s3:::ATHENA-QUERY-RESULT-LOCATION/${identitystore:UserId}/
*",
    "Sid": "ObjectLevelReadPermissions"
  }, {
    "Action": ["s3:PutObject", "s3:AbortMultipartUpload"],
    "Condition": {
      "ArnEquals": {
        "s3:AccessGrantsInstanceArn": "arn:aws:s3:${region}:${account}:access-
grants/default"
      },
      "StringEquals": {
        "aws:ResourceAccount": "${account}"
      }
    },
    "Effect": "Allow",
    "Resource": "arn:aws:s3:::ATHENA-QUERY-RESULT-LOCATION/${identitystore:UserId}/
*",
    "Sid": "ObjectLevelWritePermissions"
  }, {
    "Action": "s3:ListBucket",
    "Condition": {
      "ArnEquals": {
        "s3:AccessGrantsInstanceArn": "arn:aws:s3:${region}:${account}:access-
grants/default"
      },
      "StringEquals": {
        "aws:ResourceAccount": "${account}"
      },
      "StringLikeIfExists": {
        "s3:prefix": [ "${identitystore:UserId}", "${identitystore:UserId}/*" ]
      }
    }
  }

```

```
    },
    "Effect": "Allow",
    "Resource": "arn:aws:s3:::ATHENA-QUERY-RESULT-LOCATION",
    "Sid": "BucketLevelReadPermissions"
  }, {
    "Action": ["kms:GenerateDataKey", "kms:Decrypt"],
    "Effect": "Allow",
    "Resource": "arn:aws:kms:${region}:${account}:key/${keyid}",
    "Sid": "KMSPermissions"
  }],
  "Version": "2012-10-17"
}
```

## API de grupos de trabajo de Athena

A continuación, se indican algunas de las operaciones de la API de REST que se utilizan para grupos de trabajo de Athena. En todas las siguientes operaciones, excepto `ListWorkGroups`, debe especificar un grupo de trabajo. En otras operaciones, como, por ejemplo `StartQueryExecution`, el parámetro de grupo de trabajo es opcional y las operaciones no se incluyen aquí. Para obtener la lista completa de operaciones, consulte la [Referencia de la API de Amazon Athena](#).

- [CreateWorkGroup](#)
- [DeleteWorkGroup](#)
- [GetWorkGroup](#)
- [ListWorkGroups](#)
- [UpdateWorkGroup](#)

## Solución de problemas de grupos de trabajo

Utilice los siguientes consejos para solucionar problemas de grupos de trabajo.

- Compruebe los permisos de los usuarios individuales en su cuenta. Deben tener acceso a la ubicación para los resultados de las consultas, así como al grupo de trabajo en el que desean ejecutar consultas. Si desean cambiar de grupo de trabajo, también necesitan permisos para ambos grupos de trabajo. Para obtener más información, consulte [Políticas de IAM para acceder a los grupos de trabajo](#).
- Preste atención al contexto en la consola de Athena para ver en qué grupo de trabajo va a ejecutar consultas. Si utiliza el controlador, asegúrese de establecer el grupo de trabajo que necesita. Para



obtener más información, consulte [the section called “Especificar un grupo de trabajo en el que se ejecutarán consultas”](#).

- Si utiliza la API o los controladores para ejecutar consultas, debe especificar la ubicación de los resultados de la consulta de una de las siguientes dos formas: para consultas individuales, utilice [OutputLocation](#) (del cliente). En el grupo de trabajo, utilice [WorkGroupConfiguration](#). Si no se especifica la ubicación de alguna de las dos maneras, Athena genera un error de tiempo de ejecución de consulta.
- Si invalida la configuración del lado del cliente con la configuración de grupo de trabajo, pueden producirse errores relacionados con la ubicación de resultados de la consulta. Por ejemplo, un usuario del grupo de trabajo podría no tener permisos en la ubicación del grupo de trabajo en Amazon S3 para almacenar los resultados de las consultas. En este caso, añada los permisos necesarios.
- Los grupos de trabajo introducen cambios en el comportamiento de las operaciones de la API. Las llamadas a las siguientes operaciones de la API requieren que los usuarios de su cuenta tengan permisos basados en recursos en IAM a los grupos de trabajo donde realizan dichas operaciones. Si no existen permisos para el grupo de trabajo y para las acciones del grupo de trabajo, las siguientes acciones de la API arrojan `AccessDeniedException`: `CreateNamedQuery`, `DeleteNamedQuery`, `GetNamedQuery`, `ListNamedQueries`, `StartQueryExecution`, `StopQueryExecution`, `ListQueryExecutions`, `GetQueryExecution`, `GetQueryResults` y `GetQueryResultsStream` (esta acción de la API solo está disponible para su uso con el controlador y no está expuesta de otra manera para uso público). Para obtener más información, consulte [Acciones, recursos y claves de condición de Amazon Athena](#) en la Referencia de autorizaciones de servicio.

Las llamadas a las operaciones de la API `BatchGetQueryExecution` y `BatchGetNamedQuery` devuelven información solo para aquellas consultas que se ejecutan en grupos de trabajo a los que los usuarios tienen acceso. Si el usuario no tiene acceso al grupo de trabajo, estas operaciones de la API devuelven los ID de las consultas no autorizadas como parte de la lista de ID sin procesar. Para obtener más información, consulte [the section called “API de grupos de trabajo de Athena”](#).

- Si el grupo de trabajo en el que se ejecuta una consulta se ha configurado con una [ubicación de resultados de consulta impuesta](#), no especifique una `external_location` para la consulta CTAS. Athena emite un error y falla una consulta que especifica un `external_location` en este caso. Por ejemplo, se produce un error en esta consulta, si invalida la configuración del lado del cliente para la ubicación de resultados de la consulta, que obliga a que el grupo de trabajo utilice su propia ubicación: `CREATE TABLE <DB>.<TABLE1> WITH (format='Parquet',`

```
external_location='s3://DOC-EXAMPLE-BUCKET/test/') AS SELECT * FROM
<DB>.<TABLE2> LIMIT 10;
```

Podría ver los siguientes errores. Esta tabla proporciona una lista de algunos de los errores relacionados con grupos de trabajo y sugiere soluciones.

### Errores de grupos de trabajo

Error	Se produce cuando...
estado de consulta CANCELADO. Se ha superado el límite de bytes escaneados.	Una consulta alcanza un límite de datos por consulta y se cancela. Considere la posibilidad de volver a escribir la consulta para que lea menos datos o póngase en contacto con su administrador de cuenta.
User: <i>arn:aws:iam::123456789012:user/abc</i> is not authorized to perform: athena:StartQueryExecution on resource: <i>arn:aws:athena:us-east-1:123456789012:workgroup/workgroupname</i> (El usuario: arn:aws:iam::123456789012:user/abc no tiene autorización para realizar: athena:StartQueryExecution en el recurso: arn:aws:athena:us-east-1:123456789012:workgroup/workgroupname)	Un usuario ejecuta una consulta en un grupo de trabajo, pero no tiene acceso a ella. Actualice su política para tener acceso al grupo de trabajo.
INVALID_INPUT. WorkGroup <name> is disabled. (INVALID_INPUT. WorkGroup <nombre> se ha deshabilitado)	Un usuario ejecuta una consulta en un grupo de trabajo, pero el grupo de trabajo está deshabilitado. Su administrador podría haber deshabilitado el grupo de trabajo. También es posible que no tenga acceso a él. En ambos casos, póngase en contacto con un administrador con acceso para modificar grupos de trabajo.

Error	Se produce cuando...
<p>INVALID_INPUT. WorkGroup &lt;name&gt; is not found. (INVALID_INPUT. WorkGroup &lt;nombre&gt; no se encuentra)</p>	<p>Un usuario ejecuta una consulta en un grupo de trabajo, pero el grupo de trabajo no existe. Esto podría ocurrir si se eliminó el grupo de trabajo. Cambie a otro grupo de trabajo para ejecutar su consulta.</p>
<p>InvalidRequestException: when calling the StartQueryExecution operation: No output location provided. An output location is required either through the Workgroup result configuration setting or as an API input. (InvalidRequestException: al llamar a la operación StartQueryExecution: no se proporciona ninguna ubicación de salida. Se requiere una ubicación de salida mediante la configuración de resultados del grupo de trabajo o como entrada de la API)</p>	<p>Un usuario ejecuta una consulta con la API sin especificar la ubicación para los resultados de la consulta. Debe establecer la ubicación de salida de los resultados de la consulta de una de las siguientes dos formas: para consultas individuales mediante <a href="#">OutputLocation</a> (del cliente) o en el grupo de trabajo, mediante <a href="#">WorkGroupConfiguration</a>.</p>
<p>La consulta Crear tabla como selección devolvió un error porque se envió con una propiedad "external_location" a un grupo de trabajo de Athena que aplica una ubicación de salida centralizada para todas las consultas. Elimine la propiedad 'external_location' y vuelva a enviar la consulta.</p>	<p>Si el grupo de trabajo en el que se ejecuta una consulta se ha configurado con una <a href="#">ubicación de resultados de consulta impuesta</a> y especifique a una external_location para la consulta CTAS. En este caso, elimine la external_location y vuelva a ejecutar la consulta.</p>
<p>Cannot create prepared statement <i>prepared_statement_name</i> . The number of prepared statements in this workgroup exceeds the limit of 1000. (No se puede crear la instrucción preparada "prepared_statement_name". El número de instrucciones preparadas en este grupo de trabajo supera el límite de 1000)</p>	<p>El grupo de trabajo contiene más del límite de 1000 instrucciones preparadas. Para solucionar este problema, utilice <a href="#">DEALLOCATE PREPARE</a> para eliminar una o más instrucciones preparadas del grupo de trabajo. También puede crear un grupo de trabajo nuevo.</p>

## Control de costos y supervisión de consultas con métricas y eventos de CloudWatch

Los grupos de trabajo le permiten definir los límites de control de uso de datos por consulta o por grupo de trabajo, configurar alarmas cuando se superan los límites y publicar métricas de consultas en CloudWatch.

En cada grupo de trabajo, puede:

- Configurar controles de uso de datos por consulta y por grupo de trabajo, y establecer las medidas que se adoptarán en caso de que las consultas superen los umbrales.
- Ver y analizar las métricas de consultas, y publicarlas en CloudWatch. Si crea un grupo de trabajo en la consola, la configuración para la publicación de las métricas en CloudWatch se selecciona por usted. Si utiliza las operaciones de la API, debe [habilitar la publicación de las métricas](#). Cuando se publican las métricas, se muestran en la pestaña Métrica en el panel Grupos de trabajo. Las métricas están deshabilitadas de forma predeterminada para el grupo de trabajo principal.

### Video

En el siguiente video se muestra cómo crear paneles personalizados y configurar alarmas y desencadenadores en métricas en CloudWatch. Puede utilizar los paneles completados de forma automática directamente desde la consola de Athena para consumir estas métricas de consultas.

### [Supervisión de consultas de Amazon Athena mediante Amazon CloudWatch](#)

#### Temas

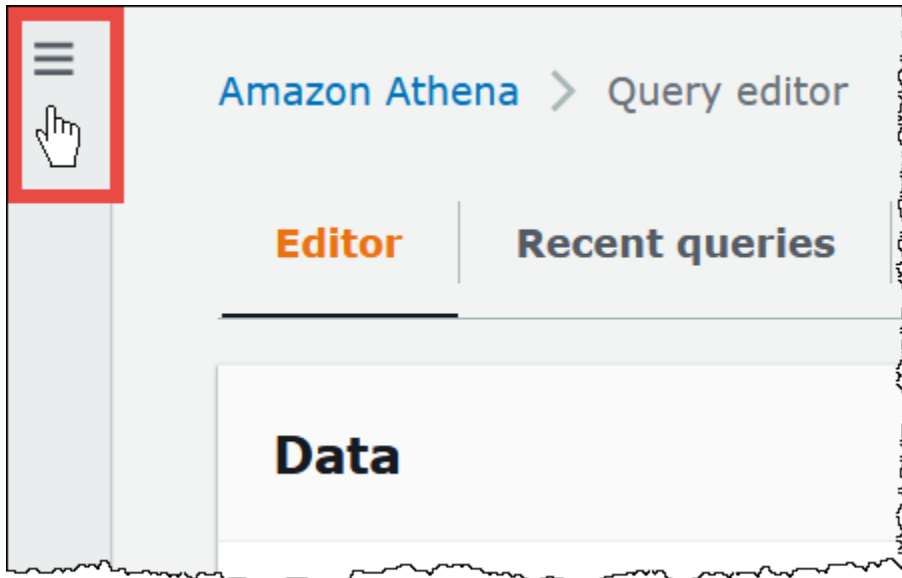
- [Habilitación de las métricas de consultas de CloudWatch](#)
- [Supervisión de las consultas de Athena con métricas de CloudWatch](#)
- [Supervisión de eventos de Athena con Amazon EventBridge](#)
- [Supervisión de las métricas de uso de Athena](#)
- [Configuración de los límites de control del uso de datos](#)

#### Habilitación de las métricas de consultas de CloudWatch

Cuando crea un grupo de trabajo en la consola, la configuración para la publicación de las métricas de consultas en CloudWatch se selecciona de forma predeterminada.

Para habilitar o deshabilitar las métricas de consulta en la consola de Athena para un grupo de trabajo

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.



3. En el panel de navegación, elija Grupos de trabajo.
4. Elija el enlace del grupo de trabajo que quiere modificar.
5. En la página de detalles del grupo de trabajo, elija Editar.
6. En la sección Configuración, seleccione Publicar métricas de consulta en AWS CloudWatch.

Si utiliza operaciones de la API, la interfaz de línea de comandos o la aplicación cliente con el controlador JDBC para crear grupos de trabajo, para habilitar la publicación de las métricas de consultas, establezca `PublishCloudWatchMetricsEnabled` en `true` en [WorkGroupConfiguration](#). En el siguiente ejemplo, se muestra únicamente la configuración de métricas y omite otra configuración:

```
"WorkGroupConfiguration": {  
  "PublishCloudWatchMetricsEnabled": "true"  
  ....  
}
```

## Supervisión de las consultas de Athena con métricas de CloudWatch

Athena publica métricas relacionadas con las consultas en Amazon CloudWatch cuando se selecciona la opción [Publicar métricas de consultas en CloudWatch](#). Puede crear tableros personalizados, configurar alarmas y desencadenadores en métricas en CloudWatch, o utilizar los paneles completados de forma automática directamente desde la consola de Athena.

Al habilitar las métricas de consultas en grupos de trabajo, las métricas se mostrarán en la pestaña Métricas en el panel Grupos de trabajo, para cada grupo de trabajo en la consola de Athena.

Athena publica las siguientes métricas en la consola de CloudWatch:

- **DPUAllocated**: el número total de DPU (unidades de procesamiento de datos) aprovisionadas en una reserva de capacidad para ejecutar consultas.
- **DPUConsumed**: el número de DPU consumidas activamente por las consultas en un estado RUNNING determinado de una reserva. La métrica se emite solo cuando el grupo de trabajo está asociado a una reserva de capacidad e incluye todos los grupos de trabajo asociados a una reserva.
- **DPUCount**: el número máximo de DPU consumidas por la consulta, que se publica exactamente una vez cuando se completa la consulta.
- **EngineExecutionTime**: El número de milisegundos que la consulta tardó en ejecutarse.
- **ProcessedBytes**: Número de bytes que Athena ha analizado por consulta de DML.
- **QueryPlanningTime**: Número de milisegundos que Athena tardó en planificar el flujo de procesamiento de consultas.
- **QueryQueueTime**: Número de milisegundos que la consulta estaba en la cola de consultas en espera de recursos.
- **ServicePreProcessingTime**: número de milisegundos que Athena tardó en preprocesar la consulta antes de enviarla al motor de consulta.
- **ServiceProcessingTime**: Número de milisegundos que Athena tardó en procesar los resultados de la consulta después de que el motor de consulta terminara de ejecutarla.
- **TotalExecutionTime**: Número de milisegundos que Athena tardó en ejecutar una consulta de DDL o DML.

Para obtener descripciones más completas, consulte [Lista de métricas y dimensiones de CloudWatch para Athena](#) más adelante en este documento.

Estas métricas tienen las dimensiones siguientes:

- `CapacityReservation`: el nombre de la reserva de capacidad utilizada para ejecutar la consulta, si corresponde.
- `QueryState`: SUCCEEDED, FAILED, o CANCELED
- `QueryType`: DML, DDL, o UTILITY
- `WorkGroup`: nombre del grupo de trabajo

Athena publica las siguientes métricas en la consola de CloudWatch bajo el espacio de nombres `AmazonAthenaForApacheSpark`:

- `DPUCount`: número de DPU consumidas durante la sesión para ejecutar los cálculos.

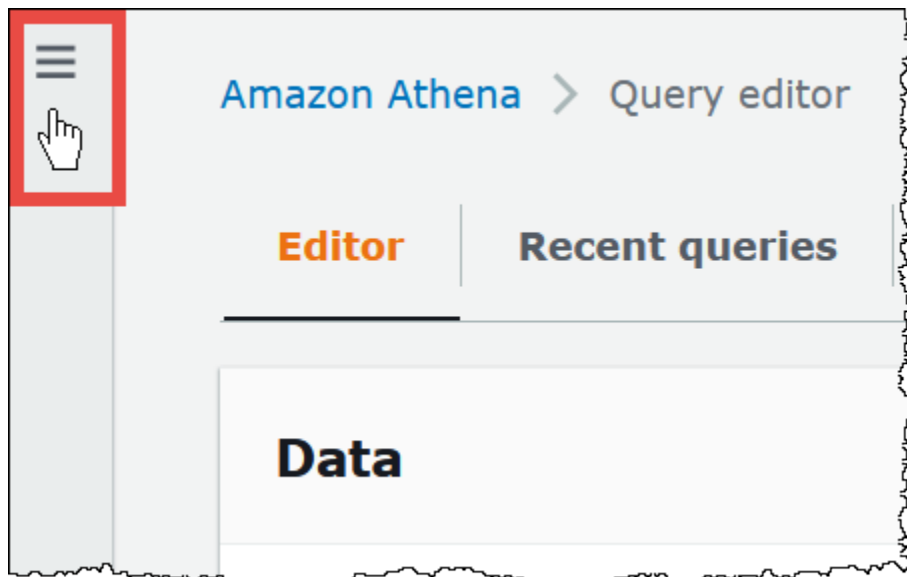
Esta métrica tiene las siguientes dimensiones:

- `SessionId`: ID de la sesión en la que se envían los cálculos.
- `WorkGroup`: nombre del grupo de trabajo.

Para obtener más información, consulte [Lista de métricas y dimensiones de CloudWatch para Athena](#) más adelante en este tema. Para obtener información sobre las métricas del uso de Athena, consulte [Supervisión de las métricas de uso de Athena](#).

Para ver las métricas de la consulta para un grupo de trabajo en la consola

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.



3. En el panel de navegación, elija Grupos de trabajo.
4. Elija el grupo de trabajo que quiera de la lista y, a continuación, elija la pestaña Métricas.

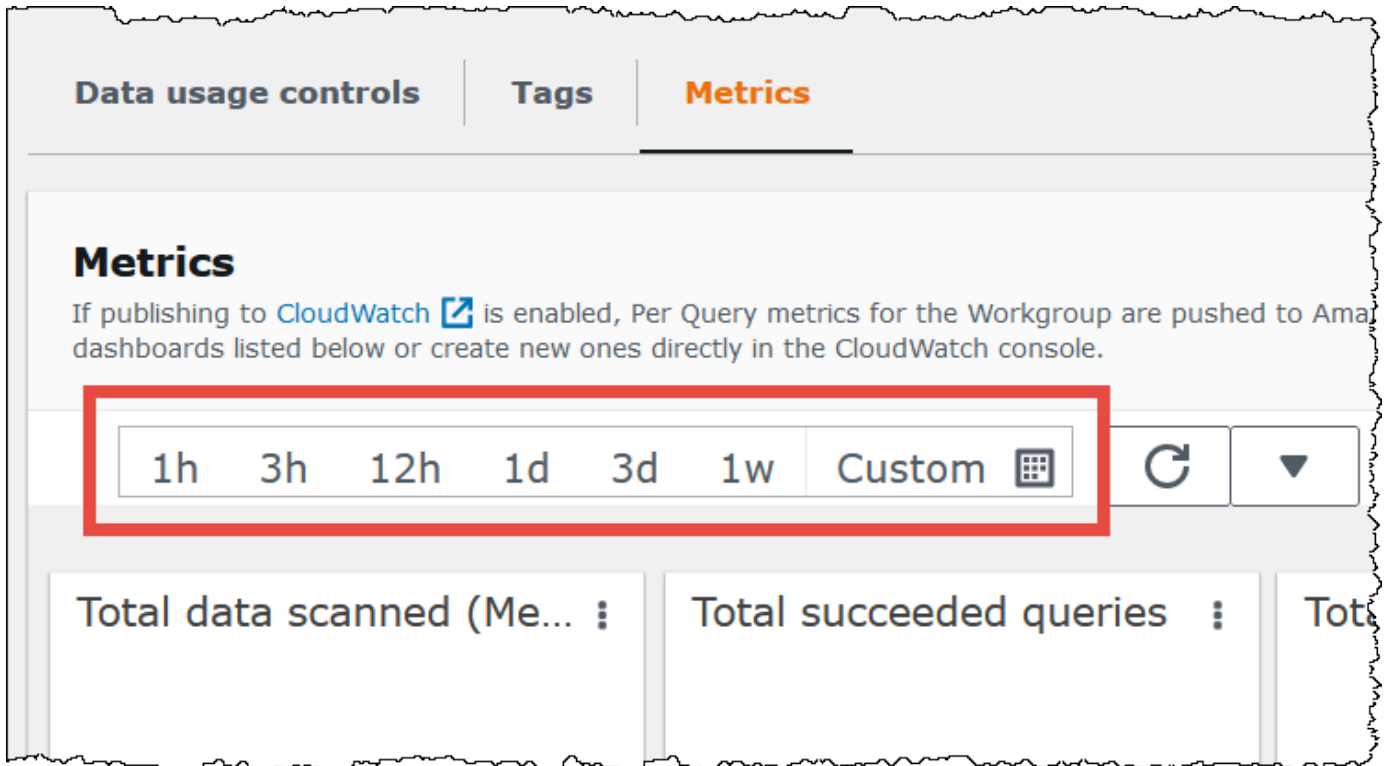
Aparecerá el panel de métricas.

#### Note

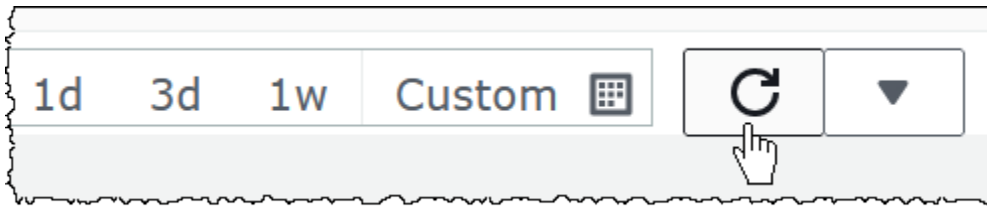
Si ha habilitado recientemente las métricas para el grupo de trabajo y/o no se ha producido ninguna actividad de consulta reciente, los gráficos del panel pueden estar vacíos. La actividad de consulta se recupera de CloudWatch en función del intervalo que especifique en el siguiente paso.

5. En la sección Métricas, elija el intervalo de métricas que Athena debe utilizar para obtener las métricas de consulta de CloudWatch, o bien especifique un intervalo personalizado.

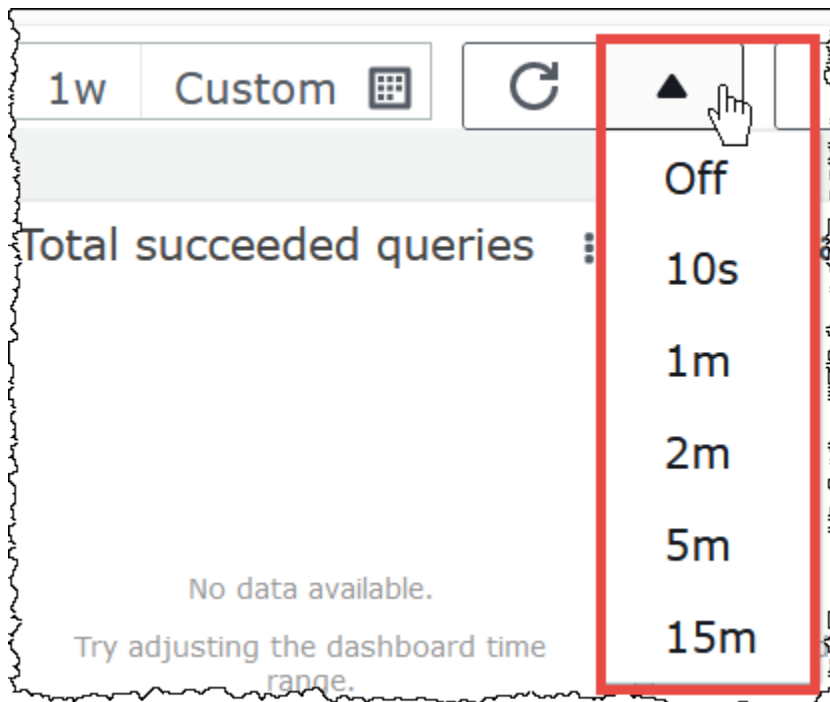




6. Para actualizar las métricas mostradas, elija el icono de actualización.



7. Haga clic en la flecha situada junto al icono de actualización para elegir con qué frecuencia quiere que se actualice la visualización de las métricas.



Para consultar las métricas en la consola de Amazon CloudWatch

1. Abra la consola de CloudWatch en <https://console.aws.amazon.com/cloudwatch/>.
2. En el panel de navegación, seleccione Métricas y, a continuación, Todas las métricas.
3. Seleccione el espacio de nombres de AWS/Athena.

Para ver métricas mediante la CLI

- Realice una de las siguientes acciones siguientes:
  - Para enumerar las métricas de Athena, abra un símbolo del sistema y utilice el siguiente comando:

```
aws cloudwatch list-metrics --namespace "AWS/Athena"
```

- Para mostrar todas las métricas disponibles, utilice el siguiente comando:

```
aws cloudwatch list-metrics"
```

## Lista de métricas y dimensiones de CloudWatch para Athena

Si habilitó métricas de CloudWatch en Athena, envía las siguientes métricas a CloudWatch por grupo de trabajo. Las siguientes métricas utilizan el espacio de nombres AWS/Athena.

Nombre de métrica	Descripción
DPUAllocated	El número total de DPUs (unidades de procesamiento de datos) aprovisionadas en una reserva de capacidad para ejecutar consultas.
DPUConsumed	El número de DPU consumidas activamente por las consultas en un estado RUNNING determinado de una reserva. Esta métrica se emite solo cuando el grupo de trabajo está asociado a una reserva de capacidad e incluye todos los grupos de trabajo asociados a una reserva. Si mueve un grupo de trabajo de una reserva a otra, la métrica incluye datos del momento en que el grupo de trabajo pertenecía a la primera reserva. Para obtener información sobre cómo compartir reservas de capacidad, consulte <a href="#">Administración de la capacidad de procesamiento de consultas</a> .
DPUCount	El número máximo de DPU consumidas por la consulta, que se publica exactamente una vez cuando se completa la consulta. Esta métrica se emite solo para los grupos de trabajo asociados a una reserva de capacidad.
EngineExecutionTime	El número de milisegundos que la consulta tardó en ejecutarse.
ProcessedBytes	Número de bytes que Athena ha analizado por consulta de DML. En el caso de las consultas canceladas (ya sea por los usuarios, automáticamente o por alcanzar el límite), se incluye la cantidad de datos escaneados antes de la cancelación. Esta métrica no se informa para las consultas de DDL.
QueryPlanningTime	Número de milisegundos que Athena tardó en planificar el flujo de procesamiento de consultas. Esto incluye el tiempo dedicado a recuperar las particiones de tabla del origen de datos. Tenga en cuenta que debido a que el motor de consultas

Nombre de métrica	Descripción
	realiza la planificación de consultas, el tiempo de planificación de consultas es un subconjunto de EngineExecutionTime.
QueryQueueTime	Número de milisegundos que la consulta estaba en la cola de consultas en espera de recursos. Tenga en cuenta que si se producen errores transitorios, la consulta se puede agregar automáticamente de nuevo a la cola.
ServicePreProcessingTime	Número de milisegundos que Athena tardó en preprocesar la consulta antes de enviarla al motor de consulta.
ServiceProcessingTime	Número de milisegundos que Athena tardó en procesar los resultados de la consulta después de que el motor de consulta terminara de ejecutarla.
TotalExecutionTime	Número de milisegundos que Athena tardó en ejecutar una consulta de DDL o DML. TotalExecutionTime incluye QueryQueueTime, QueryPlanningTime, EngineExecutionTime y ServicePreProcessingTime.

Las métricas de CloudWatch para Athena tienen las siguientes dimensiones.

Dimensión	Descripción
CapacityReservation	El nombre de la reserva de capacidad utilizada para ejecutar la consulta, si corresponde. Cuando no se utiliza una reserva de capacidad, esta dimensión no devuelve ningún dato.
QueryState	El estado de la consulta.  Estadísticas válidas: SUCCEEDED, FAILED o CANCELED.
QueryType	El tipo de consulta.  Estadísticas válidas: DDL, DML o UTILITY. El tipo de instrucción de consulta que se ejecutó. DDL indica las instrucciones de consulta DDL (lenguaje de definición de datos). DML indica

Dimensión	Descripción
	instrucciones de consulta DML (lenguaje de manipulación de datos), como CREATE TABLE AS SELECT. UTILITY indica instrucciones de consulta distintas de DDL y DML, como SHOW CREATE TABLE o DESCRIBE TABLE.
WorkGroup	El nombre del grupo de trabajo.

## Supervisión de eventos de Athena con Amazon EventBridge

Puede utilizar Amazon Athena con Amazon EventBridge para recibir notificaciones en tiempo real sobre el estado de las consultas. Cuando una consulta ha enviado estados de transiciones, Athena publica un evento en EventBridge que contiene información sobre esa transición de estado de consulta. Puede escribir reglas simples para eventos que le interesen y realizar acciones automatizadas cuando un evento coincida con una regla. Por ejemplo, puede crear una regla que invoque una función AWS Lambda cuando una consulta alcance un estado terminal. Los eventos se emiten en la medida de lo posible.

Antes de crear reglas de eventos para Athena, debe hacer lo siguiente:

- Familiarizarse con los eventos, las reglas y los destinos de EventBridge. Para obtener más información, consulte [¿Qué es Amazon EventBridge?](#) Para obtener más información sobre cómo configurar reglas, consulte [Introducción a Amazon EventBridge](#).
- Crear el destino o destinos que se van a usar en las reglas de eventos.

### Note

Actualmente, Athena ofrece un tipo de evento, Athena Query State Change (Cambio de estado de consulta de Athena), pero puede agregar otros tipos de eventos y detalles. Si va a deserializar datos JSON de eventos mediante programación, asegúrese de que la aplicación esté preparada para tratar propiedades desconocidas para evitar problemas si se agregan estas propiedades adicionales.

## Formato de eventos de Athena

El siguiente es el patrón básico de un evento de Amazon Athena.

```
{
  "source": [
    "aws.athena"
  ],
  "detail-type": [
    "Athena Query State Change"
  ],
  "detail": {
    "currentState": [
      "SUCCEEDED"
    ]
  }
}
```

## Evento de cambio de estado de la consulta de Athena

En el ejemplo siguiente se muestra el evento de cambio de estado de la consulta de Athena con un valor `currentState` de `SUCCEEDED`.

```
{
  "version": "0",
  "id": "abcdef00-1234-5678-9abc-def012345678",
  "detail-type": "Athena Query State Change",
  "source": "aws.athena",
  "account": "123456789012",
  "time": "2019-10-06T09:30:10Z",
  "region": "us-east-1",
  "resources": [

  ],
  "detail": {
    "versionId": "0",
    "currentState": "SUCCEEDED",
    "previousState": "RUNNING",
    "statementType": "DDL",
    "queryExecutionId": "01234567-0123-0123-0123-012345678901",
    "workgroupName": "primary",
    "sequenceNumber": "3"
  }
}
```

En el ejemplo siguiente se muestra el evento de cambio de estado de la consulta de Athena con un valor `currentState` de `FAILED`. El bloque `athenaError` aparece solo cuando el valor de `currentState` es `FAILED`. Para obtener información acerca de los valores para `errorCategory` y `errorType`, consulte [Catálogo de errores de Athena](#).

```
{
  "version": "0",
  "id": "abcdef00-1234-5678-9abc-def012345678",
  "detail-type": "Athena Query State Change",
  "source": "aws.athena",
  "account": "123456789012",
  "time": "2019-10-06T09:30:10Z",
  "region": "us-east-1",
  "resources": [
  ],
  "detail": {
    "athenaError": {
      "errorCategory": 2.0, //Value depends on nature of exception
      "errorType": 1306.0, //Type depends on nature of exception
      "errorMessage": "Amazon S3 bucket not found", //Message depends on nature
of exception
      "retryable": false //Retryable value depends on nature of exception
    },
    "versionId": "0",
    "currentState": "FAILED",
    "previousState": "RUNNING",
    "statementType": "DML",
    "queryExecutionId": "01234567-0123-0123-0123-012345678901",
    "workgroupName": "primary",
    "sequenceNumber": "3"
  }
}
```

## Propiedades de salida

La salida JSON incluye las siguientes propiedades.

Propiedad	Descripción
<code>athenaError</code>	Aparece solo cuando el valor de <code>currentState</code> es <code>FAILED</code> . Contiene información sobre el error que se ha producido, incluida la categoría de error, el tipo de error, el mensaje de error y si se puede reintentar

Propiedad	Descripción
	la acción que ha provocado el error. Los valores de cada uno de estos campos dependen de la naturaleza del error. Para obtener información acerca de los valores para <code>errorCategory</code> y <code>errorType</code> , consulte <a href="#">Catálogo de errores de Athena</a> .
<code>versionId</code>	Número de versión del esquema del objeto de detalle.
<code>currentState</code>	El estado que adoptó la consulta cuando se produjo el evento.
<code>previousState</code>	El estado que tenía la consulta cuando se produjo el evento.
<code>statementType</code>	El tipo de instrucción de consulta que se ejecutó.
<code>queryExecutionId</code>	El identificador único de la consulta que se ejecutó.
<code>workgroupName</code>	El nombre del grupo de trabajo en el que se ejecutó la consulta.
<code>sequenceNumber</code>	Un número monótonamente creciente que permite la deduplicación y la ordenación de eventos entrantes que implican una sola ejecución de consulta. Cuando se publican eventos duplicados para la misma transición de estado, el valor de <code>sequenceNumber</code> es el mismo. Cuando una consulta experimenta una transición de estado más de una vez, como las consultas que experimentan una puesta en cola extraña, puede utilizar <code>sequenceNumber</code> para ordenar eventos con valores de <code>previousState</code> y <code>currentState</code> idénticos.

## Ejemplo

En el siguiente ejemplo se publican eventos en un tema de Amazon SNS al que se ha suscrito. Cuando se realice una consulta en Athena, recibirá un correo electrónico. En el ejemplo se presupone que el tema de Amazon SNS existe y que se ha suscrito a él.



## Para publicar eventos de Athena en un tema de Amazon SNS

1. Cree el destino del tema de Amazon SNS. Conceda a la entidad principal de EventBridge Events Service el permiso `events.amazonaws.com` para publicar en el tema de Amazon SNS, como en el siguiente ejemplo.

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Action": "sns:Publish",
  "Resource": "arn:aws:sns:us-east-1:111111111111:your-sns-topic"
}
```

2. Utilice el comando `events put-rule` de la AWS CLI para crear una regla para eventos de Athena, como en el siguiente ejemplo.

```
aws events put-rule --name {ruleName} --event-pattern '{"source": ["aws.athena"]}'
```

3. Utilice el comando `events put-targets` de la AWS CLI para asociar el destino del tema de Amazon SNS, como en el siguiente ejemplo.

```
aws events put-targets --rule {ruleName} --targets Id=1,Arn=arn:aws:sns:us-east-1:111111111111:your-sns-topic
```

4. Realice una consulta en Athena y observe el destino que se está invocando. Debería recibir mensajes de correo electrónico correspondientes del tema de Amazon SNS.

## Uso de AWS User Notifications con Amazon Athena

Puede utilizar las [AWS User Notifications](#) para configurar los canales de entrega a fin de recibir notificaciones sobre los eventos de Amazon Athena. Recibirá una notificación cuando un evento coincida con una regla que especifique. Puede recibir notificaciones de eventos a través de varios canales, como correo electrónico, notificaciones por chat de [AWS Chatbot](#) o notificaciones de inserción de [AWS Console Mobile Application](#). También puede ver las notificaciones en el [Centro de notificaciones de la consola](#). Las Notificaciones de usuario admiten la agregación, lo que puede reducir el número de notificaciones que recibe durante eventos específicos.

Para obtener más información, consulte la [Guía del usuario de AWS User Notifications](#).

## Supervisión de las métricas de uso de Athena

Puede usar las métricas de uso de CloudWatch para proporcionar visibilidad de cómo su cuenta usa los recursos mostrando el uso actual del servicio en los gráficos y paneles de CloudWatch.

Para Athena, las métricas de disponibilidad de uso corresponden a las cuotas de Servicio de AWS para Athena. Puede configurar alarmas que le avisen cuando su uso se acerque a una Service Quota. Para obtener más información acerca de Service Quotas de Athena, consulte [Service Quotas](#). Para obtener más información sobre las métricas de uso de AWS, consulte [Métricas de uso de AWS](#) en la Guía del usuario de Amazon CloudWatch.

Athena publica las siguientes métricas en el espacio de nombres AWS/Usage.

Nombre de métrica	Descripción
ResourceCount	<p>La suma de todas las consultas en cola y en ejecución de una cuenta por Región de AWS, separadas por tipo de consulta (DML o DDL). El máximo es la única estadística útil para esta métrica.</p> <p>Esta métrica se publica periódicamente cada minuto. Si no se ejecuta ninguna consulta, la métrica no reporta nada (ni siquiera 0). La métrica solo se publica si se ejecutan consultas activas en el momento en que se toma la métrica.</p>

Las siguientes dimensiones se utilizan para ajustar las métricas de uso publicadas por Athena.

Dimensión	Descripción
Service	El nombre de Servicio de AWS que contiene el recurso. Para Athena, el valor de esta dimensión es Athena.
Resource	El tipo de recurso que se está ejecutando. El valor del recurso para el uso de consultas de Athena es ActiveQueryCount .
Type	El tipo de entidad que se notifica. Actualmente, el único valor válido para las métricas de uso de Athena es Resource.

Dimensión	Descripción
Class	La clase de recurso a la que se realiza el seguimiento. Para Athena, Class puede ser un DML o un DDL.

Visualización de las métricas de uso de los recursos de Athena en la consola de CloudWatch

Puede usar la consola de CloudWatch para ver un gráfico de métricas de uso de Athena y configurar alarmas que alerten cuando el uso se acerca a una cuota de servicio.

Para ver las métricas de uso de los recursos de Athena

1. Abra la consola de CloudWatch en <https://console.aws.amazon.com/cloudwatch/>.
2. En el panel de navegación, seleccione Métricas y, a continuación, Todas las métricas.
3. Elija Uso y, luego, Por recurso de AWS.

Aparecerá la lista de métricas de uso de cuotas de servicio.

4. Seleccione la casilla que está al lado de Athena y ActiveQueryCount.
5. Elija la pestaña Métricas diagramadas.

El gráfico anterior muestra el uso actual del recurso de AWS.

Para obtener información sobre cómo agregar cuotas de servicio al gráfico y configurar una alarma que notifique si se acerca a la cuota de servicio, consulte [Visualización de las cuotas de servicio y configuración de alarmas](#) en la Guía del usuario de Amazon CloudWatch. Para obtener información sobre la configuración de los límites de uso por grupo de trabajo, consulte [Configuración de los límites de control del uso de datos](#).

Configuración de los límites de control del uso de datos

Athena le permite configurar dos tipos de controles de costos: límite por consulta y límite por grupo de trabajo. Para cada grupo de trabajo, puede establecer un solo límite por consulta y varios límites por grupo de trabajo.

- El límite de control por consulta especifica la cantidad total de datos escaneados por consulta. Si cualquier consulta que se ejecuta en el grupo de trabajo supera el límite, se cancela. Puede crear solo un límite de control por consulta en un grupo de trabajo y se aplica a cada consulta que se

ejecuta en ella. Edite el límite si necesita cambiarlo. Para conocer los pasos en detalle, consulte [Para crear un control de uso de datos por consulta](#).

- El límite de control de uso de datos a escala de grupo de trabajo especifica la cantidad total de datos escaneados para todas las consultas que se ejecutan en este grupo de trabajo durante el periodo de tiempo especificado. Puede crear varios límites por grupo de trabajo. El límite de consulta en todo el grupo de trabajo le permite definir varios umbrales en resúmenes por hora o diarios en los datos examinados por las consultas que se ejecutan en el grupo de trabajo.

Si la cantidad agregada de datos escaneados supera el umbral, puede enviar una notificación a un tema de Amazon SNS. Para ello, configure una alarma de Amazon SNS y una acción en la consola de Athena para notificar a un administrador cuando se supere el límite. Para conocer los pasos en detalle, consulte [Para crear un control de uso de datos por grupo de trabajo](#). También puede crear una alarma y una acción en cualquiera de las métricas que Athena publica desde la consola de CloudWatch. Por ejemplo, puede establecer una alerta en una serie de consultas fallidas. Esta alerta puede desencadenar un correo electrónico a un administrador si el número supera un determinado umbral. Si se supera el límite, una acción envía una notificación de alarma de Amazon SNS a los usuarios especificados.

Otras acciones que puede llevar a cabo:

- Invoque una función de Lambda. Para obtener más información, consulte [Invocación de funciones de Lambda mediante notificaciones de Amazon SNS](#) en la Guía para desarrolladores de Amazon Simple Notification Service.
- Deshabilite el grupo trabajo para detener la ejecución de consultas adicionales. Para ver los pasos, consulte [Habilitar o deshabilitar un grupo de trabajo](#).

Los límites por consulta y por grupo de trabajo son independientes entre sí. Se toma una acción especificada siempre que se supera cualquiera de los límites. Si dos o más usuarios ejecutan consultas al mismo tiempo en el mismo grupo de trabajo, es posible que cada consulta no supere cualquiera de los límites especificados, pero la suma total de datos escaneados supera el límite de uso de datos por grupo de trabajo. En este caso, se envía al usuario una alarma de Amazon SNS.

Para crear un control de uso de datos por consulta

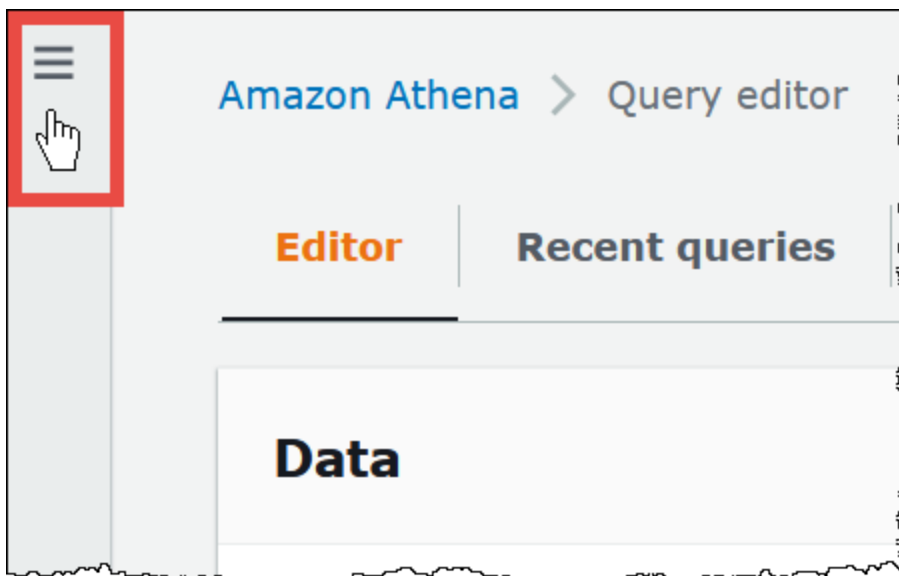
El límite de control por consulta especifica la cantidad total de datos escaneados por consulta. Si cualquier consulta que se ejecuta en el grupo de trabajo supera el límite, se cancela. Las consultas canceladas se cobran de acuerdo con los [Precios de Amazon Athena](#).

**Note**

En el caso de consultas canceladas o fallidas, es posible que Athena ya haya escrito resultados parciales en Amazon S3. En estos casos, Athena no elimina los resultados parciales del prefijo de Amazon S3 donde se almacenan los resultados. Debe eliminar el prefijo de Amazon S3 con resultados parciales. Athena utiliza cargas multiparte de Amazon S3 para escribir datos en Amazon S3. Le recomendamos que establezca la política del ciclo de vida del bucket para finalizar las cargas multiparte en casos en los que las consultas devuelven un error. Para obtener más información, consulte [Anulación de cargas multiparte incompletas con una política de ciclo de vida de bucket](#) en la Guía del usuario de Amazon Simple Storage Service.


Puede crear solo un límite de control por consulta en un grupo de trabajo y se aplica a cada consulta que se ejecuta en ella. Edite el límite si necesita cambiarlo.

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.



3. En el panel de navegación, elija Grupos de trabajo.
4. Elija el nombre del grupo de trabajo en la lista.
5. En la pestaña Controles de uso de datos, en la sección Control de uso de datos por consulta, elija Administrar.

6. En la página Administrar control de uso de datos por consulta, especifique los siguientes valores:
  - En Límite de datos, especifique un valor comprendido entre 10 MB (mínimo) y 7 EB (máximo).

 Note

Estos son los límites impuestos por la consola para controles de uso de datos dentro de los grupos de trabajo. No representan ningún límite de la consulta en Athena.

- Para unidades, seleccione el valor de la unidad de la lista desplegable (por ejemplo, Kilobytes KB o Exabytes EB).

La acción predeterminada consiste en cancelar la consulta si se supera el límite. Esta configuración no se puede cambiar.

7. Seleccione Guardar.

Para crear o editar una alerta de uso de datos por grupo de trabajo

Puede establecer varios umbrales de alerta cuando las consultas que se ejecutan en un grupo de trabajo analizan una cantidad específica de datos en un periodo específico. Las alertas se implementan mediante las alarmas de Amazon CloudWatch y se aplican a todas las consultas del grupo de trabajo. Cuando se alcanza un umbral, puede hacer que Amazon SNS envíe un correo electrónico a los usuarios que especifique. Las consultas no se cancelan automáticamente cuando se alcanza un umbral.

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.
3. En el panel de navegación, elija Grupos de trabajo.
4. Elija el nombre del grupo de trabajo en la lista.
5. Elija Editar para editar la configuración del grupo de trabajo.
6. Desplácese hasta Alertas de uso de datos de grupos de trabajo: opcional y expándalo.
7. Elija Agregar alerta.
8. En Configuración del umbral de uso de datos, especifique los siguientes valores:

- En Umbral de datos, especifique un número y, a continuación, seleccione un valor de la unidad de la lista desplegable.
  - En Periodo, elija un periodo de la lista desplegable.
  - En Selección de temas de SNS, elija un tema de Amazon SNS de la lista desplegable. O bien, elija Crear un tema de SNS para ir directamente a la [Consola de Amazon SNS](#), cree el tema de Amazon SNS y configure una suscripción para uno de los usuarios de su cuenta de Athena. Para obtener más información, consulte [Introducción a Amazon SNS](#) en la Guía para desarrolladores de Amazon Simple Notification Service.
9. Elija Agregar alerta si va a crear una nueva alerta, o Guardar para guardar una alerta existente.

## Administración de la capacidad de procesamiento de consultas

Puede utilizar las reservas de capacidad a fin de obtener una capacidad de procesamiento dedicada para las consultas que ejecute en Athena. Con las reservas de capacidad, puede aprovechar las funciones de administración de la carga de trabajo que lo ayudan a priorizar, controlar y escalar sus cargas de trabajo interactivas más importantes. Por ejemplo, puede agregar capacidad en cualquier momento para aumentar la cantidad de consultas que puede ejecutar en simultáneo, controlar qué cargas de trabajo pueden utilizar la capacidad y compartir la capacidad entre las cargas de trabajo. Athena administra por completo la capacidad y la conserva durante el tiempo que la necesite. La configuración es sencilla y no es necesario realizar cambios en las instrucciones SQL.

A fin de obtener capacidad de procesamiento para sus consultas, cree una reserva de capacidad, especifique la cantidad de unidades de procesamiento de datos (DPU) que necesita y asigne uno o más grupos de trabajo a la reserva.

Los grupos de trabajo desempeñan un rol importante cuando se utilizan reservas de capacidad. Los grupos de trabajo permiten organizar las consultas en agrupaciones lógicas. Con las reservas de capacidad, puede asignar capacidad a los grupos de trabajo de forma selectiva para controlar el comportamiento de las consultas de cada grupo de trabajo y la forma en que se facturan. Para obtener información acerca de los grupos de trabajo, consulte [Uso de grupos de trabajo para controlar el acceso a las consultas y los costos](#).

Al asignar grupos de trabajo a las reservas, puede dar prioridad a las consultas que envía a los grupos de trabajo asignados. Por ejemplo, puede asignar capacidad a un grupo de trabajo que se utilice para consultas de informes financieros urgentes a fin de aislarlas de las consultas menos importantes de otro grupo de trabajo. Esto permite una ejecución coherente de las consultas para las

cargas de trabajo importantes y, al mismo tiempo, permite que otras cargas de trabajo se ejecuten de forma independiente.

Puede utilizar las reservas de capacidad y los grupos de trabajo de forma conjunta para cumplir distintos requisitos. A continuación se muestran algunas situaciones de ejemplo:

- **Aislamiento:** para aislar una carga de trabajo importante, se asigna un único grupo de trabajo a una reserva. Solo las consultas del grupo de trabajo asignado utilizan la capacidad de procesamiento de la reserva elegida.
- **Uso compartido:** varias cargas de trabajo pueden compartir la capacidad de una reserva. Por ejemplo, si desea un costo mensual predecible para un conjunto específico de cargas de trabajo, puede asignar varios grupos de trabajo a una sola reserva. Los grupos de trabajo asignados comparten la capacidad de la reserva.
- **Modelo múltiple:** puede utilizar las reservas de capacidad y la facturación por consulta en la misma cuenta. Por ejemplo, para garantizar la ejecución fiable de las consultas compatibles con una aplicación de producción, debe asignar un grupo de trabajo para esas consultas a una reserva de capacidad. Al desarrollar las consultas antes de pasarlas al grupo de trabajo de producción, utilice un grupo de trabajo independiente que no forme parte de una reserva y utilice la facturación por consulta.

## Descripción de las DPU

La capacidad se mide en unidades de procesamiento de datos (DPU). Las DPU representan los recursos de cómputo y memoria que Athena utiliza para acceder a los datos y procesarlos en su nombre. Una DPU proporciona 4 vCPU y 16 GB de memoria. La cantidad de DPU que especifique influirá en la cantidad de consultas que puede ejecutar de forma simultánea. Por ejemplo, una reserva con 256 DPU puede permitir aproximadamente el doble de consultas simultáneas que una reserva con 128 DPU.

Puede crear hasta 100 reservas de capacidad con un total de hasta 1000 DPU por cuenta y región. La cantidad mínima de DPU que puede solicitar es 24. Si necesita más de 1000 DPU para su caso de uso, póngase en contacto con [athena-feedback@amazon.com](mailto:athena-feedback@amazon.com).

Para obtener información sobre cómo calcular sus requisitos de capacidad, consulte [Determinación de los requisitos de capacidad](#). Para obtener información sobre precios, consulte [Precios de Amazon Athena](#).



## Consideraciones y limitaciones

- La característica requiere la [versión 3 del motor de Athena](#).
- Se puede asignar un solo grupo de trabajo a como máximo una reserva a la vez, y puede agregar un máximo de 20 grupos de trabajo a una reserva.
- No puede agregar grupos de trabajo habilitados para Spark a una reserva de capacidad.
- Para eliminar un grupo de trabajo que se ha asignado a una reserva, elimine primero el grupo de trabajo de la reserva.
- La cantidad mínima de DPU que puede aprovisionar es 24.
- Puede crear hasta 100 reservas de capacidad con un total de hasta 1000 DPU por cuenta y región.
- Las solicitudes de capacidad no están garantizadas y se pueden completar en hasta 30 minutos.
- Hay un periodo de facturación mínimo de 1 hora por reserva. Después de 1 hora, la capacidad se factura por minuto. Para obtener información sobre precios, consulte [Precios de Amazon Athena](#).
- La capacidad reservada no es transferible a otra reserva de capacidad, Cuenta de AWS o Región de AWS.
- Las consultas DDL sobre las reservas de capacidad consumen DPU.
- Las consultas que se ejecutan con la capacidad aprovisionada no se tienen en cuenta para los límites de consultas activas de DDL y DML.
- Si se utilizan todas las DPU, las consultas enviadas se ponen en cola. Estas consultas no se rechazan y no se utilizan bajo demanda.
- La métrica de CloudWatch DPUConsumed es por grupo de trabajo y no por reserva. Por lo tanto, si mueve un grupo de trabajo de una reserva a otra, la métrica DPUConsumed incluye datos del momento en que el grupo de trabajo pertenecía a la primera reserva. Para obtener más información sobre las métricas de CloudWatch en Athena, consulte [Supervisión de las consultas de Athena con métricas de CloudWatch](#).
- En la actualidad, la característica está disponible en las siguientes Regiones de AWS:
  - US East (Ohio)
  - Este de EE. UU. (Norte de Virginia)
  - Oeste de EE. UU. (Oregón)
  - Asia Pacífico (Singapur)
  - Asia-Pacífico (Sídney)
  - Asia Pacífico (Tokio)
  - Europa (Irlanda)

- [Europa \(Estocolmo\)](#)

## Temas

- [Determinación de los requisitos de capacidad](#)
- [Creación de reservas de capacidad](#)
- [Administración de reservas](#)
- [Políticas de IAM para reservas de capacidad](#)
- [API de reserva de capacidad de Athena](#)

## Determinación de los requisitos de capacidad

Antes de crear una reserva de capacidad, puede calcular la capacidad necesaria para poder asignarle el número correcto de DPU. Y, una vez utilizada una reserva, es posible que desee comprobar si la capacidad de la reserva es insuficiente o excesiva. En este tema se describen las técnicas que puede utilizar para realizar estos cálculos y también se describen algunas herramientas de AWS para evaluar el uso y el costo.

## Temas

- [Cálculo de la capacidad requerida](#)
- [Señales de que se necesita más capacidad](#)
- [Comprobación de la capacidad inactiva](#)
- [Herramientas para evaluar los requisitos de capacidad y el costo](#)

## Cálculo de la capacidad requerida

Al calcular los requisitos de capacidad, es útil tener en cuenta dos perspectivas: cuánta capacidad podría requerir una consulta en particular y cuánta capacidad podría necesitar en general.

## Cálculo de los requisitos de capacidad por consulta

Para determinar la cantidad de DPU que podría necesitar una consulta, puede utilizar las siguientes pautas:

- Las consultas DDL consumen 4 DPU.
- Las consultas DML consumen entre 4 y 124 DPU.

Athena determina el número de DPU que necesita una consulta DML cuando esta se envía. El número varía según el tamaño de los datos, el formato de almacenamiento, la construcción de la consulta y otros factores. Por lo general, Athena intenta seleccionar el número de DPU más bajo y eficiente. Si Athena determina que se necesita más potencia computacional para que la consulta se complete correctamente, aumentará el número de DPU asignadas a la consulta.

Cálculo de los requisitos de capacidad específicos de la carga de trabajo

Para determinar la capacidad que podría necesitar para ejecutar varias consultas al mismo tiempo, tenga en cuenta las pautas generales de la siguiente tabla:

Consultas simultáneas	DPU requeridas
10	40 o más
20	96 o más
30 o más	240 o más

Tenga en cuenta que la cantidad real de DPU que necesita depende de sus objetivos y patrones de análisis. Por ejemplo, si desea que las consultas comiencen inmediatamente sin colas, determine su demanda máxima de consultas simultáneas y, a continuación, aprovisione la cantidad de DPU en consecuencia.

Puede aprovisionar menos DPU que la demanda máxima, pero es posible que las consultas se pongan en cola cuando se produzca la demanda máxima. Cuando las consultas se ponen en cola, Athena mantiene las consultas en una cola y las ejecuta cuando hay capacidad disponible.

Si su objetivo es ejecutar las consultas dentro de un presupuesto fijo, puede utilizar la [AWS calculadora de precios](#) para determinar la cantidad de DPU necesarias.

Por último, recuerde que el tamaño de los datos, el formato de almacenamiento y la forma en que se escribe una consulta influyen en las DPU que requiere una consulta. Para aumentar el rendimiento de las consultas, puede comprimir o particionar los datos o convertirlos en formatos de columnas. Para obtener más información, consulte [Ajuste del rendimiento en Athena](#).

Señales de que se necesita más capacidad

Los mensajes de error de capacidad insuficiente y la cola de consultas son dos indicios de que la capacidad asignada es inadecuada.

Si las consultas fallan y aparece un mensaje de error de capacidad insuficiente, es probable que el recuento de DPU de la reserva de capacidad sea demasiado bajo para la carga de trabajo de la consulta. Por ejemplo, si tiene una reserva con 24 DPU y ejecuta una consulta que requiere más de 24 DPU, la consulta fallará. Para controlar este error de consulta, puede utilizar [Eventos de EventBridge](#) de Athena. Intente agregar más DPU y vuelva a ejecutar la consulta.

Si hay muchas consultas en cola, significa que otras consultas utilizan al máximo su capacidad. Para reducir las colas, realice una de las siguientes acciones:

- Agregue las DPU a su reserva para aumentar la simultaneidad de consultas.
- Elimine los grupos de trabajo de su reserva para liberar capacidad para otras consultas.

Para comprobar si hay demasiadas colas de consultas, utilice la [métrica de CloudWatch](#) de tiempo de cola de consultas de Athena para los grupos de trabajo de su reserva de capacidad. Si el valor supera el umbral preferido, puede agregar las DPU a la reserva de capacidad.

### Comprobación de la capacidad inactiva

Para comprobar la capacidad inactiva, puede reducir el número de DPU de la reserva o aumentar su carga de trabajo y, a continuación, observar los resultados.

### Para comprobar la capacidad inactiva

1. Realice una de las acciones siguientes:
  - Reduzca la cantidad de DPU de su reserva (reduzca los recursos disponibles).
  - Agregue grupos de trabajo a su reserva (aumente la carga de trabajo).
2. Utilice [CloudWatch](#) para medir el tiempo de espera de las consultas.
3. Si el tiempo de espera aumenta más allá del nivel deseado, realice una de las siguientes acciones.
  - Elimine los grupos de trabajo.
  - Agregue las DPU a su reserva de capacidad.
4. Después de cada cambio, compruebe el rendimiento y el tiempo de espera de las consultas.
5. Siga ajustando la carga de trabajo o el recuento de DPU para lograr el equilibrio deseado.

Si no desea mantener la capacidad fuera del periodo de tiempo preferido, puede [cancelar](#) la reserva y crear otra más adelante. Sin embargo, aunque haya cancelado recientemente la capacidad de otra reserva, las solicitudes de capacidad nueva no están garantizadas y la creación de reservas nuevas lleva tiempo.

Herramientas para evaluar los requisitos de capacidad y el costo

Puede utilizar los siguientes servicios y características en AWS para medir el uso y los costos de Athena.

### Métricas de CloudWatch

Puede configurar Athena para que publique métricas relacionadas con consultas en Amazon CloudWatch a nivel de grupo de trabajo. Después de habilitar las métricas para el grupo de trabajo, las métricas de las consultas del grupo de trabajo se muestran en la consola de Athena, en la página de detalles del grupo de trabajo.

Para obtener información sobre las métricas de Athena que se publican en CloudWatch y sus dimensiones, consulte [Supervisión de las consultas de Athena con métricas de CloudWatch](#).

### Métricas de uso de CloudWatch

Puede usar las métricas de uso de CloudWatch para proporcionar visibilidad de cómo su cuenta usa los recursos mostrando el uso actual del servicio en los gráficos y paneles de CloudWatch. En Athena, las métricas de disponibilidad de uso corresponden a las [cuotas de servicio](#) de AWS para Athena. Puede configurar alarmas que le avisen cuando su uso se acerque a una Service Quota.

Para obtener más información, consulte [Supervisión de las métricas de uso de Athena](#).

### Eventos de Amazon EventBridge

Puede utilizar Amazon Athena con Amazon EventBridge para recibir notificaciones en tiempo real sobre el estado de las consultas. Cuando una consulta ha enviado estados de transiciones, Athena publica un evento en EventBridge que contiene información sobre la transición de estado de consulta. Puede escribir reglas simples para eventos que le interesen y realizar acciones automatizadas cuando un evento coincida con una regla.

Para obtener más información, consulte los recursos siguientes.

- [Supervisión de eventos de Athena con Amazon EventBridge](#)
- [¿Qué es Amazon EventBridge?](#)

- [Eventos de Amazon EventBridge](#)

## Etiquetas

En Athena, las reservas de capacidad admiten etiquetas. Una etiqueta consta de una clave y un valor. Para realizar un seguimiento de los costos en Athena, puede utilizar etiquetas de asignación de costos generadas por AWS. AWS utiliza las etiquetas de asignación de costos para organizar los costos de los recursos en el [informe de costos y uso](#). Esto le facilita la categorización y el seguimiento de los costos de AWS. [Para activar las etiquetas de asignación de costos para Athena, utilice la consola de AWS Billing and Cost Management.](#)

Para obtener más información, consulte los recursos siguientes.

- [Etiquetado de recursos de Athena](#)
- [Activación de etiquetas de asignación de costos generadas por AWS](#)
- [Uso de etiquetas de asignación de costos de AWS](#)

## Creación de reservas de capacidad

Para empezar, cree una reserva de capacidad con el número de DPU que necesite y, a continuación, asigne uno o más grupos de trabajo que utilizarán esa capacidad para sus consultas. Puede ajustar su capacidad más adelante según sea necesario para ofrecer un rendimiento más uniforme o administrar mejor los costos. Para obtener información sobre cómo calcular sus requisitos de capacidad, consulte [Determinación de los requisitos de capacidad](#).

### Important

Las solicitudes de capacidad no están garantizadas y se pueden completar en hasta 30 minutos.

Para crear una reserva de capacidad

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.
3. Elija Administración, Reservas de capacidad.

4. Elija Crear reserva de capacidad.
5. En la página Crear reserva de capacidad, en Nombre de reserva de capacidad, introduzca el nombre. El nombre debe ser único, de 1 a 128 caracteres de longitud y usar solo los caracteres a-z, A-Z, 0-9, \_ (guion de subrayado), . (punto) y - (guion). No se puede cambiar el nombre después de crear la reserva.
6. Para DPU, elija o introduzca la cantidad de unidades de procesamiento de datos (DPU) que desee en incrementos de 4. Para obtener más información, consulte [Descripción de las DPU](#).
7. (Opcional) Amplíe la opción Etiquetas y, a continuación, elija Agregar nueva etiqueta para agregar uno o más pares de clave/valor personalizados y asociarlos al recurso de reserva de capacidad. Para obtener más información, consulte [Etiquetado de recursos de Athena](#).
8. Elija Revisar.
9. Cuando aparezca el mensaje Confirmar creación de reserva de capacidad, confirme la cantidad de DPU, Región de AWS y demás información. Si acepta, seleccione Enviar.

En la página de detalles, el estado de la reserva de capacidad aparece como Pendiente. Cuando la capacidad de reserva está disponible para ejecutar consultas, su estado se muestra como Activa.

En este punto, ya está listo para agregar uno o más grupos de trabajo a su reserva. Para ver los pasos, consulte [Cómo agregar grupos de trabajo a una reserva](#).

## Administración de reservas

Puede ver y administrar sus reservas de capacidad en la página Reservas de capacidad. Puede realizar tareas de administración, como agregar o reducir las DPU, modificar las asignaciones de los grupos de trabajo y etiquetar o cancelar las reservas.

Para ver y administrar reservas de capacidad

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.
3. Elija Administración, Reservas de capacidad.
4. En la página de reservas de capacidad, puede realizar las siguientes tareas:
  - Para [crear](#) una reserva de capacidad, seleccione Crear reserva de capacidad.

- Utilice el cuadro de búsqueda para filtrar las reservas por nombre o cantidad de DPU.
- Seleccione el menú desplegable de estado para filtrar por estado de reserva de capacidad (por ejemplo, Activa o Cancelada). Para obtener más información acerca del estado de las reservas, consulte [Información sobre el estado de las reservas](#).
- Para ver los detalles de una reserva de capacidad, seleccione el enlace de la reserva. La página de detalles de la reserva incluye opciones para [editar la capacidad](#), [agregar grupos de trabajo](#), [eliminar grupos de trabajo](#) y [cancelar](#) la reserva.
- Para editar una reserva (por ejemplo, mediante la adición o eliminación de DPU), seleccione el botón de la reserva y, a continuación, seleccione Editar.
- Para cancelar una reserva, seleccione el botón de la reserva y, a continuación, seleccione Cancelar.

### Información sobre el estado de las reservas

En la siguiente tabla, se describen los valores de estado que una reserva de capacidad puede tener.

Estado	Descripción
Pendiente	Athena está procesando su solicitud de capacidad. La capacidad no está lista para ejecutar consultas.
Activo	Hay capacidad disponible para ejecutar consultas.
Con error	Su solicitud de capacidad no se completó correctamente. Tenga en cuenta que no se garantiza el cumplimiento de las solicitudes de capacidad. Las reservas fallidas se tienen en cuenta para los límites de DPU de su cuenta. Para liberar el uso, debe cancelar la reserva.
Actualización pendiente	Athena está procesando un cambio en la reserva. Por ejemplo, este estado se produce después de editar la reserva a fin de agregar o eliminar DPU.
Cancelling	Athena está procesando una solicitud para cancelar la reserva. Las consultas que aún se estén ejecutando en los grupos de trabajo que utilizaban la reserva se pueden finalizar, pero las demás consultas del grupo de trabajo utilizarán la capacidad bajo demanda (no aprovisio nada).



Estado	Descripción
Cancelado	<p>La cancelación de la reserva de capacidad se completó. Las reservas canceladas permanecen en la consola durante 45 días. Después de 45 días, Athena eliminará la reserva. Durante los 45 días, no podrá readaptar ni reutilizar la reserva, pero podrá consultar las etiquetas y ver los detalles como referencia histórica.</p> <p>No se garantiza que la capacidad cancelada se pueda volver a reservar en una fecha futura. La capacidad no se puede transferir a otra reserva, Cuenta de AWS o Región de AWS.</p>

### Descripción de las DPU activas y DPU objetivo

En la lista de reservas de capacidad de la consola de Athena, su reserva muestra dos valores de DPU: DPU activa y DPU objetivo.

- **DPU activa:** la cantidad de DPU disponibles en su reserva para ejecutar consultas. Por ejemplo, si solicita 100 DPU y se completa su solicitud, la DPU activa muestra 100.
- **DPU objetivo:** la cantidad de DPU a las que se está trasladando su reserva. La DPU objetivo muestra un valor diferente al de la DPU activa cuando se crea una reserva o cuando está pendiente un aumento o una disminución de la cantidad de DPU.

Por ejemplo, después de enviar una solicitud para crear una reserva con 24 DPU, el estado de la reserva será Pendiente, la DPU activa será 0 y la DPU objetivo será 24.

Si tiene una reserva con 100 DPU y la edita para solicitar un aumento de 20 DPU, el estado será Actualización pendiente, la DPU activa será 100 y la DPU objetivo será 120.

Si tiene una reserva con 100 DPU y la edita para solicitar una reducción de 20 DPU, el estado será Actualización pendiente, la DPU activa será 100 y la DPU objetivo será 80.

Durante estas transiciones, Athena trabaja activamente para adquirir o reducir la cantidad de DPU en función de su solicitud. Cuando la DPU activa se iguala a la DPU objetivo, se ha alcanzado el número objetivo y no hay cambios pendientes.

Para recuperar estos valores mediante programación, puede llamar a la acción de la API [GetCapacityReservation](#). La API hace referencia a la DPU activa y la DPU objetivo como `AllocatedDpus` y `TargetDpus`.

## Temas

- [Edición de reservas de capacidad](#)
- [Cómo agregar grupos de trabajo a una reserva](#)
- [Eliminación de un grupo de trabajo de una reserva](#)
- [Cancelación de una reserva de capacidad](#)
- [Eliminación de una reserva de capacidad](#)

## Edición de reservas de capacidad

Tras crear una reserva de capacidad, puede ajustar la cantidad de DPU y agregar o eliminar sus etiquetas personalizadas.

Para editar una reserva de capacidad

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.
3. Elija Administración, Reservas de capacidad.
4. En la lista de reservas de capacidad, realice una de las siguientes acciones:
  - Seleccione el botón situado junto a la reserva y, a continuación, elija Editar.
  - Elija el enlace a la reserva y, a continuación, elija Editar.
5. Para DPU, elija o introduzca la cantidad de unidades de procesamiento de datos que desee en incrementos de 4. La cantidad mínima de DPU que puede tener es 24. Para obtener más información, consulte [Descripción de las DPU](#).

### Note

Puede agregar las DPU a una reserva de capacidad existente en cualquier momento. Sin embargo, no puede reducir la cantidad de DPU hasta 1 hora después de crear la reserva o de agregarle DPU.

6. (Opcional) En Etiquetas, seleccione Eliminar para eliminar una etiqueta o Agregar nueva etiqueta para agregar una nueva etiqueta.
7. Elija Enviar. La página de detalles de la reserva muestra la configuración actualizada.

### Cómo agregar grupos de trabajo a una reserva

Tras crear una reserva de capacidad, puede agregar hasta 20 grupos de trabajo a la reserva. Al agregar un grupo de trabajo a una reserva, Athena indica qué consultas debe ejecutar en la capacidad reservada. Las consultas de los grupos de trabajo que no están asociados a una reserva siguen ejecutándose según el modelo de precios predeterminado por terabyte (TB) analizado.

Cuando una reserva tiene dos o más grupos de trabajo, las consultas de esos grupos de trabajo pueden utilizar la capacidad de la reserva. Puede agregar y eliminar grupos de trabajo en cualquier momento. Al agregar o eliminar grupos de trabajo, las consultas que se están ejecutando no se interrumpen.

Cuando la reserva esté en estado pendiente, las consultas de los grupos de trabajo que haya agregado seguirán ejecutándose según el modelo de precios predeterminado por terabyte (TB) analizado hasta que la reserva se active.

### Para agregar uno o más grupos de trabajo a la reserva de capacidad

1. En la página de detalles de la reserva de capacidad, seleccione Agregar grupos de trabajo.
2. En la página Agregar grupos de trabajo, seleccione los grupos de trabajo que desee agregar y, a continuación, elija Agregar grupos de trabajo. No puede asignar un grupo de trabajo a más de una reserva.

La página de detalles de la reserva de capacidad muestra los grupos de trabajo que ha agregado. Las consultas que se ejecuten en esos grupos de trabajo utilizarán la capacidad que haya reservado cuando la reserva esté activa.

### Eliminación de un grupo de trabajo de una reserva

Si ya no necesita capacidad dedicada para un grupo de trabajo o desea mover un grupo de trabajo a su propia reserva, puede eliminarla en cualquier momento. Eliminar un grupo de trabajo de una reserva es un proceso sencillo. Tras eliminar un grupo de trabajo de una reserva, las consultas del grupo de trabajo eliminado utilizan de forma predeterminada la capacidad bajo demanda (no aprovisionada) y se facturan en función de los terabytes (TB) analizados.

## Para eliminar uno o más grupos de trabajo de una reserva

1. En la página de detalles de la reserva de capacidad, seleccione los grupos de trabajo que desee eliminar.
2. Seleccione Eliminar grupos de trabajo. El mensaje ¿Desea eliminar los grupos de trabajo? le informa que todas las consultas actualmente activas finalizarán antes de eliminar el grupo de trabajo de la reserva.
3. Elija Eliminar. La página de detalles de la reserva de capacidad muestra que los grupos de trabajo eliminados ya no están presentes.

## Cancelación de una reserva de capacidad

Si ya no quiere usar una reserva de capacidad, puede cancelarla. Las consultas que aún se estén ejecutando en los grupos de trabajo que utilizaban la reserva se podrán finalizar, pero las demás consultas del grupo de trabajo ya no utilizarán la reserva.

### Note

No se garantiza que la capacidad cancelada se pueda volver a reservar en una fecha futura. La capacidad no se puede transferir a otra reserva, Cuenta de AWS o Región de AWS.

## Para cancelar una reserva de capacidad

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.
3. Elija Administración, Reservas de capacidad.
4. En la lista de reservas de capacidad, realice una de las siguientes acciones:
  - Seleccione el botón situado junto a la reserva y, a continuación, elija Cancelar.
  - Elija el enlace a la reserva y, a continuación, elija Cancelar reserva de capacidad.
5. En el mensaje ¿Desea cancelar la reserva de capacidad?, introduzca cancelar y, a continuación, seleccione Cancelar reserva de capacidad.

El estado de la reserva cambia a Cancelando y un cartel de progreso le informa que la cancelación está en curso.

Cuando se complete la cancelación, la reserva de capacidad se mantendrá, pero su estado aparecerá como Cancelada. La reserva se eliminará 45 días después de la cancelación. Durante los 45 días, no podrá readaptar ni reutilizar una reserva que se ha cancelado, pero podrá consultar las etiquetas y ver los detalles como referencia histórica.

## Eliminación de una reserva de capacidad

Si desea eliminar todas las referencias a una reserva de capacidad cancelada, puede eliminar la reserva. Se debe cancelar una reserva antes de que esta pueda eliminarse. Una reserva eliminada se elimina inmediatamente de su cuenta y ya no se puede invocar, ni siquiera mediante su ARN.

### Para eliminar una reserva de capacidad

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.
3. Elija Administración, Reservas de capacidad.
4. En la lista de reservas de capacidad, realice una de las siguientes acciones:
  - Seleccione el botón situado junto a la reserva cancelada y, a continuación, elija Acciones, Eliminar.
  - Elija el enlace a la reserva y, a continuación, elija Eliminar.
5. En el mensaje ¿Desea eliminar la reserva de capacidad?, seleccione Eliminar.

Aparecerá un mensaje en el que se le informa que la reserva de capacidad se ha eliminado correctamente. La reserva eliminada ya no aparecerá en la lista de reservas de capacidad.

## Políticas de IAM para reservas de capacidad

Para controlar el acceso a las reservas de capacidad, utilice permisos de IAM de nivel de recursos o políticas de IAM basadas en identidad. Siempre que utilice políticas de IAM, compruebe que sigue las mejores prácticas IAM. Para obtener más información, consulte la sección [Prácticas recomendadas de seguridad de IAM](#) en la Guía del usuario de IAM.

El siguiente procedimiento es específico de Athena.

Para obtener información específica sobre IAM, consulte los enlaces que se enumeran al final de esta sección. Para obtener información sobre las políticas de reservas de capacidad JSON de ejemplo, consulte [Ejemplos de políticas de reserva de capacidad](#).

Para utilizar el editor visual en la consola de IAM para crear una política de reserva de capacidad

1. Inicie sesión en la AWS Management Console y abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación de la izquierda, elija Políticas y, a continuación, elija Crear política.
3. En la pestaña Editor visual, elija Elegir un servicio. A continuación, elija un servicio de Athena para agregar a la política.
4. Elija Seleccionar acciones y, a continuación, elija las acciones que desea añadir a la política. El editor visual muestra las acciones disponibles en Athena. Para obtener más información, consulte [Acciones, recursos y claves de condición de Amazon Athena](#) en la Referencia de autorizaciones de servicio.
5. Elija agregar acciones para escribir una acción específica o utilice caracteres comodín (\*) para especificar varias acciones.

De forma predeterminada, la política que está creando permite las acciones que usted elija. Si eligió una o más acciones que admiten permisos en el nivel de recursos para el recurso `capacity-reservation` en Athena, el editor visual enumera el recurso `capacity-reservation`.

6. Elija Recursos a fin de detallar las reservas de capacidad específicas para su política. Para obtener ejemplos de políticas de reservas de capacidad JSON, consulte [Ejemplos de políticas de reserva de capacidad](#).
7. Especifique el recurso `capacity-reservation` como se indica a continuación:

```
arn:aws:athena:<region>:<user-account>:capacity-reservation/<capacity-reservation-name>
```

8. Elija Review policy (Revisar la política) y, a continuación, escriba un Name (Nombre) y una Description (Descripción) (opcional) para la política que está creando. Revise el resumen de la política para asegurarse de que ha concedido los permisos deseados.
9. Elija Create Policy (Crear política) para guardar la nueva política.
10. Adjunte esta política basada en identidad a un usuario, grupo o rol.

Para obtener más información, consulte los siguientes temas en la Referencia de autorizaciones de servicio y la Guía del usuario de IAM:

- [Acciones, recursos y claves de condición de Amazon Athena](#)
- [Creación de políticas con el editor visual](#)
- [Adición y eliminación de políticas de IAM](#)
- [Control del acceso a los recursos](#)

Para obtener ejemplos de políticas de reservas de capacidad JSON, consulte [Ejemplos de políticas de reserva de capacidad](#).

Para obtener una lista completa de las acciones de Amazon Athena, consulte los nombres de acciones de la API en la [Referencia de API de Amazon Athena](#).

### Ejemplos de políticas de reserva de capacidad

En esta sección se incluyen ejemplos de políticas que puede utilizar para habilitar varias acciones en las reservas de capacidad. Siempre que utilice políticas de IAM, compruebe que sigue las mejores prácticas IAM. Para más información, consulte [Prácticas recomendadas de seguridad en IAM](#) en la Guía del usuario de IAM.

Una reserva de capacidad es un recurso de IAM administrado por Athena. Por lo tanto, si la política de reserva de capacidad utiliza acciones que toman `capacity-reservation` como entrada, debe especificar el ARN de la reserva de capacidad de la siguiente manera:

```
"Resource": [arn:aws:athena:<region>:<user-account>:capacity-reservation/<capacity-reservation-name>]
```

`<capacity-reservation-name>` es el nombre de su reserva de capacidad. Por ejemplo, para una reserva de capacidad denominada `test_capacity_reservation`, especifíquela como un recurso de la siguiente manera:

```
"Resource": ["arn:aws:athena:us-east-1:123456789012:capacity-reservation/test_capacity_reservation"]
```

Para obtener una lista completa de las acciones de Amazon Athena, consulte los nombres de acciones de la API en la [Referencia de API de Amazon Athena](#). Para obtener más información sobre las políticas de IAM, consulte [Creación de políticas con el editor visual](#) en la Guía del usuario de IAM.

- [Example policy to list capacity reservations](#)
- [Example policy for management operations](#)

### Example Ejemplo de política para enumerar las reservas de capacidad

La siguiente política permite a todos los usuarios enumerar todas las reservas de capacidad.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:ListCapacityReservations"
      ],
      "Resource": "*"
    }
  ]
}
```

### Example Ejemplo de política para operaciones de administración

La siguiente política permite a un usuario crear, cancelar, obtener detalles y actualizar la reserva de capacidad `test_capacity_reservation`. La política también permite a un usuario asignar `workgroupA` y `workgroupB` a `test_capacity_reservation`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:CreateCapacityReservation",
        "athena:GetCapacityReservation",
        "athena:CancelCapacityReservation",
        "athena:UpdateCapacityReservation",
        "athena:GetCapacityAssignmentConfiguration",
        "athena:PutCapacityAssignmentConfiguration"
      ],
      "Resource": [
        "arn:aws:athena:us-east-1:123456789012:capacity-
reservation/test_capacity_reservation",

```



```
        "arn:aws:athena:us-east-1:123456789012:workgroup/workgroupA",  
        "arn:aws:athena:us-east-1:123456789012:workgroup/workgroupB"  
    ]  
}  
]  
}
```

## API de reserva de capacidad de Athena

La siguiente lista contiene enlaces de referencia a las acciones de la API de reserva de capacidad. Para ver las estructuras de datos y otras acciones de la API de Athena, consulte [Amazon Athena API Reference](#) (Referencia de la API de Amazon Athena).

- [CancelCapacityReservation](#)
- [CreateCapacityReservation](#)
- [GetCapacityAssignmentConfiguration](#)
- [GetCapacityReservation](#)
- [ListCapacityReservations](#)
- [PutCapacityAssignmentConfiguration](#)
- [UpdateCapacityReservation](#)

## Ajuste del rendimiento en Athena

En este tema se proporciona información general y sugerencias específicas para mejorar el rendimiento de las consultas de Athena y para solucionar los errores relacionados con los límites y el uso de recursos.

### Service Quotas

Athena aplica cuotas para métricas como el tiempo de ejecución de las consultas, el número de consultas simultáneas en una cuenta y las tasas de solicitudes de API. Para obtener más información acerca de estas cuotas, consulte [Service Quotas](#). Si se superan estas cuotas, se produce un error en una consulta, ya sea al enviarla o durante la ejecución.

Muchos de los consejos de optimización del rendimiento de esta página pueden ayudar a reducir el tiempo de ejecución de las consultas. La optimización libera capacidad para que pueda ejecutar más consultas dentro de la cuota de simultaneidad y evita que las consultas se cancelen por ejecutarse durante demasiado tiempo.

Las cuotas del número de consultas y solicitudes de API simultáneas son por Cuenta de AWS y Región de AWS. Recomendamos ejecutar una carga de trabajo por Cuenta de AWS (o utilizar reservas de capacidad aprovisionadas independientes) para evitar que las cargas de trabajo compitan por la misma cuota.

Si se ejecutan dos cargas de trabajo en la misma cuenta, una de las cargas de trabajo puede ejecutar una ráfaga de consultas. Esto puede provocar que la carga de trabajo restante se limite o impida la ejecución de consultas. Para evitarlo, puede mover las cargas de trabajo a cuentas independientes a fin de asignar a cada carga de trabajo su propia cuota de simultaneidad. Al crear una reserva de capacidad aprovisionada para una o ambas cargas de trabajo, se logra el mismo objetivo.

### Cuotas en otros servicios

Cuando Athena ejecuta una consulta, puede llamar a otros servicios que imponen cuotas. Durante la ejecución de la consulta, Athena puede realizar llamadas a la API de AWS Glue Data Catalog, Amazon S3 y otros servicios de AWS, como IAM y AWS KMS. Si se utilizan [consultas federadas](#), Athena también llama a AWS Lambda. Todos estos servicios tienen sus propios límites y cuotas que pueden superarse. Cuando la ejecución de una consulta detecta errores de estos servicios, se produce un error e incluye el error del servicio de origen. Los errores recuperables se vuelven a intentar, pero las consultas pueden seguir fallando si el problema no se resuelve a tiempo. Asegúrese de leer detenidamente los mensajes de error para determinar si provienen de Athena o de otro servicio. En este documento se describen algunos de los errores más relevantes.

Para obtener más información sobre cómo solucionar los errores relacionados con las cuotas de servicio de Amazon S3, consulte [Cómo evitar tener demasiados archivos](#) más adelante en este documento. Para obtener más información sobre la optimización del rendimiento de Amazon S3, consulte [Prácticas recomendadas para patrones de diseño: optimizar el rendimiento de Amazon S3](#) en la Guía del usuario de Amazon S3.

### Límites de recursos

Athena ejecuta las consultas en un motor de consultas distribuido. Al enviar una consulta, el planificador de consultas del motor de Athena calcula la capacidad de procesamiento necesaria para ejecutar la consulta y prepara un clúster de nodos de computación en consecuencia. Algunas consultas, como las consultas DDL, se ejecutan en un solo nodo. Las consultas complejas sobre conjuntos de datos de gran tamaño se ejecutan en clústeres mucho más grandes. Los nodos son uniformes y tienen las mismas configuraciones de disco, CPU y memoria. Athena escala horizontalmente, no verticalmente, para procesar consultas más exigentes.

A veces, las demandas de una consulta superan los recursos disponibles para el clúster que ejecuta la consulta. Cuando esto ocurre, la consulta genera el error La consulta agotó los recursos con este factor de escala.

El recurso que se agota con más frecuencia es la memoria, pero en raras ocasiones también puede ser el espacio en disco. Los errores de memoria suelen producirse cuando el motor ejecuta una función de unión o ventana, pero también pueden producirse en distintos recuentos y agregaciones.

Aunque una consulta falle una vez y muestre el error de “falta de recursos”, es posible que se ejecute correctamente cuando se vuelva a ejecutar. La ejecución de la consulta no es determinista. Factores como el tiempo que se tarda en cargar los datos y la forma en que se distribuyen los conjuntos de datos intermedios en los nodos pueden provocar un uso diferente de los recursos. Imagine una consulta que une dos tablas y presenta un gran sesgo en la distribución de los valores de la condición de unión. Una consulta de este tipo puede funcionar correctamente la mayoría de las veces, pero en ocasiones falla cuando los valores más comunes terminan siendo procesados por el mismo nodo.

Para evitar que las consultas superen los recursos disponibles, utilice los consejos de ajuste de rendimiento que se mencionan en este documento. Para obtener consejos sobre cómo optimizar las consultas que agotan los recursos disponibles, consulte [Optimización de uniones](#), [Optimización de las funciones de ventana](#) y [Optimización de las consultas mediante aproximaciones](#).

## Técnicas de optimización de consultas

Utilice las técnicas de optimización de consultas que se describen en esta sección para hacer que las consultas se ejecuten más rápido o como soluciones para las consultas que superan los límites de recursos en Athena.

### Optimización de uniones

Existen muchas estrategias diferentes para ejecutar uniones en un motor de consultas distribuido. Dos de las más comunes son las uniones hash distribuidas y las consultas con condiciones de unión complejas.

#### Uniones hash distribuidas

El tipo de unión más común utiliza una comparación de igualdad como condición de unión. Athena ejecuta este tipo de unión como una unión hash distribuida.

En una unión hash distribuida, el motor crea una tabla de consulta (tabla hash) a partir de uno de los lados de la unión. Este lado se denomina lado de compilación. Los registros del lado de compilación

se distribuyen entre los nodos. Cada nodo crea una tabla de búsqueda para su subconjunto. El otro lado de la unión, denominado lado de sondeo, se transmite a través de los nodos. Los registros del lado de sondeo se distribuyen entre los nodos de la misma manera que en el lado de compilación. Esto permite que cada nodo realice la unión buscando los registros coincidentes en su propia tabla de búsqueda.

Cuando las tablas de búsqueda creadas a partir del lado de compilación de la unión no caben en la memoria, las consultas pueden fallar. Aun si el tamaño total del lado de compilación es inferior a la memoria disponible, las consultas pueden fallar si la distribución de los registros presenta un sesgo significativo. En un caso extremo, todos los registros podrían tener el mismo valor para la condición de unión y tener que caber en la memoria de un único nodo. Incluso una consulta con menos asimetría puede fallar si se envía un conjunto de valores al mismo nodo y los valores suman más que la memoria disponible. Los nodos tienen la capacidad de almacenar registros en el disco, pero esto ralentiza la ejecución de la consulta y puede que no sea suficiente para evitar que la consulta falle.

Athena intenta reordenar las uniones para usar la relación más grande como el lado de sondeo y la relación más pequeña como el lado de compilación. Sin embargo, como Athena no gestiona los datos de las tablas, tiene información limitada y, a menudo, debe suponer que la primera tabla es la más grande y la segunda es la más pequeña.

Al escribir uniones con condiciones de unión basadas en la igualdad, suponga que la tabla situada a la izquierda de la palabra clave JOIN corresponde al lado de sondeo y la tabla a la derecha corresponde al lado de compilación. Asegúrese de que la tabla correcta, la del lado de compilación, sea la más pequeña de las tablas. Si no es posible hacer que el lado de compilación de la unión sea lo suficientemente pequeño como para caber en la memoria, considere la posibilidad de ejecutar varias consultas que unan subconjuntos de la tabla de compilación.

## Otros tipos de uniones

Las consultas con condiciones de unión complejas (por ejemplo, las consultas que utilizan LIKE, > u otros operadores) suelen ser exigentes desde el punto de vista computacional. En el peor de los casos, todos los registros de un lado de la unión deben compararse con todos los registros del otro lado de la unión. Como el tiempo de ejecución aumenta con el cuadrado del número de registros, estas consultas corren el riesgo de superar el tiempo máximo de ejecución.

Para saber de antemano cómo Athena ejecutará su consulta, puede usar la instrucción EXPLAIN. Para obtener más información, consulte [Uso de EXPLAIN y EXPLAIN ANALYZE en Athena](#) y [Descripción de los resultados de la instrucción EXPLAIN de Athena](#).

## Optimización de las funciones de ventana

Como las funciones de ventana son operaciones que consumen muchos recursos, pueden hacer que las consultas se ejecuten con lentitud o incluso que se produzcan errores con el mensaje La consulta agotó los recursos con este factor de escala. Las funciones de ventana guardan en la memoria todos los registros en los que operan para calcular su resultado. Cuando la ventana es muy grande, la función de ventana puede quedarse sin memoria.

Para asegurarse de que las consultas se ejecuten dentro de los límites de memoria disponibles, reduzca el tamaño de las ventanas sobre las que trabajan las funciones de ventana. Para ello, puede agregar una cláusula `PARTITIONED BY` o reducir el alcance de las cláusulas de partición existentes.

## Uso de funciones que no sean de ventana

A veces, las consultas con funciones de ventana se pueden reescribir sin funciones de ventana. Por ejemplo, en lugar de utilizar `row_number` para buscar los registros de N principales, puede utilizar `ORDER BY` y `LIMIT`. En lugar de usar `row_number` o `rank` para desduplicar registros, puede usar funciones de agregado como [max\\_by](#), [min\\_by](#) y [arbitrary](#).

Por ejemplo, supongamos que tiene un conjunto de datos con actualizaciones de un sensor. El sensor informa periódicamente el estado de la batería e incluye algunos metadatos, como la ubicación. Si desea saber el último estado de la batería de cada sensor y su ubicación, puede usar la siguiente consulta:

```
SELECT sensor_id,
       arbitrary(location) AS location,
       max_by(battery_status, updated_at) AS battery_status
FROM sensor_readings
GROUP BY sensor_id
```

Los metadatos, como la ubicación, son los mismos para todos los registros, por lo que puede utilizar la función `arbitrary` a fin de seleccionar cualquier valor del grupo.

Para obtener el último estado de la batería, puede usar la función `max_by`. La función `max_by` selecciona el valor de una columna del registro en el que se encontró el valor máximo de otra columna. En este caso, devuelve el estado de la batería del registro con la hora de la última actualización del grupo. Esta consulta se ejecuta más rápido y utiliza menos memoria que una consulta equivalente con una función de ventana.

## Optimización de las agregaciones

Cuando Athena realiza una agregación, distribuye los registros entre los nodos de trabajo mediante las columnas de la cláusula `GROUP BY`. Para que la tarea de hacer coincidir los registros con los grupos sea lo más eficiente posible, los nodos intentan mantener los registros en la memoria, pero los transfieren al disco si es necesario.

También es una buena idea evitar incluir columnas redundantes en las cláusulas `GROUP BY`. Dado que un menor número de columnas requiere menos memoria, resulta más eficaz una consulta que describa un grupo con menos columnas. Las columnas numéricas también utilizan menos memoria que las cadenas. Por ejemplo, al agregar un conjunto de datos que tiene un ID de categoría numérico y un nombre de categoría, utilice únicamente la columna de ID de categoría en la cláusula `GROUP BY`.

A veces, las consultas incluyen columnas en la cláusula `GROUP BY` para evitar el hecho de que una columna sea parte de la cláusula `GROUP BY` o una expresión agregada. Si no se sigue esta regla, puede producirse un error con el siguiente mensaje:

```
EXPRESSION_NOT_AGGREGATE: line 1:8: 'category' must be an aggregate expression or appear in GROUP BY clause (EXPRESSION_NOT_AGGREGATE: línea 1:8: "categoría" debe ser una expresión agregada o aparecer en la cláusula GROUP BY).
```

Para evitar tener que agregar columnas redundantes a la cláusula `GROUP BY`, puede utilizar la función [arbitrary](#), como en el siguiente ejemplo.

```
SELECT country_id,  
       arbitrary(country_name) AS country_name,  
       COUNT(*) AS city_count  
FROM world_cities  
GROUP BY country_id
```

La función `ARBITRARY` devuelve un valor arbitrario del grupo. La función resulta útil cuando se sabe que todos los registros del grupo tienen el mismo valor para una columna, pero el valor no identifica al grupo.

## Optimización de las consultas N principales

La cláusula `ORDER BY` devuelve los resultados de una consulta ordenados. Athena usa la clasificación distribuida para ejecutar la operación de clasificación en paralelo en varios nodos.

Si no necesita estrictamente ordenar el resultado, evite agregar una cláusula `ORDER BY`. Además, evite agregar `ORDER BY` a las consultas internas si no son estrictamente necesarias. En muchos casos, el planificador de consultas puede eliminar la ordenación redundante, pero esto no está garantizado. Una excepción a esta regla es si una consulta interna está realizando una operación `N` principal, como buscar los valores `N` más recientes o los `N` más comunes.

Cuando Athena ve `ORDER BY` junto con `LIMIT`, entiende que se está ejecutando una consulta `N` principal y utiliza operaciones dedicadas en consecuencia.

#### Note

Aunque Athena también suele detectar funciones de ventana como `row_number` que usan `N` principales, recomendamos la versión más sencilla que usa `ORDER BY` y `LIMIT`. Para obtener más información, consulte [Optimización de las funciones de ventana](#).

## Inclusión de las columnas necesarias únicamente

Si no necesita estrictamente una columna, no la incluya en la consulta. Cuantos menos datos tenga que procesar una consulta, más rápido se ejecutará. Esto reduce tanto la cantidad de memoria necesaria como la cantidad de datos que deben enviarse entre los nodos. Si utiliza un formato de archivo en columnas, al reducir el número de columnas también se reduce la cantidad de datos que se leen desde Amazon S3.

Athena no tiene un límite específico para el número de columnas de un resultado, pero la forma en que se ejecutan las consultas limita el tamaño combinado de las columnas posible. El tamaño combinado de las columnas incluye los nombres y tipos.

Por ejemplo, el siguiente error se debe a una relación que supera el límite de tamaño de un descriptor de relación:

```
GENERIC_INTERNAL_ERROR: io.airlift.bytecode.CompilationException
```

Para solucionar este problema, reduzca el número de columnas de la consulta o cree subconsultas y utilice una cláusula `JOIN` que recupere una cantidad menor de datos. Si tiene consultas que aplican `SELECT *` en la consulta más externa, debe cambiar `*` a una lista de solo las columnas que necesita.

## Optimización de las consultas mediante aproximaciones

Athena admite [funciones agregadas de aproximación](#) para contar valores distintos, los valores más frecuentes, los percentiles (incluidas las medianas aproximadas) y crear histogramas. Utilice estas funciones siempre que no necesite valores exactos.

A diferencia de las operaciones `COUNT(DISTINCT col)`, [approx\\_distinct](#) utiliza mucha menos memoria y se ejecuta más rápido. Del mismo modo, si se utiliza [numeric\\_histogram](#) en lugar de [histogram](#), se utilizan métodos aproximados y, por lo tanto, se utiliza menos memoria.

## Optimización de LIKE

Puede usar LIKE para encontrar cadenas coincidentes, pero con cadenas largas, esto requiere un uso intensivo de cómputos. La función [regexp\\_like](#) es, en la mayoría de los casos, una alternativa más rápida que proporciona más flexibilidad.

A menudo, puede optimizar una búsqueda anclando la subcadena que está buscando. Por ejemplo, si busca un prefijo, es mucho mejor usar “*substr%*” en lugar de “%*substr%*”. O bien, si está usando `regexp_like`, “*^substr*”.

## Uso de UNION ALL en lugar de UNION

UNION ALL y UNION son dos formas de combinar los resultados de dos consultas en un solo resultado. UNION ALL concatena los registros de la primera consulta con la segunda y UNION hace lo mismo, pero también elimina los duplicados. UNION necesita procesar todos los registros y encontrar los duplicados, lo que requiere mucha memoria y procesamiento, pero UNION ALL es una operación relativamente rápida. A menos que necesite deduplicar registros, use UNION ALL para obtener el mejor rendimiento.

## Uso de UNLOAD para conjuntos de resultados grandes

Si se espera que los resultados de una consulta sean grandes (por ejemplo, decenas de miles de filas o más), utilice UNLOAD para exportar los resultados. En la mayoría de los casos, esto resulta más rápido que ejecutar una consulta normal y, además, usar UNLOAD permite tener más control sobre el resultado.

Cuando termina de ejecutarse una consulta, Athena almacena el resultado como un único archivo CSV sin comprimir en Amazon S3. Esto lleva más tiempo que UNLOAD, no solo porque el resultado no está comprimido, sino también porque la operación no se puede paralelizar. Por el contrario, UNLOAD escribe los resultados directamente desde los nodos de trabajo y aprovecha al máximo



el paralelismo del clúster de computación. Además, puede configurar UNLOAD para escribir los resultados en formato comprimido y en otros formatos de archivo, como JSON y Parquet.

Para obtener más información, consulte [UNLOAD](#).

### Uso de CTAS o ETL de Glue para materializar las agregaciones de uso frecuente

La “materialización” de una consulta es una forma de acelerar el rendimiento de la consulta mediante el almacenamiento de resultados de consultas complejas precalculados (por ejemplo, agregaciones y uniones) para reutilizarlos en consultas posteriores.

Si varias de las consultas incluyen las mismas uniones y agregaciones, puede materializar la subconsulta común como una tabla nueva y, a continuación, ejecutar las consultas en esa tabla. Puede crear la nueva tabla con [Creación de una tabla a partir de los resultados de una consulta \(CTAS\)](#) o una herramienta ETL específica, como [ETL de Glue](#).

Por ejemplo, supongamos que tiene un panel con widgets que muestran diferentes aspectos de un conjunto de datos de pedidos. Cada widget tiene su propia consulta, pero todas las consultas comparten las mismas uniones y filtros. Una tabla de pedidos se une a una tabla de líneas de artículos y hay un filtro que muestra solo los últimos tres meses. Si se identifican las características comunes de estas consultas, se puede crear una tabla nueva que los widgets puedan usar. Esto reduce la duplicación y mejora el rendimiento. La desventaja es que se debe mantener la nueva tabla actualizada.

### Reutilización de resultados de las consultas

Es habitual que la misma consulta se ejecute varias veces en poco tiempo. Por ejemplo, esto puede ocurrir cuando varias personas abren el mismo panel de datos. Al ejecutar una consulta, puede decirle a Athena que reutilice los resultados calculados con anterioridad. Usted especifica la antigüedad máxima de los resultados que se van a reutilizar. Si la misma consulta se ejecutó anteriormente dentro de ese periodo de tiempo, Athena devuelve esos resultados en lugar de volver a ejecutar la consulta. Para obtener más información, consulte [Reutilización de los resultados de las consultas](#) en la Guía del usuario de Amazon Athena y [Reducir los costos y mejorar el rendimiento de las consultas con la reutilización de resultados de las consultas de Amazon Athena](#) en el blog sobre macrodatos de AWS.

### Técnicas de optimización de datos

El rendimiento no solo depende de las consultas, sino también en gran medida de cómo esté organizado el conjunto de datos y del formato de archivo y de la compresión que utilice.

## Partición de datos

La creación de particiones divide la tabla en partes y mantiene los datos relacionados juntos de acuerdo con propiedades como la fecha, el país o la región. Las claves de partición actúan como columnas virtuales. Las claves de partición se definen al crear la tabla y se utilizan para filtrar las consultas. Al filtrar las columnas de claves de partición, solo se leen los datos de las particiones coincidentes. Por ejemplo, si el conjunto de datos está dividido por fecha y la consulta tiene un filtro que solo coincide con la última semana, solo se leerán los datos de la última semana. Para obtener más información sobre la creación de particiones, consulte [Particiones de datos en Athena](#).

### Elección de claves de partición compatibles con las consultas

Dado que la creación de particiones tiene un impacto significativo en el rendimiento de las consultas, asegúrese de considerar cuidadosamente cómo crear las particiones al diseñar el conjunto de datos y las tablas. Tener demasiadas claves de partición puede provocar conjuntos de datos fragmentados con demasiados archivos y archivos demasiado pequeños. Por el contrario, tener muy pocas claves de partición, o no tener ninguna partición, hace que las consultas analicen más datos de los necesarios.

### Cómo evitar optimizar consultas poco frecuentes

Una buena estrategia consiste en optimizar las consultas más comunes y evitar la optimización para las consultas poco frecuentes. Por ejemplo, si sus consultas están basadas en intervalos de días, no las divida por hora, incluso si algunas consultas se filtran a ese nivel. Si sus datos tienen una columna de marca de tiempo detallada, las consultas poco frecuentes que se filtran por hora pueden usar la columna de marca de tiempo. Incluso si en los casos poco frecuentes se analizan un poco más de datos de los necesarios, reducir el rendimiento general en aras de los casos excepcionales no suele ser una buena compensación.

Para reducir la cantidad de datos que se deben analizar en las consultas y, por lo tanto, mejorar el rendimiento, utilice un formato de archivo en columnas y mantenga los registros ordenados. En lugar de particionar por hora, mantenga los registros ordenados por marca de tiempo. Para las consultas en intervalos de tiempo más cortos, ordenar por marca de tiempo es casi tan eficaz como particionar por hora. Además, la clasificación por marca de tiempo no suele afectar el rendimiento de las consultas en intervalos de tiempo contados en días. Para obtener más información, consulte [Utilizar formatos de archivo en columnas](#).

Tenga en cuenta que las consultas en tablas con decenas de miles de particiones funcionan mejor si hay predicados en todas las claves de partición. Esta es otra razón para diseñar un esquema de

particiones para las consultas más comunes. Para obtener más información, consulte [Consulta de las particiones por igualdad](#).

## Uso de la proyección de particiones

La proyección de particiones es una característica de Athena que no almacena la información de la partición en el AWS Glue Data Catalog, sino como reglas en las propiedades de la tabla en AWS Glue. Cuando Athena planifica una consulta en una tabla configurada con proyección de particiones, lee las reglas de proyección de particiones de la tabla. Athena calcula las particiones para leerlas en la memoria en función de la consulta y las reglas en lugar de buscar las particiones en el AWS Glue Data Catalog.

Además de simplificar la administración de particiones, la proyección de particiones puede mejorar el rendimiento de los conjuntos de datos que tienen un gran número de particiones. Cuando una consulta incluye intervalos en lugar de valores específicos para las claves de partición, la búsqueda de particiones coincidentes en el catálogo lleva más tiempo a medida que aumenta el número de particiones. Con la proyección de particiones, el filtro se puede calcular en la memoria sin tener que consultar el catálogo, lo que puede ser mucho más rápido.

En determinadas circunstancias, la proyección de particiones puede reducir el rendimiento. Un ejemplo ocurre cuando una tabla se encuentra “dispersa”. Una tabla dispersa no contiene datos para todas las permutaciones de los valores de clave de partición descritos en la configuración de proyección de particiones. Con una tabla dispersa, el conjunto de particiones calculadas a partir de la consulta y la configuración de proyección de particiones se muestran en Amazon S3, incluso cuando no contienen datos.

Cuando utilice la proyección de particiones, asegúrese de incluir los predicados en todas las claves de partición. Limite el alcance de los valores posibles para evitar operaciones de lista innecesarias de Amazon S3. Imagine una clave de partición que tiene un intervalo de un millón de valores y una consulta que no tiene ningún filtro en esa clave de partición. Para ejecutar la consulta, Athena debe realizar al menos un millón de operaciones de lista en Amazon S3. Las consultas son más rápidas cuando se consultan valores específicos, independientemente de si se utiliza la proyección de particiones o se almacena la información de las particiones en el catálogo. Para obtener más información, consulte [Consulta de las particiones por igualdad](#).

Al configurar una tabla para la proyección de particiones, asegúrese de que los intervalos que especifique sean razonables. Si una consulta no incluye un predicado en una clave de partición, se utilizan todos los valores del intervalo de esa clave. Si el conjunto de datos se creó en una fecha específica, utilice esa fecha como punto de partida para cualquier intervalo de fechas. Utilice NOW

como el final de los intervalos de fechas. Evite los intervalos numéricos que tengan un gran número de valores y considere usar el tipo [inyectado](#) en su lugar.

Para obtener más información sobre la proyección de particiones, consulte [Proyección de particiones con Amazon Athena](#).

### Uso de índices de particiones

Los índices de particiones son una característica del AWS Glue Data Catalog que mejora el rendimiento de la búsqueda de particiones en las tablas que tienen un gran número de particiones.

La lista de particiones del catálogo es como una tabla de una base de datos relacional. La tabla tiene las columnas para las claves de partición y una columna adicional para la ubicación de la partición. Al consultar una tabla particionada, las ubicaciones de las particiones se buscan mediante el análisis de esta tabla.

Al igual que con las bases de datos relacionales, puede aumentar el rendimiento de las consultas gracias a la adición de índices. Puede agregar varios índices para admitir diferentes patrones de consulta. El índice de particiones del AWS Glue Data Catalog admite operadores de igualdad y comparación, como  $>$ ,  $>=$  y  $<$  combinados con el operador AND. Para obtener más información, consulte [Trabajo con índices de partición en AWS Glue](#) en la Guía para desarrolladores de AWS Glue y [Mejora del rendimiento de consultas de Amazon Athena con índices de particiones de AWS Glue Data Catalog](#) en el blog sobre macrodatos de AWS.

### Uso siempre de STRING como tipo para las claves de partición

Cuando consulte las claves de partición, recuerde que Athena requiere que las claves de partición sean del tipo STRING para poder aplicar el filtrado de particiones en AWS Glue. Si el número de particiones no es pequeño, el uso de otros tipos puede reducir el rendimiento. Si los valores de las claves de partición son similares a una fecha o a un número, cámbielos al tipo adecuado en la consulta.

### Eliminación de las particiones antiguas y vacías

Si elimina datos de una partición de Amazon S3 (por ejemplo, mediante el [ciclo de vida](#) de Amazon S3), también debe eliminar la entrada de la partición del AWS Glue Data Catalog. Durante la planificación de la consulta, cualquier partición que coincida con la consulta aparece en Amazon S3. Si tiene muchas particiones vacías, la sobrecarga que supone listarlas puede ser perjudicial.

Además, si tiene muchos miles de particiones, considere la posibilidad de eliminar los metadatos de las particiones de los datos antiguos que ya no son relevantes. Por ejemplo, si las consultas nunca

analizan datos de más de un año, puede eliminar periódicamente los metadatos de las particiones más antiguas. Si el número de particiones aumenta a decenas de miles, eliminar las particiones no utilizadas puede acelerar las consultas que no incluyen predicados en todas las claves de partición. Para obtener información sobre cómo incluir predicados en todas las claves de partición en las consultas, consulte [Consulta de las particiones por igualdad](#).

### Consulta de las particiones por igualdad

Las consultas que incluyen predicados de igualdad en todas las claves de partición se ejecutan más rápido porque los metadatos de las particiones se pueden cargar de forma directa. Evite las consultas en las que una o más claves de partición no tengan un predicado o en las que el predicado seleccione un intervalo de valores. En este tipo de consultas, se debe filtrar la lista de todas las particiones para encontrar valores coincidentes. En la mayoría de las tablas, la sobrecarga es mínima, pero en las tablas con decenas de miles o más particiones, la sobrecarga puede llegar a ser significativa.

Si no es posible reescribir las consultas para filtrar las particiones por igualdad, puede probar la proyección de particiones. Para obtener más información, consulte [Uso de la proyección de particiones](#).

### Cómo evitar el uso de MSCK REPAIR TABLE para el mantenimiento de las particiones

Como MSCK REPAIR TABLE puede tardar mucho en ejecutarse, solo agrega particiones nuevas y no elimina las antiguas, no es una forma eficaz de administrar las particiones (consulte [Consideraciones y limitaciones](#)).

Las particiones se administran mejor de forma manual mediante las [API del AWS Glue Data Catalog](#), [ALTER TABLE ADD PARTITION](#) o los [rastreadores de AWS Glue](#). Como alternativa, puede utilizar la proyección de particiones, que elimina por completo la necesidad de administrar las particiones. Para obtener más información, consulte [Proyección de particiones con Amazon Athena](#).

### Comprobar que las consultas sean compatibles con el esquema de particiones

Puede comprobar de antemano qué particiones se analizarán en una consulta mediante la instrucción [EXPLAIN](#). Agregue el prefijo a la consulta con la palabra clave EXPLAIN y, a continuación, busque el fragmento de origen (por ejemplo, `Fragment 2 [SOURCE]`) de cada tabla cerca de la parte inferior del resultado EXPLAIN. Busque tareas en las que el lado derecho esté definido como una clave de partición. La línea inferior incluye una lista de todos los valores de esa clave de partición que se analizarán cuando se ejecute la consulta.

Por ejemplo, supongamos que tiene una consulta en una tabla con una clave de partición `dt` y agrega el prefijo a la consulta con `EXPLAIN`. Si los valores de la consulta son fechas y con un filtro se selecciona un intervalo de tres días, el resultado `EXPLAIN` puede ser similar a lo siguiente:

```
dt := dt:string:PARTITION_KEY
    :: [[2023-06-11], [2023-06-12], [2023-06-13]]
```

El resultado `EXPLAIN` muestra que el planificador encontró tres valores para esta clave de partición que coincidían con la consulta. También muestra cuáles son esos valores. Para obtener más información sobre el uso de `EXPLAIN`, consulte [Uso de EXPLAIN y EXPLAIN ANALYZE en Athena](#) y [Descripción de los resultados de la instrucción EXPLAIN de Athena](#).

### Utilizar formatos de archivo en columnas

Los formatos de archivo en columnas, como Parquet y ORC, están diseñados para cargas de trabajo de análisis distribuidas. Organizan los datos por columnas, no por filas. La organización de los datos en formato de columnas ofrece las siguientes ventajas:

- Solo se cargan las columnas necesarias para la consulta.
- Se reduce la cantidad total de datos que deben cargarse.
- Los valores de las columnas se almacenan juntos, por lo que los datos se pueden comprimir de forma eficiente.
- Los archivos pueden contener metadatos que permiten que el motor omita la carga de datos innecesarios.

Como ejemplo de cómo se pueden utilizar los metadatos de los archivos, los metadatos de los archivos pueden contener información sobre los valores mínimos y máximos de una página de datos. Si los valores consultados no están dentro del intervalo indicado en los metadatos, se puede omitir la página.

Una forma de utilizar estos metadatos para mejorar el rendimiento es garantizar que los datos de los archivos estén ordenados. Por ejemplo, supongamos que tiene consultas que buscan registros en los que la entrada `created_at` se encuentra dentro de un periodo de tiempo breve. Si los datos están ordenados por la columna `created_at`, Athena puede usar los valores mínimo y máximo de los metadatos del archivo para omitir las partes innecesarias de los archivos de datos.

Cuando utilice formatos de archivo en columnas, asegúrese de que los archivos no sean demasiado pequeños. Como se indica en [Cómo evitar tener demasiados archivos](#), los conjuntos de datos con

muchos archivos pequeños generan problemas de rendimiento. Esto se aplica aún más cuando se habla de formatos de archivo en columnas. En el caso de los archivos pequeños, la sobrecarga del formato de archivo en columnas supera las ventajas.

Tenga en cuenta que Parquet y ORC están organizados internamente por grupos de filas (Parquet) y bandas (ORC). El tamaño predeterminado para los grupos de filas es de 128 MB y para las bandas, de 64 MB. Si tiene muchas columnas, puede aumentar el tamaño del grupo de filas y de las bandas para obtener un mejor rendimiento. No se recomienda reducir el tamaño del grupo de filas o de las bandas a un valor inferior a sus valores predeterminados.

Para convertir otros formatos de datos a Parquet u ORC, puede utilizar ETL de AWS Glue o Athena. Para obtener más información acerca del uso de Athena para ETL, consulte [Uso de CTAS e INSERT INTO en ETL y análisis de datos](#).

## Compresión de datos

Athena admite una amplia gama de formatos de compresión. La consulta de datos comprimidos es más rápida y económica, ya que se paga por el número de bytes analizados antes de la descompresión.

El formato [gzip](#) ofrece buenas relaciones de compresión y es compatible con una amplia gama de herramientas y servicios. El formato [zstd](#) (Zstandard) es un formato de compresión más reciente con un buen equilibrio entre rendimiento y relación de compresión.

Al comprimir archivos de texto, como datos JSON y CSV, intente lograr un equilibrio entre el número de archivos y su tamaño. La mayoría de los formatos de compresión requieren que el lector lea los archivos desde el principio. Esto significa que, en general, los archivos de texto comprimidos no se pueden procesar en paralelo. Los archivos grandes sin comprimir suelen dividirse entre los trabajos para lograr un mayor paralelismo durante el procesamiento de las consultas, pero esto no es posible con la mayoría de los formatos de compresión.

Como se explica en [Cómo evitar tener demasiados archivos](#), lo mejor es no tener demasiados archivos ni muy pocos. Como el número de archivos es el límite del número de trabajos que se pueden procesar en la consulta, esta regla se aplica aún más a los archivos comprimidos.

Para obtener más información sobre el uso de la compresión en Athena, consulte [Compatibilidad con la compresión de Athena](#).

## Creación de buckets para buscar claves con cardinalidad alta

La agrupación en buckets es una técnica para distribuir los registros en archivos separados según el valor de una de las columnas. Esto garantiza que todos los registros con el mismo valor estén en el mismo archivo. La agrupación en buckets resulta útil cuando se tiene una clave con una cardinalidad alta y muchas de las consultas buscan valores específicos de la clave.

Por ejemplo, supongamos que consulta un conjunto de registros para un usuario específico. Si los datos están agrupados por ID de usuario, Athena sabe de antemano qué archivos contienen registros para un ID específico y cuáles no. Esto permite a Athena leer solo los archivos que pueden contener el ID, lo que reduce de forma significativa la cantidad de datos leídos. También reduce el tiempo de computación que, de otro modo, se necesitaría para buscar el ID específico en los datos.

### Desventajas de agrupar en buckets

La agrupación en buckets es menos valiosa cuando las consultas buscan con frecuencia múltiples valores en la columna por la que están agrupados los datos. Cuantos más valores se consulten, mayor será la probabilidad de que se tengan que leer todos los archivos o la mayoría de ellos. Por ejemplo, si tiene tres buckets y una consulta busca tres valores diferentes, es posible que deban leerse todos los archivos. La agrupación en buckets funciona mejor cuando las consultas buscan valores únicos.

Para obtener más información, consulte [Creación de particiones y asignación de buckets en Athena](#).

### Cómo evitar tener demasiados archivos

Los conjuntos de datos que consisten en muchos archivos pequeños dan como resultado un rendimiento general de las consultas deficiente. Cuando Athena planifica una consulta, lista todas las ubicaciones de las particiones, lo que lleva tiempo. La administración y solicitud de cada archivo también supone una sobrecarga computacional. Por lo tanto, cargar un solo archivo más grande desde Amazon S3 es más rápido que cargar los mismos registros desde muchos archivos más pequeños.

En casos extremos, es posible que se encuentre con límites de servicio de Amazon S3. Amazon S3 admite hasta 5500 solicitudes por segundo en una sola partición de índice. Inicialmente, un bucket se trata como una partición de índice único, pero a medida que aumentan las cargas de solicitudes, se puede dividir en varias particiones de índice.

Amazon S3 analiza los patrones de solicitudes y los divide en función de los prefijos clave. Si su conjunto de datos consiste en muchos miles de archivos, las solicitudes procedentes de Athena



pueden superar la cuota de solicitudes. Incluso con menos archivos, se puede superar la cuota si se realizan varias consultas simultáneas en el mismo conjunto de datos. Otras aplicaciones que tengan acceso a los mismos archivos pueden contribuir al número total de solicitudes.

Cuando se supera la tasa de solicitudes `limit`, Amazon S3 muestra el siguiente error. Este error se incluye en la información de estado de la consulta en Athena.

SlowDown: reduzca la tasa de solicitudes.

Para solucionar el problema, comience por determinar si el error se debe a una sola consulta o a varias consultas que leen los mismos archivos. Si se debe a lo último, coordine la ejecución de las consultas para que no se ejecuten al mismo tiempo. Para lograrlo, agregue un mecanismo de colas o incluso de reintentos en su aplicación.

Si al ejecutar una sola consulta se desencadena el error, intente combinar archivos de datos o modificar la consulta para leer menos archivos. El mejor momento para combinar archivos pequeños es antes de escribirlos. Para ello, tenga en cuenta las siguientes técnicas:

- Cambie el proceso de escritura de los archivos para escribir archivos más grandes. Por ejemplo, puede almacenar los registros en búfer durante más tiempo antes de que se escriban.
- Coloque los archivos en una ubicación de Amazon S3 y utilice una herramienta como ETL de Glue para combinarlos en archivos más grandes. Luego, mueva los archivos más grandes a la ubicación a la que apunta la tabla. Para obtener más información, consulte [Lectura de archivos de entrada en grupos más grandes](#) en la Guía para desarrolladores de AWS Glue y [¿Cómo puedo configurar un trabajo de ETL en AWS Glue para generar archivos más grandes?](#) en el Centro de conocimientos de AWS re:Post.
- Reduzca el número de claves de partición. Si tiene demasiadas claves de partición, es posible que cada partición tenga solo algunos registros, lo que se traduce en un número excesivo de archivos pequeños. Para obtener información sobre cómo decidir qué particiones crear, consulte [Elección de claves de partición compatibles con las consultas](#).

### Cómo evitar jerarquías de almacenamiento adicionales más allá de la partición

Para evitar la sobrecarga de planificación de consultas, almacene los archivos en una estructura plana en cada ubicación de partición. No utilice jerarquías de directorios adicionales.

Cuando Athena planifica una consulta, lista todos los archivos de todas las particiones que coinciden con la consulta. Aunque Amazon S3 no tiene directorios propiamente dichos, la convención

consiste en interpretar la barra diagonal / como un separador de directorios. Cuando Athena lista las ubicaciones de las particiones, lista de forma recursiva cualquier directorio que encuentre. Cuando los archivos de una partición se organizan en una jerarquía, se producen varias rondas de operaciones de lista.

Cuando todos los archivos están directamente en la ubicación de la partición, la mayoría de las veces solo se debe realizar una operación de lista. Sin embargo, se requieren varias operaciones de lista secuencial si tiene más de 1000 archivos en una partición, ya que Amazon S3 devuelve solo 1000 objetos por operación de lista. Tener más de 1000 archivos en una partición también puede provocar otros problemas de rendimiento más graves. Para obtener más información, consulte [Cómo evitar tener demasiados archivos](#).

Uso de `SymlinkTextInputFormat` solo cuando sea necesario

El uso de la técnica [SymlinkTextInputFormat](#) puede ser una forma de evitar situaciones en las que los archivos de una tabla no estén bien organizados en particiones. Por ejemplo, los enlaces simbólicos pueden resultar útiles cuando todos los archivos tienen el mismo prefijo o si los archivos con esquemas diferentes se encuentran en la misma ubicación.

Sin embargo, el uso de enlaces simbólicos agrega niveles de indirección a la ejecución de la consulta. Estos niveles de indirección afectan el rendimiento general. Se deben leer los archivos de enlace simbólico y se deben listar las ubicaciones que definen. Esto agrega varios viajes de ida y vuelta a Amazon S3 que las tablas de Hive habituales no requieren. En conclusión, `SymlinkTextInputFormat` solo debe utilizarse cuando no haya mejores opciones disponibles, como la reorganización de archivos.

## Recursos adicionales de

Para obtener información adicional acerca del ajuste de rendimiento en Athena, tenga en cuenta los siguientes recursos:

- Lea la publicación del blog de macrodatos de AWS [Los 10 consejos principales para el ajuste del rendimiento de Amazon Athena](#).
- Para obtener información sobre cómo usar la inserción de predicados a fin de mejorar el rendimiento en las consultas federadas, consulte el artículo [Mejorar las consultas federadas con la inserción de predicados en Amazon Athena](#) en el blog sobre macrodatos de AWS.
- Para ver un artículo sobre las optimizaciones del rendimiento en el motor de consultas Athena, consulte [Ejecutar consultas 3 veces más rápido con un ahorro de costos de hasta un 70 % en el último motor de Amazon Athena](#) en el blog de macrodatos de AWS.

- Lea otras [publicaciones sobre Athena en el blog de macrodatos de AWS](#).
- Haga una pregunta en [AWS re:Post](#) con la etiqueta Amazon Athena
- Consulte los [temas sobre Athena en el Centro de conocimientos de AWS](#).
- Póngase en contacto con AWS Support (en la AWS Management Console, haga clic en Soporte, Centro de asistencia).

## Evitar la limitación de Amazon S3

La limitación es el proceso de limitar la velocidad a la que se utiliza un servicio, una aplicación o un sistema. En AWS, se puede utilizar la limitación para evitar el uso excesivo del servicio Amazon S3 y aumentar la disponibilidad y la capacidad de respuesta de Amazon S3 para todos los usuarios. Sin embargo, dado que la limitación limita la velocidad a la que se pueden transferir los datos hacia Amazon S3 o desde este, es importante considerar la posibilidad de evitar que se limiten las interacciones.

### Reduzca las limitaciones a nivel de servicio

Para evitar que Amazon S3 se limite a nivel de servicio, puede supervisar el uso y ajustar las [cuotas de servicio](#), o bien utilizar determinadas técnicas, como las particiones. A continuación, se detallan algunas de las condiciones que pueden provocar la limitación:

- Superar los límites de solicitudes de la API de su cuenta: Amazon S3 tiene límites de solicitudes de API predeterminados que se basan en el tipo de cuenta y el uso. Si supera el número máximo de solicitudes por segundo para un solo objeto, es posible que sus solicitudes se limiten para evitar la sobrecarga del servicio Amazon S3.
- Partición insuficiente de los datos: si no particiona correctamente los datos y transfiere una gran cantidad de datos, Amazon S3 puede limitar sus solicitudes. Para obtener más información sobre las particiones, consulte la sección [Utilice particiones](#) mencionada en este documento.
- Gran cantidad de objetos pequeños: si es posible, evite tener una gran cantidad de archivos pequeños. Amazon S3 tiene un límite de [5500 solicitudes GET](#) por segundo por prefijo particionado y las consultas de Athena tienen este mismo límite. Si analiza millones de objetos pequeños en una sola consulta, Amazon S3 puede limitar la consulta.

Para evitar un análisis excesivo, utilice ETL de AWS Glue para compactar periódicamente los archivos o particionar la tabla y agregar filtros de clave de partición. Para obtener más información, consulte los siguientes recursos.

- [¿Cómo puedo configurar un trabajo de ETL en AWS Glue para generar archivos más grandes?](#) (Centro de conocimientos de AWS)
- [Lectura de archivos de entrada en grupos más grandes](#) (Guía para desarrolladores de AWS Glue)

## Optimice las tablas

Si tiene problemas de limitación, es importante que estructure los datos. Si bien Amazon S3 puede gestionar grandes cantidades de datos, a veces se produce una limitación debido a la forma en que están estructurados los datos.

En las siguientes secciones se ofrecen algunas sugerencias sobre cómo estructurar los datos en Amazon S3 para evitar problemas de limitación.

## Utilice particiones

Puede utilizar las particiones para reducir la limitación al limitar la cantidad de datos a los que se puede acceder en cualquier momento. Al dividir los datos en columnas específicas, puede distribuir las solicitudes de manera uniforme entre varios objetos y reducir el número de solicitudes de un solo objeto. Al reducir la cantidad de datos que se deben analizar, el rendimiento de las consultas mejora y los costos se reducen.

Al crear una tabla, puede definir particiones, que actúan como columnas virtuales. Para crear una tabla con particiones en una instrucción `CREATE TABLE`, utilice la cláusula `PARTITIONED BY` (*column\_name data\_type*) a fin de definir las claves para dividir los datos.

Para restringir las particiones analizadas en una consulta, puede especificarlas como predicados en una cláusula `WHERE` de la consulta. De esta manera, las columnas que se utilizan con frecuencia como filtros son buenas candidatas para la creación de particiones. Una práctica común consiste en particionar los datos en función de intervalos de tiempo, lo que puede dar lugar a un esquema de particiones de varios niveles.

Tenga en cuenta que la creación de particiones también tiene un costo. Al aumentar el número de particiones de la tabla, también aumenta el tiempo necesario para recuperar y procesar los metadatos de las particiones. Por lo tanto, la sobrepartición puede eliminar los beneficios que se obtienen al particionar con un mejor criterio. Si sus datos están muy sesgados hacia un valor de partición y la mayoría de las consultas utilizan ese valor, es posible que incurra en una sobrecarga adicional.

Para obtener más información sobre la creación de particiones en Athena, consulte [¿Qué es la creación de particiones?](#).

## Organice los datos en buckets

Otra forma de particionar los datos consiste en organizarlos en buckets en una sola partición. Al organizar los datos en buckets, especifica una o más columnas que contienen las filas que desea agrupar. A continuación, pone esas filas en varios buckets. De esta forma, solo consulta el bucket que debe leerse, lo que reduce el número de filas de datos que deben analizarse.

Al seleccionar una columna para utilizarla en los buckets, seleccione la columna que tenga una cardinalidad alta (es decir, que tenga muchos valores distintos), que esté distribuida de manera uniforme y que se utilice con frecuencia para filtrar los datos. Un ejemplo de una buena columna que se puede usar para organizar datos en buckets es una clave principal, como una columna de ID.

Para obtener más información acerca de la asignación de buckets en Athena, consulte [¿Qué es la asignación de buckets?](#).

## Utilice índices de particiones de AWS Glue

Puede usar índices de partición de AWS Glue para organizar los datos de una tabla en función de los valores de una o más particiones. Los índices de partición de AWS Glue pueden reducir el número de transferencias de datos, la cantidad de procesamiento de datos y el tiempo de procesamiento de las consultas.

Un índice de particiones de AWS Glue es un archivo de metadatos que contiene información sobre las particiones de la tabla, incluidas las claves de partición y sus valores. El índice de particiones se almacena en un bucket de Amazon S3 y AWS Glue lo actualiza automáticamente a medida que se agregan nuevas particiones a la tabla.

Cuando un índice de particiones de AWS Glue está presente, las consultas pueden recuperar un subconjunto de las particiones en lugar de cargar todas las particiones de la tabla. Las consultas solo se ejecutan en el subconjunto de datos que es relevante para la consulta.

Al crear una tabla en AWS Glue, puede crear un índice de particiones en cualquier combinación de claves de partición definidas en la tabla. Tras crear uno o más índices de partición en una tabla, debe agregar una propiedad a la tabla que permita el filtrado de particiones. A continuación, puede consultar la tabla desde Athena.

Para obtener información acerca de cómo crear índices de particiones en AWS Glue, consulte [Trabajar con índices de partición en AWS Glue](#) en la Guía para desarrolladores de AWS Glue. Para obtener información sobre cómo agregar una propiedad de tabla para habilitar el filtrado de particiones, consulte [Indexación y filtrado de particiones de AWS Glue](#).

## Utilice la compresión de datos y la división de archivos

La compresión de datos puede acelerar de forma significativa las consultas si los archivos tienen el tamaño óptimo o si se pueden dividir en grupos lógicos. Por lo general, las relaciones de compresión más altas requieren más ciclos de CPU para comprimir y descomprimir los datos. Para Athena, se recomienda utilizar Apache Parquet o Apache ORC, que comprimen los datos de forma predeterminada. Para obtener información sobre la compresión de datos en Athena, consulte [Compatibilidad con la compresión de Athena](#).

La división de archivos aumenta el paralelismo al permitir que Athena distribuya la tarea de leer un solo archivo entre varios lectores. Si un único archivo no se puede dividir, solo un único lector puede leerlo mientras los demás lectores permanecen inactivos. Apache Parquet y Apache ORC también admiten la división de archivos.

## Utilice almacenes de datos en columnas optimizados

El rendimiento de las consultas de Athena mejora de forma significativa si convierte los datos a un formato de columnas. Al generar archivos en columnas, una técnica de optimización a tener en cuenta es ordenar los datos en función de la clave de partición.

Apache Parquet y Apache ORC son almacenes de datos en columnas de código abierto cuyo uso está generalizado. Para obtener información sobre cómo convertir un origen de datos de Amazon S3 existente a uno de estos formatos, consulte [Conversión a formatos de columnas](#).

## Utilice un tamaño de bloque de Parquet o de banda de ORC más grande

Parquet y ORC tienen parámetros de almacenamiento de datos que puede ajustar para su optimización. En Parquet, puede optimizar el tamaño del bloque. En ORC, puede optimizar el tamaño de las bandas. Cuanto más grande sea el bloque o la banda, más filas podrá almacenar en cada uno. De forma predeterminada, el tamaño del bloque de Parquet es de 128 MB y el tamaño de la banda de ORC es de 64 MB.

Si una banda de ORC ocupa menos de 8 MB (el valor predeterminado de `hive.orc.max_buffer_size`), Athena lee toda la banda de ORC. Esta es la compensación que Athena hace entre la selectividad de columnas y las operaciones de entrada/salida por segundo para bandas más pequeñas.

Si tiene tablas con un gran número de columnas, un tamaño de bloque o banda pequeño puede provocar que se analicen más datos de los necesarios. En estos casos, un tamaño de bloque más grande puede ser más eficiente.

## Utilice ORC para tipos complejos

Actualmente, cuando se consultan columnas almacenadas en Parquet con tipos de datos complejos (por ejemplo, `array`, `map` o `struct`), Athena lee toda una fila de datos, en lugar de leer solo de manera selectiva las columnas especificadas. Este es un problema conocido en Athena. Como solución, considere usar ORC.

## Elija un algoritmo de compresión

Otro parámetro que puede configurar es el algoritmo de compresión de los bloques de datos. Para obtener información sobre los algoritmos de compresión compatibles con Parquet y ORC en Athena, consulte [Compatibilidad con la compresión de Athena](#).

Para obtener más información sobre la optimización de los formatos de almacenamiento en columnas en Athena, consulte la sección “Optimizar la generación de almacenes de datos en columnas” en la publicación del Blog de macrodatos de AWS [Top 10 Performance Tuning Tips for Amazon Athena](#) (Los 10 mejores consejos para ajustar el rendimiento de Amazon Athena).

## Utilice tablas de Iceberg

Apache Iceberg es un formato de tabla de código abierto para conjuntos de datos analíticos de gran tamaño diseñado para el uso optimizado en Amazon S3. Puede usar tablas de Iceberg para ayudar a reducir las limitaciones en Amazon S3.

Las tablas de Iceberg ofrecen las siguientes ventajas:

- Puede dividir las tablas de Iceberg en una o más columnas. Esto optimiza el acceso a los datos y reduce la cantidad de datos que se deben analizar en las consultas.
- Como el modo de almacenamiento de objetos de Iceberg optimiza las tablas de Iceberg para que funcionen con Amazon S3, puede procesar grandes volúmenes de datos y cargas de trabajo de consultas pesadas.
- Las tablas de Iceberg en el modo de almacenamiento de objetos son escalables, tolerantes a errores y duraderas, lo que puede ayudar a reducir las limitaciones.
- La compatibilidad con transacciones ACID significa que varios usuarios pueden agregar y eliminar objetos de Amazon S3 de forma atómica.

Para obtener más información sobre Apache Iceberg, consulte [Apache Iceberg](#). Para obtener información acerca del uso de tablas de Apache Iceberg en Athena, consulte [Utilización de tablas de Iceberg](#).

## Optimización de consultas

Utilice las sugerencias de esta sección para optimizar sus consultas SQL en Athena.

### Utilice LIMIT con la cláusula ORDER BY

La cláusula ORDER BY devuelve los datos en un orden clasificado. Esto requiere que Athena envíe todas las filas de datos a un único nodo de trabajo y luego las ordene. Este tipo de consulta puede ejecutarse durante mucho tiempo o, incluso, fallar.

Para aumentar la eficacia de las consultas, observe los valores *N* en la parte superior o inferior y utilice también una cláusula LIMIT. Esto reduce de forma significativa el costo de la clasificación, ya que tanto la clasificación como la limitación recaen en nodos de trabajo individuales en lugar de en un solo trabajo.

### Optimice las cláusulas JOIN

Al unir dos tablas, Athena distribuye la tabla de la derecha entre los nodos de trabajo y luego incluye la tabla de la izquierda para realizar la unión.

Por este motivo, especifique la tabla más grande en el lado izquierdo de la unión y la tabla más pequeña en el lado derecho de la unión. De esta forma, Athena utiliza menos memoria y ejecuta la consulta con una latencia menor.

También tenga en cuenta los siguientes puntos:

- Cuando utilice varios comandos JOIN, especifique las tablas de mayor a menor.
- Evite las uniones cruzadas a menos que la consulta las requiera.

### Optimice las cláusulas GROUP BY

El operador GROUP BY distribuye las filas en función de las columnas GROUP BY a los nodos de trabajo. Se hace referencia a estas columnas en la memoria y los valores se comparan a medida que se ingieren las filas. Los valores se agregan cuando la columna GROUP BY coincide. Teniendo en cuenta la forma en que funciona este proceso, se recomienda ordenar las columnas desde la cardinalidad más alta hasta la más baja.

### Utilice números en lugar de cadenas

Como los números requieren menos memoria y son más rápidos de procesar en comparación con las cadenas, utilice números en lugar de cadenas siempre que sea posible.



## Limite el número de columnas

A fin de reducir la cantidad total de memoria necesaria para almacenar los datos, limite el número de columnas especificado en la instrucción SELECT.

## Utilice expresiones comunes en lugar de LIKE

Las consultas que incluyen cláusulas como LIKE '%string%' en cadenas grandes pueden ser muy exigentes desde el punto de vista computacional. Al filtrar varios valores en una columna de cadena, utilice la función [regexp\\_like\(\)](#) y una expresión común en su lugar. Esto resulta muy útil cuando se compara una lista larga de valores.

## Utilice la cláusula LIMIT

En lugar de seleccionar todas las columnas al ejecutar una consulta, utilice la cláusula LIMIT para devolver solo las columnas que necesite. Esto reduce el tamaño del conjunto de datos que se procesa a través de la canalización de ejecución de la consulta. Las cláusulas LIMIT son más útiles cuando se consultan tablas que tienen un gran número de columnas basadas en cadenas. Las cláusulas LIMIT también son útiles cuando se realizan múltiples uniones o agregaciones en cualquier consulta.

## Recursos adicionales de

[Prácticas recomendadas para patrones de diseño: optimizar el rendimiento de Amazon S3](#) en la Guía del usuario de Amazon Simple Storage Service.

## [Ajuste del rendimiento en Athena](#)

## Compatibilidad con la compresión de Athena

### Temas

- [Especificación de los formatos de compresión](#)
- [Especificación sin compresión](#)
- [Notas y recursos](#)
- [Compatibilidad de compresión de tablas de Hive por formato de archivo](#)
- [Compatibilidad de compresión de tablas de Iceberg por formato de archivo](#)
- [Uso de los niveles de compresión ZSTD en Athena](#)

Athena admite una variedad de formatos de compresión para leer y escribir datos, como la lectura de una tabla que utiliza varios formatos de compresión. Por ejemplo, Athena puede leer correctamente los datos de una tabla que utiliza el formato de archivo Parquet cuando algunos archivos Parquet se comprimen con Snappy y otros archivos Parquet se comprimen con GZIP. El mismo principio se aplica a los formatos de almacenamiento ORC, archivo de texto y JSON.

Athena es compatible con los siguientes formatos de compresión:

- BZIP2: formato que utiliza el algoritmo Burrows-Wheeler.
- DEFLATE: algoritmo de compresión basado en [LZSS](#) y [codificación Huffman](#). [Deflate](#) solo es relevante para el formato de archivo Avro.
- GZIP: algoritmo de compresión basado en Deflate. Para las tablas de Hive en las versiones 2 y 3 del motor Athena y las tablas de Iceberg en la versión 2 del motor Athena, GZIP es el formato de compresión de escritura predeterminado para los archivos con formato de almacenamiento de archivo de texto y Parquet. Los archivos con el formato `tar.gz` no son compatibles.
- LZ4: este miembro de la familia Lempel-Ziv 77 (LZ7) también se centra en la velocidad de compresión y descompresión en lugar de la máxima compresión de los datos. LZ4 tiene los siguientes formatos de encuadre:
  - LZ4 Raw/Unframed (LZ4 sin formato o sin encuadre): una implementación estándar y sin encuadre del formato de compresión de bloques LZ4. Para obtener más información, consulte [LZ4 block format description](#) (Descripción del formato de bloques LZ4) en GitHub.
  - LZ4 framed (LZ4 con encuadre): la implementación con el encuadre habitual de LZ4. Para obtener más información, consulte [LZ4 frame format description](#) (Descripción del formato de encuadre LZ4) en GitHub.
  - LZ4 hadoop-compatible (LZ4 compatible con Hadoop): la implementación con Apache Hadoop de LZ4. Esta implementación completa la compresión LZ4 con la clase [BlockCompressorStream.java](#).
- LZO: formato que utiliza el algoritmo Lempel-Ziv-Oberhumer, que se centra en la alta velocidad de compresión y descompresión en lugar de la compresión máxima de datos. LZO tiene dos implementaciones:
  - Standard LZO (LZO estándar): para obtener más información, consulte el [resumen](#) sobre LZO en el sitio web de Oberhumer.
  - LZO hadoop-compatible (LZO compatible con Hadoop): esta implementación completa el algoritmo de LZO con la clase [BlockCompressorStream.java](#).

- SNAPPY: algoritmo de compresión que forma parte de la familia Lempel-Ziv 77 (LZ7). Snappy se centra en la alta velocidad de compresión y descompresión en lugar de la máxima compresión de datos.
- ZLIB: basado en Deflate, ZLIB es el formato de compresión de escritura predeterminado para archivos con el formato de almacenamiento de datos ORC. Para obtener más información, consulte la página [zlib](#) en GitHub.
- ZSTD: el [algoritmo de compresión de datos en tiempo real Zstandard](#) es un algoritmo de compresión rápida que proporciona ratios altas de compresión. La biblioteca Zstandard (ZSTD) se proporciona como software de código abierto mediante una licencia BSD. ZSTD es la compresión predeterminada para las tablas de Iceberg. Cuando se escriben datos comprimidos con ZSTD, Athena utiliza el nivel 3 de compresión Zstandard. Para obtener información sobre el uso de los niveles de compresión ZSTD en Athena, consulte [Uso de los niveles de compresión ZSTD en Athena](#).

## Especificación de los formatos de compresión

Al escribir instrucciones CREATE TABLE o CTAS, puede especificar las propiedades de compresión que especifican el tipo de compresión que se utilizará cuando Athena escriba en esas tablas.

- Para CTAS, consulte [Propiedades de la tabla CTAS](#). Para ver ejemplos, consulte [Ejemplos de consultas CTAS](#).
- Para obtener información sobre CREATE TABLE, consulte [ALTER TABLE SET TBLPROPERTIES](#) para obtener una lista de las propiedades de la tabla de compresión.

## Especificación sin compresión

Las declaraciones CREATE TABLE permiten escribir archivos sin comprimir. Para escribir archivos sin comprimir, utilice la siguiente sintaxis:

- CREATE TABLE (archivo de texto o JSON): en TBLPROPERTIES, especifique `write.compression = NONE`.
- CREATE TABLE (Parquet): en TBLPROPERTIES, especifique `parquet.compression = UNCOMPRESSED`.
- CREATE TABLE (ORC): en TBLPROPERTIES, especifique `orc.compress = NONE`.

## Notas y recursos

- Actualmente, Athena no reconoce las extensiones de archivo en mayúsculas, como .GZ o .BZIP2. Evite utilizar conjuntos de datos con extensiones de archivo en mayúsculas o cambie el nombre de las extensiones de archivo de datos a minúsculas.
- Para datos en CSV, TSV y JSON, Athena determina el tipo de compresión a partir de la extensión de archivo. Si no hay ninguna extensión de archivo, Athena trata los datos como texto sin formato y sin comprimir. Si los datos están comprimidos, asegúrese de que el nombre de archivo incluye la extensión de compresión como, por ejemplo, gz.
- No se admite el formato de archivo ZIP.
- Para consultar registros de Amazon Data Firehose desde Athena, los formatos admitidos incluyen la compresión GZIP o archivos ORC con compresión SNAPPY.
- Para obtener más información sobre el uso de la compresión, consulte la sección 3, (“Comprimir y dividir archivos”), de la publicación del blog de macrodatos de AWS [Los 10 consejos más importantes sobre el ajuste de rendimiento de Amazon Athena](#).

## Compatibilidad de compresión de tablas de Hive por formato de archivo

La compatibilidad de compresión para Hive en Athena depende de la versión del motor.

### Compatibilidad de compresión para Hive en la versión 3 del motor Athena

En la siguiente tabla se resume la compatibilidad con formatos de compresión de la versión 3 del motor Athena para los formatos de archivo de almacenamiento de Apache Hive. El formato de archivo de texto incluye TSV, CSV, JSON y SerDes personalizado para texto. Los términos “Yes” (Sí) o “No” de una celda se aplican por igual a las operaciones de lectura y escritura, excepto donde se indique lo contrario. A los efectos de esta tabla, CREATE TABLE, CTAS e INSERT INTO se consideran operaciones de escritura. Para obtener información sobre el uso de los niveles de compresión ZSTD en Athena, consulte [Uso de los niveles de compresión ZSTD en Athena](#).

	Avro	Ion	ORC	Parquet	Archivo de texto
BZIP2	Sí	Sí	No	No	Sí
DEFLATE	Sí	No	No	No	No

	Avro	Ion	ORC	Parquet	Archivo de texto
GZIP	No	Sí	No	Sí	Sí
LZ4	No	Sí	Sí	Sí	Sí
LZO	No	Escritura: no Lectura: sí	No	Sí	Escritura: : no  Lectura: sí
SNAPPY	Sí	Sí	Sí	Sí	Sí
ZLIB	No	No	Sí	No	No
ZSTD	Sí	Sí	Sí	Sí	Sí
NONE	Sí	Sí	Sí	Sí	Sí

### Compatibilidad de compresión para Hive en la versión 2 del motor Athena

En la siguiente tabla se resume la compatibilidad con formatos de compresión de la versión 2 del motor Athena para Apache Hive. El formato de archivo de texto incluye TSV, CSV, JSON y SerDes personalizado para texto. Los términos “Yes” (Sí) o “No” de una celda se aplican por igual a las operaciones de lectura y escritura, excepto donde se indique lo contrario. A los efectos de esta tabla, CREATE TABLE, CTAS e INSERT INTO se consideran operaciones de escritura.

	Avro	Ion	ORC	Parquet	Archivo de texto
BZIP2	Sí	Sí	No	No	Sí
DEFLATE	Sí	No	No	No	No
GZIP	No	Sí	No	Sí	Sí

	Avro	Ion	ORC	Parquet	Archivo de texto
LZ4	No	No	Sí	Escritura: sí Lectura: no	Escritura: no Lectura: sí
LZO	No	Escritura: no Lectura: sí	No	Sí	Escritura: no Lectura: sí
SNAPPY	Sí	Sí	Sí	Sí	Sí
ZLIB	No	No	Sí	No	No
ZSTD	No	Sí	Sí	Sí	Sí
NONE	Sí	Sí	Sí	Sí	Sí

## Compatibilidad de compresión de tablas de Iceberg por formato de archivo

La compatibilidad de compresión para Apache Iceberg en Athena depende de la versión del motor.

### Compatibilidad de compresión para Iceberg en la versión 3 del motor Athena

En la siguiente tabla se resume la compatibilidad con formatos de compresión de la versión 3 del motor Athena para los formatos de archivo de almacenamiento de Apache Iceberg. Los términos “Yes” (Sí) o “No” de una celda se aplican por igual a las operaciones de lectura y escritura, excepto donde se indique lo contrario. A los efectos de esta tabla, CREATE TABLE, CTAS e INSERT INTO se consideran operaciones de escritura. El formato de almacenamiento predeterminado para Iceberg en la versión 3 del motor de Athena es Parquet. El formato de compresión predeterminado para Iceberg en la versión 3 del motor de Athena es ZSTD. Para obtener información sobre el uso de los niveles de compresión ZSTD en Athena, consulte [Uso de los niveles de compresión ZSTD en Athena](#).

	Avro	ORC	Parquet (predeterminado)
BZIP2	No	No	No
GZIP	Sí	No	Sí
LZ4	No	Sí	No
SNAPPY	Sí	Sí	Sí
ZLIB	No	Sí	No
ZSTD	Sí	Sí	Sí (predeterminado)
NONE	Sí (especifique None o Deflate)	Sí	Sí (especifique None o Uncompressed )

### Compatibilidad de compresión para Iceberg en la versión 2 del motor Athena

En la siguiente tabla se resume la compatibilidad con formatos de compresión de la versión 2 del motor Athena para Apache Iceberg. Los términos “Yes” (Sí) o “No” de una celda se aplican por igual a las operaciones de lectura y escritura, excepto donde se indique lo contrario. A los efectos de esta tabla, CREATE TABLE, CTAS e INSERT INTO se consideran operaciones de escritura. El formato de almacenamiento predeterminado para Iceberg en la versión 2 del motor de Athena es Parquet. El formato de compresión predeterminado para Iceberg en la versión 2 del motor de Athena es GZIP.

	Avro (No admitido)	ORC (No admitido)	Parquet (predeterminado)
BZIP2	No	No	No
GZIP	No	No	Sí (predeterminado)
LZ4	No	No	No
SNAPPY	No	No	Sí
ZLIB	No	No	No

	Avro (No admitido)	ORC (No admitido)	Parquet (predeterminado)
ZSTD	No	No	Sí
NONE	No	No	Sí

## Uso de los niveles de compresión ZSTD en Athena

El [algoritmo de compresión de datos en tiempo real Zstandard](#) es un algoritmo de compresión rápida que proporciona relaciones de compresión elevadas. La biblioteca Zstandard (ZSTD) es software de código abierto y utiliza una licencia BSD. Athena es compatible con la lectura y la escritura de datos de archivos de texto, Parquet y ORC comprimidos con ZSTD.

Puede utilizar los niveles de compresión ZSTD para ajustar la relación y velocidad de compresión de acuerdo con sus requisitos. La biblioteca ZSTD admite niveles de compresión comprendidos entre 1 y 22. Athena utiliza el nivel 3 de compresión ZSTD de manera predeterminada.

Los niveles de compresión permiten conseguir un equilibrio preciso entre la velocidad de compresión y la cantidad de compresión lograda. Los niveles de compresión más bajos proporcionan más velocidad, pero también tamaños de archivo más mayores. Por ejemplo, se puede utilizar el nivel 1 si la velocidad es la prioridad, y el nivel 22 si el tamaño es lo más importante. El nivel 3 es adecuado para muchos casos de uso, y es el predeterminado. Utilice los niveles superiores a 19 con precaución, ya que requieren más memoria. La biblioteca ZSTD también ofrece niveles de compresión negativos que amplían el intervalo de velocidades y relaciones de compresión. Para obtener más información, consulte el [documento RFC sobre compresión Zstandard](#).

La abundancia de niveles de compresión ofrece importantes posibilidades de realizar ajustes precisos. No obstante, asegúrese de medir los datos y tener en cuenta las ventajas y desventajas a la hora de decidir el nivel de compresión. Se recomienda utilizar el nivel predeterminado 3 o un nivel comprendido entre 6 y 9 para lograr un equilibrio razonable entre la velocidad de compresión y el tamaño de los datos comprimidos. Reserve los niveles de 20 y posteriores para aquellos casos en que el tamaño sea lo más importante y la velocidad de compresión no suponga un problema.

### Consideraciones y limitaciones

Cuando utilice los niveles de compresión ZSTD en Athena, tenga en cuenta los siguientes puntos.



- La propiedad `compression_level` de ZSTD solo es compatible con la versión 3 del motor Athena.
- La propiedad `compression_level` de ZSTD es compatible con las instrucciones `ALTER TABLE`, `CREATE TABLE` y `CREATE TABLE AS (CTAS)`, y `UNLOAD`.
- La propiedad `compression_level` es opcional.
- La propiedad `compression_level` solo es compatible con la compresión ZSTD.
- Los niveles de compresión posibles están comprendidos entre 1 y 22.
- El nivel de compresión predeterminado es 3.

Para obtener información sobre la compatibilidad de compresión para Apache Hive ZSTD en Athena, consulte [Compatibilidad de compresión de tablas de Hive por formato de archivo](#). Para obtener información sobre la compatibilidad de compresión para Apache Iceberg ZSTD en Athena, consulte [Compatibilidad de compresión de tablas de Iceberg por formato de archivo](#).

### Especificación de los niveles de compresión ZSTD

Para especificar el nivel de compresión ZSTD de las instrucciones `ALTER TABLE`, `CREATE TABLE`, `CREATE TABLE AS` y `UNLOAD`, utilice la propiedad `compression_level`. Para especificar la compresión ZSTD propiamente dicha, debe utilizar la propiedad de compresión individual que utilice la sintaxis de la instrucción.

### `ALTER TABLE SET TBLPROPERTIES`

En la cláusula `SET TBLPROPERTIES` de la instrucción [ALTER TABLE SET TBLPROPERTIES](#), especifique la compresión ZSTD mediante `'write.compression' = 'ZSTD'` o `'parquet.compression' = 'ZSTD'`. Después, utilice la propiedad `compression_level` para especificar un valor comprendido entre 1 y 22 (por ejemplo, `'compression_level' = 5`). Si no especifica ninguna propiedad de nivel de compresión, el nivel de compresión se establece en 3 de manera predeterminada.

### Ejemplo

En el siguiente ejemplo se modifica la tabla `existing_table` para que utilice el formato de archivo Parquet con compresión ZSTD y nivel 4 de compresión ZSTD. Tenga en cuenta que el valor del nivel de compresión debe introducirse como una cadena y no como un entero.

```
ALTER TABLE existing_table
SET TBLPROPERTIES ('parquet.compression' = 'ZSTD', 'compression_level' = 4)
```

## CREATE TABLE

En la cláusula TBLPROPERTIES de la instrucción [CREATE TABLE](#), especifique 'write.compression' = 'ZSTD' o 'parquet.compression' = 'ZSTD' y, a continuación, utilice `compression_level = compression_level` y especifique un valor comprendido entre 1 y 22. Si no se especifica la propiedad `compression_level`, el nivel de compresión predeterminado es 3.

### Ejemplo

En el siguiente ejemplo se crea una tabla con formato de archivo Parquet mediante compresión ZSTD y el nivel 4 de compresión ZSTD.

```
CREATE EXTERNAL TABLE new_table (  
  `col0` string COMMENT '',  
  `col1` string COMMENT ''  
)  
STORED AS PARQUET  
LOCATION 's3://DOC-EXAMPLE-BUCKET/'  
TBLPROPERTIES ('write.compression' = 'ZSTD', 'compression_level' = 4)
```

## CREATE TABLE AS (CTAS)

En la cláusula WITH de la instrucción [CREATE TABLE AS](#), especifique `write_compression = 'ZSTD'` o `parquet_compression = 'ZSTD'` y, a continuación, utilice `compression_level = compression_level` y especifique un valor comprendido entre 1 y 22. Si no se especifica la propiedad `compression_level`, el nivel de compresión predeterminado es 3.

### Ejemplo

En el siguiente ejemplo de CTAS se especifica Parquet como formato de archivo mediante compresión ZSTD con un nivel de compresión 4.

```
CREATE TABLE new_table  
WITH ( format = 'PARQUET', write_compression = 'ZSTD', compression_level = 4)  
AS SELECT * FROM old_table
```

## UNLOAD

En la cláusula WITH de la instrucción [UNLOAD](#), especifique `compression = 'ZSTD'` y, a continuación, utilice `compression_level = compression_level` y especifique un valor

comprendido entre 1 y 22. Si no se especifica la propiedad `compression_level`, el nivel de compresión predeterminado es 3.

## Ejemplo

En el siguiente ejemplo se descargan los resultados de la consulta en la ubicación especificada mediante el formato de archivo Parquet, compresión ZSTD y el nivel 4 de compresión ZSTD.

```
UNLOAD (SELECT * FROM old_table)
TO 's3://DOC-EXAMPLE-BUCKET/'
WITH (format = 'PARQUET', compression = 'ZSTD', compression_level = 4)
```

## Etiquetado de recursos de Athena

Una etiqueta consta de una clave y un valor, ambos definidos por el usuario. Cuando etiqueta un recurso de Athena, le asigna metadatos personalizados. Puede utilizar etiquetas para clasificar los recursos de AWS de diversas maneras, por ejemplo, según su finalidad, propietario o entorno. En Athena, los recursos como los grupos de trabajo, los catálogos de datos y las reservas de capacidad son recursos que se pueden etiquetar. Por ejemplo, puede crear un conjunto de etiquetas para grupos de trabajo en su cuenta que le ayude a realizar un seguimiento de los propietarios de grupos de trabajo o identificar grupos por su finalidad. Si también habilita las etiquetas como etiquetas de asignación de costos en la consola de Administración de facturación y costos, los costos asociados con la ejecución de consultas aparecen en el Informe de costos y usos con esa etiqueta de asignación de costos. Se recomienda seguir las [prácticas recomendadas de etiquetado](#) de AWS para crear un conjunto de etiquetas coherente que satisfaga los requisitos de la organización.

Puede trabajar con etiquetas mediante la consola de Athena o las operaciones de la API.

### Temas

- [Conceptos básicos de etiquetas](#)
- [Restricciones de las etiquetas](#)
- [Uso de las etiquetas en grupos de trabajo en la consola](#)
- [Uso de operaciones de etiquetas](#)
- [Políticas de control de acceso de IAM basado en etiquetas](#)

## Conceptos básicos de etiquetas

Una etiqueta es una marca que se asigna a un recurso de Athena. Cada etiqueta está formada por una clave y un valor opcional, ambos definidos por el usuario.

Las etiquetas le permiten clasificar los recursos de AWS de diversas maneras. Por ejemplo, puede definir un conjunto de etiquetas para los grupos de trabajo de su cuenta que le ayude a realizar un seguimiento del propietario de cada grupo de trabajo o finalidad.

Puede agregar etiquetas al crear un grupo de trabajo de Athena nuevo o catálogo de datos, o puede agregar, editar o eliminar etiquetas de los mismos. Puede editar una etiqueta en la consola. Para utilizar las operaciones de la API para editar una etiqueta, elimine la etiqueta antigua y agregue una nueva. Si elimina un recurso, también se eliminará cualquier etiqueta asignada a dicho recurso.

Athena no asigna automáticamente etiquetas a los recursos. Puede editar las claves y los valores de las etiquetas y también puede eliminar etiquetas de un recurso en cualquier momento. Puede establecer el valor de una etiqueta como una cadena vacía, pero no puede asignarle un valor nulo. No agregue claves de etiqueta duplicadas al mismo recurso. Si lo hace, Athena emite un mensaje de error. Si utiliza la acción `TagResource` para etiquetar un recurso con una clave de etiqueta existente, el nuevo valor de etiqueta sobrescribe el valor anterior.

En IAM puede controlar qué usuarios de la cuenta de Amazon Web Services tienen permiso para crear, editar, eliminar o enumerar etiquetas. Para obtener más información, consulte [Políticas de control de acceso de IAM basado en etiquetas](#).

Para obtener una lista completa de las acciones de etiquetas de Amazon Athena, consulte los nombres de acciones de la API en la [Referencia de API de Amazon Athena](#).

Puede usar etiquetas para la facturación. Para obtener más información, consulte [Uso de etiquetas para facturación](#) en la Guía del usuario de AWS Billing and Cost Management.

Para obtener más información, consulte [Restricciones de las etiquetas](#).

## Restricciones de las etiquetas

Las etiquetas tienen las siguientes restricciones:

- En Athena, puede etiquetar grupos de trabajo y catálogos de datos. No puede etiquetar consultas.
- El número máximo de etiquetas por recurso es 50. Para no sobrepasar el límite, revise y elimine las etiquetas sin usar.

- Para cada recurso, cada clave de etiqueta debe ser única y solo puede tener un valor. No agregue claves de etiquetas duplicadas al mismo tiempo al mismo recurso. Si lo hace, Athena emite un mensaje de error. Si etiqueta un recurso mediante una clave de etiqueta existente en una acción TagResource separada, el valor de la etiqueta nueva sobrescribe el valor antiguo.
- La longitud de la clave de etiqueta es de 1-128 caracteres Unicode en UTF-8.
- La longitud del valor de etiqueta es de 0-256 caracteres Unicode en UTF-8.

Las operaciones de etiquetado, como, por ejemplo, agregar, editar, eliminar o enumerar etiquetas, requieren que especifique un ARN para el recurso de grupo de trabajo.

- Athena le permite utilizar letras, números, espacios representados en UTF-8 y los siguientes caracteres: + - = . \_ : / @.
- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas.
- El prefijo "aws :" de las claves de etiqueta está reservado para el uso de AWS. Las claves de etiquetas que tienen este prefijo no se pueden editar ni eliminar. Las etiquetas que tengan este prefijo no cuentan para el límite de etiquetas por recurso.
- Las etiquetas que asigne solo están disponibles para su cuenta de Amazon Web Services.

## Uso de las etiquetas en grupos de trabajo en la consola

Con la consola de Athena, puede ver qué etiquetas están en uso en cada grupo de trabajo en su cuenta. Puede ver etiquetas solo por grupo de trabajo. También puede utilizar la consola de Athena para aplicar, editar o eliminar etiquetas en un grupo de trabajo a la vez.

Puede buscar grupos de trabajo utilizando las etiquetas que ha creado.

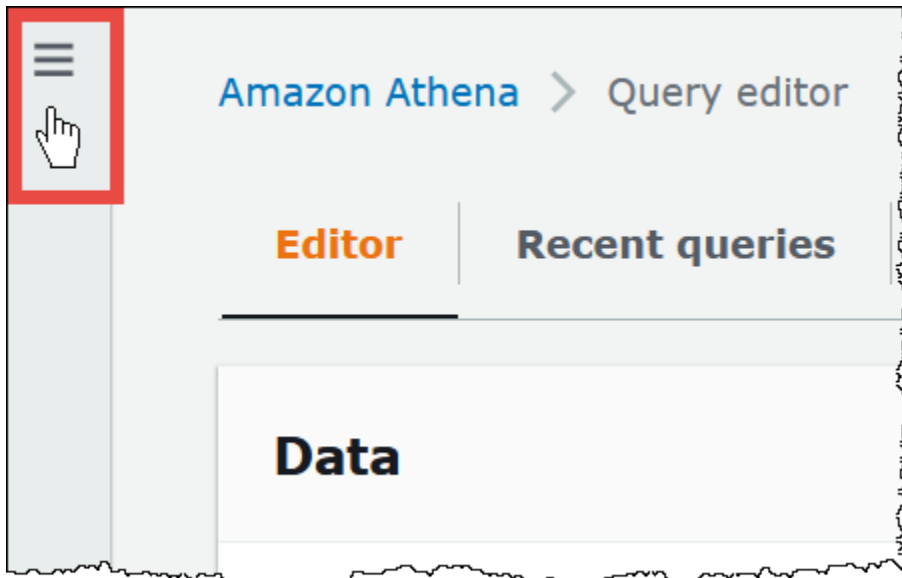
### Temas

- [Visualización de etiquetas para grupos de trabajo individuales](#)
- [Adición y eliminación de etiquetas en un grupo de trabajo individual](#)

### Visualización de etiquetas para grupos de trabajo individuales

Para mostrar las etiquetas de un grupo de trabajo individual en la consola de Athena

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.



3. En el menú de navegación, elija Workgroups (Grupos de trabajo) y, a continuación, elija el grupo de trabajo que quiere.
4. Realice una de las acciones siguientes:
  - Elija la pestaña Etiquetas. Si la lista de etiquetas es larga, utilice el cuadro de búsqueda.
  - Elija Edit (Editar) y, a continuación, desplácese hacia abajo hasta la sección Tags (Etiquetas).

### Adición y eliminación de etiquetas en un grupo de trabajo individual

Puede administrar las etiquetas para un grupo de trabajo individual directamente desde la pestaña Workgroups (Grupos de trabajo).

#### Note

Si desea que los usuarios agreguen etiquetas cuando creen un grupo de trabajo en la consola o pasen etiquetas cuando utilicen la acción CreateWorkGroup, asegúrese de conceder permisos de IAM a los usuarios para las acciones TagResource y CreateWorkGroup.

Para agregar una etiqueta se crea un nuevo grupo de trabajo

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. En el menú de navegación, elija Workgroups (Grupos de trabajo).

3. Elija Create workgroup (Crear grupo de trabajo) y rellene los valores según sea necesario. Para conocer los pasos en detalle, consulte [Creación de un grupo de trabajo](#).
4. En la sección Tags (Etiquetas), agregue una o varias etiquetas. Para ello, especifique las claves y los valores. No añada claves de etiquetas duplicadas al mismo tiempo en el mismo grupo de trabajo. Si lo hace, Athena emite un mensaje de error. Para obtener más información, consulte [Restricciones de las etiquetas](#).
5. Cuando haya terminado, elija Create Workgroup (Crear grupo de trabajo).

Para añadir una etiqueta a un grupo de trabajo existente o editarla

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. En el panel de navegación, elija Workgroups (Redes globales).
3. Elija el grupo de trabajo que quiere modificar.
4. Realice una de las acciones siguientes:
  - Elija la pestaña Etiquetas y, a continuación, elija Administrar etiquetas.
  - Elija Edit (Editar) y, a continuación, desplácese hacia abajo hasta la sección Tags (Etiquetas).
5. Especifique una clave y valor para cada etiqueta. Para obtener más información, consulte [Restricciones de las etiquetas](#).
6. Elija Guardar.

Para eliminar una etiqueta de un grupo de trabajo individual

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. En el panel de navegación, elija Workgroups (Redes globales).
3. Elija el grupo de trabajo que quiere modificar.
4. Realice una de las acciones siguientes:
  - Elija la pestaña Etiquetas y, a continuación, elija Administrar etiquetas.
  - Elija Edit (Editar) y, a continuación, desplácese hacia abajo hasta la sección Tags (Etiquetas).
5. En la lista de etiquetas, elija Remove (Quitar) de la etiqueta que quiera eliminar y, a continuación, elija Save (Guardar).

## Uso de operaciones de etiquetas

Utilice las siguientes operaciones de etiquetas para agregar, quitar o enumerar etiquetas en un recurso.

API	CLI	Descripción de la acción
TagResource	tag-resource	Agregue o sobrescriba una o más etiquetas en el recurso que tiene el ARN especificado.
UntagResource	untag-resource	Elimine una o más etiquetas del recurso que tiene el ARN especificado.
ListTagsForResource	list-tags-for-resource	Enumere una o más etiquetas para el recurso que tiene el ARN especificado.

### Adición de etiquetas al crear un recurso

Para agregar etiquetas al crear un grupo de trabajo o un catálogo de datos, utilice el parámetro `tags` con las operaciones de API `CreateWorkGroup` o `CreateDataCatalog` o con los comandos `create-work-group` o `create-data-catalog` de la AWS CLI.

### Administración de etiquetas mediante operaciones de API

Los ejemplos de esta sección muestran cómo utilizar operaciones de API de etiquetas para administrar etiquetas en grupos de trabajo y catálogos de datos. Los ejemplos están en el lenguaje de programación Java.

#### Example TagResource

En el ejemplo siguiente se agregan dos etiquetas al grupo de trabajo `workgroupA`:

```
List<Tag> tags = new ArrayList<>();
tags.add(new Tag().withKey("tagKey1").withValue("tagValue1"));
tags.add(new Tag().withKey("tagKey2").withValue("tagValue2"));

TagResourceRequest request = new TagResourceRequest()
    .withResourceARN("arn:aws:athena:us-east-1:123456789012:workgroup/workgroupA")
    .withTags(tags);
```



```
client.tagResource(request);
```

En el ejemplo siguiente se agregan dos etiquetas al catálogo de datos `datacatalogA`:

```
List<Tag> tags = new ArrayList<>();
tags.add(new Tag().withKey("tagKey1").withValue("tagValue1"));
tags.add(new Tag().withKey("tagKey2").withValue("tagValue2"));

TagResourceRequest request = new TagResourceRequest()
    .withResourceARN("arn:aws:athena:us-east-1:123456789012:datacatalog/datacatalogA")
    .withTags(tags);

client.tagResource(request);
```

### Note

No agregue claves de etiqueta duplicadas al mismo recurso. Si lo hace, Athena emite un mensaje de error. Si etiqueta un recurso mediante una clave de etiqueta existente en una acción `TagResource` separada, el valor de la etiqueta nueva sobrescribe el valor antiguo.

## Example UntagResource

En el ejemplo siguiente se quita `tagKey2` del grupo de trabajo `workgroupA`:

```
List<String> tagKeys = new ArrayList<>();
tagKeys.add("tagKey2");

UntagResourceRequest request = new UntagResourceRequest()
    .withResourceARN("arn:aws:athena:us-east-1:123456789012:workgroup/workgroupA")
    .withTagKeys(tagKeys);

client.untagResource(request);
```

En el ejemplo siguiente se quita `tagKey2` del catálogo de datos `datacatalogA`:

```
List<String> tagKeys = new ArrayList<>();
tagKeys.add("tagKey2");

UntagResourceRequest request = new UntagResourceRequest()
```

```
.withResourceARN("arn:aws:athena:us-east-1:123456789012:datacatalog/datacatalogA")
.withTagKeys(tagKeys);

client.untagResource(request);
```

## Example ListTagsForResource

En el ejemplo siguiente se enumeran las etiquetas del grupo de trabajo `workgroupA`:

```
ListTagsForResourceRequest request = new ListTagsForResourceRequest()
    .withResourceARN("arn:aws:athena:us-east-1:123456789012:workgroup/workgroupA");

ListTagsForResourceResult result = client.listTagsForResource(request);

List<Tag> resultTags = result.getTags();
```

En el ejemplo siguiente se enumeran las etiquetas del catálogo de datos `datacatalogA`:

```
ListTagsForResourceRequest request = new ListTagsForResourceRequest()
    .withResourceARN("arn:aws:athena:us-east-1:123456789012:datacatalog/datacatalogA");

ListTagsForResourceResult result = client.listTagsForResource(request);

List<Tag> resultTags = result.getTags();
```

## Administrar etiquetas con AWS CLI

En las siguientes secciones se muestra cómo utilizar la AWS CLI para crear y administrar etiquetas en catálogos de datos.

### Adición de etiquetas a un recurso: Tag-resource

El comando `tag-resource` agrega una o varias etiquetas a un recurso especificado.

#### Sintaxis

```
aws athena tag-resource --resource-arn
arn:aws:athena:region:account_id:datacatalog/catalog_name --tags
Key=string,Value=string Key=string,Value=string
```

El parámetro `--resource-arn` especifica el recurso al que se agregan las etiquetas. El parámetro `--tags` especifica una lista de pares clave-valor separados por espacios para agregar como etiquetas al recurso.

## Example

En el ejemplo siguiente se agregan etiquetas al catálogo de datos mydatacatalog.

```
aws athena tag-resource --resource-arn arn:aws:athena:us-east-1:111122223333:datacatalog/mydatacatalog --tags Key=Color,Value=Orange
Key=Time,Value=Now
```

Para mostrar el resultado, use el comando `list-tags-for-resource`.

Para obtener información sobre cómo agregar etiquetas al utilizar el comando `create-data-catalog`, consulte [Registro de un catálogo: create-data-catalog](#).

Descripción de las etiquetas de un recurso: List-tags-for-resource

El comando `list-tags-for-resource` enumera las etiquetas del recurso especificado.

### Sintaxis

```
aws athena list-tags-for-resource --resource-arn
arn:aws:athena:region:account_id:datacatalog/catalog_name
```

El parámetro `--resource-arn` especifica el recurso para el que se enumeran las etiquetas.

En el ejemplo siguiente se enumeran las etiquetas del catálogo de datos mydatacatalog.

```
aws athena list-tags-for-resource --resource-arn arn:aws:athena:us-east-1:111122223333:datacatalog/mydatacatalog
```

El siguiente resultado de ejemplo está en formato JSON.

```
{
  "Tags": [
    {
      "Key": "Time",
      "Value": "Now"
    },
    {
      "Key": "Color",
      "Value": "Orange"
    }
  ]
}
```

## Eliminación de etiquetas de un recurso: Untag-resource

El comando `untag-resource` quita las claves de etiqueta especificadas y sus valores asociados del recurso especificado.

### Sintaxis

```
aws athena untag-resource --resource-arn
arn:aws:athena:region:account_id:datacatalog/catalog_name --tag-keys
key_name [key_name ...]
```

El parámetro `--resource-arn` especifica el recurso del que se quitan las etiquetas. El parámetro `--tag-keys` toma una lista separada por espacios de nombres de las clave. Para cada nombre de clave especificado, el comando `untag-resource` elimina tanto la clave como su valor.

En el ejemplo siguiente se quitan las claves `Color` y `Time` y sus valores del recurso de catálogo `mydatacatalog`.

```
aws athena untag-resource --resource-arn arn:aws:athena:us-
east-1:111122223333:datacatalog/mydatacatalog --tag-keys Color Time
```

## Políticas de control de acceso de IAM basado en etiquetas

Tener etiquetas le permite escribir una política de IAM que incluya el bloque `Condition` para controlar el acceso a un recurso en función de sus etiquetas.

### Ejemplos de política de etiquetas para grupos de trabajo

#### Example 1. Política de etiquetado básico

La siguiente política de IAM le permite ejecutar las consultas e interactuar con etiquetas para el grupo de trabajo llamado `workgroupA`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:ListWorkGroups",
        "athena:ListEngineVersions",
        "athena:ListDataCatalogs",
        "athena:ListDatabases",
```

```

        "athena:GetDatabase",
        "athena:ListTableMetadata",
        "athena:GetTableMetadata"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "athena:GetWorkGroup",
        "athena:TagResource",
        "athena:UntagResource",
        "athena:ListTagsForResource",
        "athena:StartQueryExecution",
        "athena:GetQueryExecution",
        "athena:BatchGetQueryExecution",
        "athena:ListQueryExecutions",
        "athena:StopQueryExecution",
        "athena:GetQueryResults",
        "athena:GetQueryResultsStream",
        "athena:CreateNamedQuery",
        "athena:GetNamedQuery",
        "athena:BatchGetNamedQuery",
        "athena:ListNamedQueries",
        "athena>DeleteNamedQuery",
        "athena:CreatePreparedStatement",
        "athena:GetPreparedStatement",
        "athena:ListPreparedStatements",
        "athena:UpdatePreparedStatement",
        "athena>DeletePreparedStatement"
    ],
    "Resource": "arn:aws:athena:us-east-1:123456789012:workgroup/workgroupA"
}
]
}

```

**Example 2.** Bloqueo de política que deniega acciones en un grupo de trabajo en función de un par de claves de etiqueta y valores de etiqueta

Las etiquetas que están asociadas a un recurso como un grupo de trabajo se denominan etiquetas de recursos. Las etiquetas de recursos le permiten escribir bloques de política como los siguientes que deniegan las acciones enumeradas en cualquier grupo de trabajo etiquetado con un par clave-valor como `stack`, `production`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "athena:GetWorkGroup",
        "athena:UpdateWorkGroup",
        "athena>DeleteWorkGroup",
        "athena:TagResource",
        "athena:UntagResource",
        "athena:ListTagsForResource",
        "athena:StartQueryExecution",
        "athena:GetQueryExecution",
        "athena:BatchGetQueryExecution",
        "athena:ListQueryExecutions",
        "athena:StopQueryExecution",
        "athena:GetQueryResults",
        "athena:GetQueryResultsStream",
        "athena:CreateNamedQuery",
        "athena:GetNamedQuery",
        "athena:BatchGetNamedQuery",
        "athena:ListNamedQueries",
        "athena>DeleteNamedQuery",
        "athena:CreatePreparedStatement",
        "athena:GetPreparedStatement",
        "athena:ListPreparedStatements",
        "athena:UpdatePreparedStatement",
        "athena>DeletePreparedStatement"
      ],
      "Resource": "arn:aws:athena:us-east-1:123456789012:workgroup/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/stack": "production"
        }
      }
    }
  ]
}
```

### Example 3. Bloqueo de política que restringe las solicitudes de acciones de cambio de etiqueta a etiquetas especificadas

Las etiquetas que se pasan como parámetros a operaciones que cambian etiquetas (por ejemplo, `TagResource`, `UntagResource` o `CreateWorkGroup` con etiquetas) se denominan etiquetas de solicitud. El siguiente bloque de política de ejemplo permite la operación `CreateWorkGroup` solo si una de las etiquetas pasadas tiene la clave `costcenter` y el valor 1, 2 o 3.

#### Note

Si desea permitir que un rol de IAM pase etiquetas como parte de una operación `CreateWorkGroup`, asegúrese de conceder permisos al rol para las acciones `TagResource` y `CreateWorkGroup`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:CreateWorkGroup",
        "athena:TagResource"
      ],
      "Resource": "arn:aws:athena:us-east-1:123456789012:workgroup/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/costcenter": [
            "1",
            "2",
            "3"
          ]
        }
      }
    }
  ]
}
```

## Ejemplos de política de etiquetas para catálogos de datos

### Example 1. Política de etiquetado básico

La siguiente política de IAM le permite interactuar con etiquetas para el catálogo de datos denominado `datacatalogA`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:ListWorkGroups",
        "athena:ListEngineVersions",
        "athena:ListDataCatalogs",
        "athena:ListDatabases",
        "athena:GetDatabase",
        "athena:ListTableMetadata",
        "athena:GetTableMetadata"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "athena:GetWorkGroup",
        "athena:TagResource",
        "athena:UntagResource",
        "athena:ListTagsForResource",
        "athena:StartQueryExecution",
        "athena:GetQueryExecution",
        "athena:BatchGetQueryExecution",
        "athena:ListQueryExecutions",
        "athena:StopQueryExecution",
        "athena:GetQueryResults",
        "athena:GetQueryResultsStream",
        "athena:CreateNamedQuery",
        "athena:GetNamedQuery",
        "athena:BatchGetNamedQuery",
        "athena:ListNamedQueries",
        "athena>DeleteNamedQuery"
      ],
      "Resource": [
```



```

        "arn:aws:athena:us-east-1:123456789012:workgroup/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "athena:CreateDataCatalog",
        "athena:GetDataCatalog",
        "athena:UpdateDataCatalog",
        "athena>DeleteDataCatalog",
        "athena:ListDatabases",
        "athena:GetDatabase",
        "athena:ListTableMetadata",
        "athena:GetTableMetadata",
        "athena:TagResource",
        "athena:UntagResource",
        "athena:ListTagsForResource"
    ],
    "Resource": "arn:aws:athena:us-
east-1:123456789012:datacatalog/datacatalogA"
}
]
}

```

Example 2. Bloqueo de política que deniega acciones en un catálogo de datos en función de un par de claves de etiqueta y valores de etiqueta

Puede utilizar etiquetas de recursos para escribir bloques de política que denieguen acciones específicas en catálogos de datos etiquetados con pares clave-valor de etiqueta específicos. La siguiente política de ejemplo deniega acciones en catálogos de datos que tienen el par clave-valor de etiqueta `stack, production`.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "athena:CreateDataCatalog",
                "athena:GetDataCatalog",
                "athena:UpdateDataCatalog",
                "athena>DeleteDataCatalog",
                "athena:GetDatabase",
            ]
        }
    ]
}

```

```

        "athena:ListDatabases",
        "athena:GetTableMetadata",
        "athena:ListTableMetadata",
        "athena:StartQueryExecution",
        "athena:TagResource",
        "athena:UntagResource",
        "athena:ListTagsForResource"
    ],
    "Resource": "arn:aws:athena:us-east-1:123456789012:datacatalog/*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/stack": "production"
        }
    }
}
]
}
}

```

**Example 3.** Bloqueo de política que restringe las solicitudes de acciones de cambio de etiqueta a etiquetas especificadas

Las etiquetas que se pasan como parámetros a operaciones que cambian etiquetas (por ejemplo, TagResource, UntagResource o CreateDataCatalog con etiquetas) se denominan etiquetas de solicitud. El siguiente bloque de política de ejemplo permite la operación CreateDataCatalog solo si una de las etiquetas pasadas tiene la clave costcenter y el valor 1, 2 o 3.

#### Note

Si desea permitir que un rol de IAM pase etiquetas como parte de una operación CreateDataCatalog, asegúrese de conceder permisos al rol para las acciones TagResource y CreateDataCatalog.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:CreateDataCatalog",
        "athena:TagResource"
      ]
    }
  ]
}

```

```
    ],
    "Resource": "arn:aws:athena:us-east-1:123456789012:datacatalog/*",
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/costcenter": [
          "1",
          "2",
          "3"
        ]
      }
    }
  }
]
```

## Service Quotas

### Note

La consola de Service Quotas proporciona información sobre las cuotas de Amazon Athena. También puede utilizar la consola de Service Quotas a fin de [solicitar aumentos de cuota](#) para las cuotas ajustables. Para conocer las limitaciones del esquema relacionadas con AWS Glue, consulte la página de [puntos de conexión y cuotas de AWS Glue](#). Para obtener información general sobre AWS Service Quotas, consulte [AWS Service Quotas](#) en la Referencia general de AWS.

## Consultas

Su cuenta tiene las siguientes cuotas relacionadas con consultas para Amazon Athena. Para obtener más información, consulte la página de [puntos de conexión y cuotas de Amazon Athena](#) de la Referencia general de AWS.

- Consultas DDL activas: cantidad de consultas DDL activas. Las consultas DDL incluyen consultas CREATE TABLE y ALTER TABLE ADD PARTITION.
- DDL query timeout (Tiempo de espera de consulta DDL): intervalo máximo de tiempo en minutos durante el que se puede ejecutar una consulta DDL antes de que se cancele.

- **Active DML queries (Consultas DML activas):** cantidad de consultas DML activas. Las consultas DML incluyen consultas `SELECT`, `CREATE TABLE AS (CTAS)` y `INSERT INTO`. Las cuotas específicas varían según la región de AWS.
- **Tiempo de espera de consulta DML:** intervalo máximo de tiempo en minutos durante el que se puede ejecutar una consulta DML antes de que se cancele. Puede solicitar un aumento de este tiempo de espera hasta un máximo de 240 minutos.

Para solicitar el aumento de una cuota, puede utilizar la consola de [Athena Service Quotas](#).

Athena procesa las consultas mediante la asignación de recursos en función de la carga general del servicio y el número de solicitudes entrantes. Es posible que las consultas se pongan en cola temporalmente antes de que se ejecuten. Los procesos asíncronos recogen las consultas de las colas y las ejecutan en recursos físicos tan pronto como los recursos están disponibles y durante el tiempo que la configuración de la cuenta lo permita.

Una cuota de consulta DML o DDL incluye tanto las consultas en ejecución como en cola. Por ejemplo, si la cuota de consultas DML es de 25 y el total de consultas ejecutadas y en cola supera es 26, la consulta 26 generará un error `TooManyRequestsException`.

#### Note

Si desea controlar la simultaneidad de forma directa para las consultas que ejecuta en Athena, puede utilizar las reservas de capacidad. Para obtener más información, consulte [Administración de la capacidad de procesamiento de consultas](#).

## Longitud de cadena de consulta

La longitud máxima permitida de la cadena de consulta es 262144 bytes, donde las cadenas se codifican en UTF-8. No se trata de una cuota ajustable. Sin embargo, puede evitar esta limitación al dividir las consultas largas en varias consultas más pequeñas. Para obtener más información, consulte [¿Cómo puedo aumentar la longitud máxima de cadena de consulta en Athena?](#) en el Centro de conocimientos de AWS.

## Grupos de trabajo

Cuando trabaje con grupos de trabajo de Athena, recuerde los siguientes puntos:

- Las cuotas de servicio de Athena se comparten entre todos los grupos de trabajo de una cuenta.

- El número máximo de grupos de trabajo que puede crear por región en una cuenta es 1000.
- El número máximo de instrucciones preparadas en un grupo de trabajo es 1000.
- El número máximo de etiquetas por grupo de trabajo es 50. Para obtener más información, consulte [Restricciones de las etiquetas](#).

## Bases de datos, tablas y particiones

- Si utiliza AWS Glue Data Catalog con Athena, consulte [Puntos de conexión y cuotas de AWS Glue](#) para Service Quotas en tablas, bases de datos y particiones; por ejemplo, el número máximo de bases de datos o tablas por cuenta.
  - Aunque Athena admite consultas de tablas de AWS Glue que tienen 10 millones de particiones, no puede leer más de 1 millón de particiones en un solo escaneo.
- Si no está utilizando AWS Glue Data Catalog, el número de particiones por tabla es de 20 000. Puede [solicitar un aumento de cuota](#).

## Buckets de Amazon S3

Cuando trabaje con buckets de Amazon S3, recuerde los siguientes puntos:

- Amazon S3 tiene una cuota de servicio predeterminada de 100 buckets por cuenta.
- Athena necesita un bucket independiente para registrar los resultados.
- Puede solicitar un aumento de cuota de hasta 1000 buckets de Amazon S3 por cuenta de AWS.

## Cuotas de llamadas a la API por cuenta

Las API de Athena tienen las siguientes cuotas predeterminadas para el número de llamadas a la API por cuenta (no por consulta):

Nombre de API	Número predeterminado de llamadas por segundo	Capacidad de ampliación
BatchGetNamedQuery , ListNamedQueries , ListQueryExecutions	5	hasta 10

Nombre de API	Número predeterminado de llamadas por segundo	Capacidad de ampliación
CreateNamedQuery , DeleteNamedQuery , GetNamedQuery	5	hasta 20
BatchGetQueryExecution	20	hasta 40
StartQueryExecution , StopQueryExecution	20	hasta 80
GetQueryExecution , GetQueryResults	100	hasta 200

Por ejemplo, puede realizar hasta 20 llamadas por segundo para `StartQueryExecution`. Además, si no se llama a esta API durante cuatro segundos, la cuenta acumula una capacidad de ampliación de hasta 80 llamadas. En este caso, la aplicación puede realizar hasta 80 llamadas a esta API en modo de ráfaga.

Si utiliza cualquiera de estas API y supera la cuota predeterminada del número de llamadas por segundo o la capacidad de ampliación de su cuenta, la API de Athena genera un error similar al siguiente: “ClientError: An error occurred (ThrottlingException) when calling the <API\_name> operation: Rate exceeded”. Reduzca el número de llamadas por segundo o la capacidad de ampliación para la API para esta cuenta.

La cuota de Athena para llamadas a la API por cuenta no se puede cambiar en la consola de Service Quotas de Athena. Para solicitar un aumento de cuota para las llamadas a la API de Athena, diríjase a la página de [Aumento del límite de servicio](#) de AWS Support y complete y envíe el formulario.

## Control de versiones del motor Athena

Ocasionalmente, Athena lanza una nueva versión del motor para proporcionar un mejor rendimiento, funcionalidad y correcciones de código. Cuando una nueva versión del motor está disponible, Athena se lo notifica a través de la consola y su [AWS Health Dashboard](#). Su AWS Health Dashboard le notifica sobre los eventos que pueden afectar sus servicios o cuenta de AWS. Para obtener más información sobre AWS Health Dashboard, consulte [Introducción a AWS Health Dashboard](#).

El control de versiones del motor se configura por [WorkGroup](#). Puede utilizar grupos de trabajo para controlar qué motor de consultas utilizan sus consultas y para decidir si permite que Athena actualice los grupos de trabajo de manera automática. El motor de consultas que está en uso se muestra en el editor de consultas, en la página de detalles del grupo de trabajo y está disponible a través de las API de Athena.

- Los grupos de trabajo se configuran de forma predeterminada para actualizarse automáticamente. Cuando un grupo de trabajo está configurado para que se actualice de forma automática, Athena actualiza el grupo de trabajo por usted, a menos que encuentre incompatibilidades.
- Si configura un grupo de trabajo para usar una versión determinada, Athena no cambiará la versión del grupo de trabajo.

En ambos casos, Athena actualiza los grupos de trabajo cuando una versión ya no está disponible. Athena lo notifica a través de [AWS Health Dashboard](#) respecto a cuándo dejará de ofrecerse una versión de motor. Su AWS Health Dashboard le notifica sobre los eventos que pueden afectar sus servicios o cuenta de AWS. Para obtener más información sobre AWS Health Dashboard, consulte [Introducción a AWS Health Dashboard](#).

Cuando comienza a utilizar una nueva versión del motor, es posible que un pequeño subconjunto de consultas se rompa debido a incompatibilidades. Los cambios más importantes se anuncian cuando se lanza una nueva versión de Athena. Puede utilizar grupos de trabajo para probar las consultas antes de la actualización si crea un grupo de trabajo de prueba que utilice el nuevo motor o si prueba la actualización de un grupo de trabajo existente. Para obtener más información, consulte [Pruebas de consultas antes de una actualización de la versión del motor](#).

## Temas

- [Cambio de las versiones del motor Athena](#)
- [Referencia de la versión del motor Athena](#)

## Cambio de las versiones del motor Athena

Ocasionalmente, Athena lanza una nueva versión del motor para proporcionar un mejor rendimiento, funcionalidad y correcciones de código. Cuando hay disponible una versión más reciente, Athena se lo notifica en la consola. Puede optar por dejar que Athena decida cuándo actualizar o puede especificar manualmente una versión del motor Athena por grupo de trabajo.

## Temas

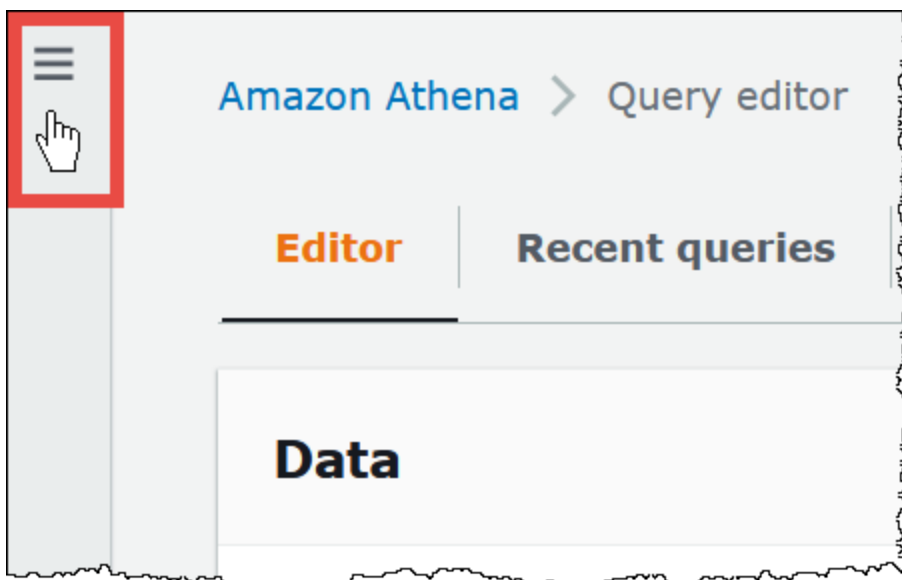
- [Búsqueda de la versión del motor de consultas para un grupo de trabajo](#)
- [Cambio de versión del motor en la consola de Athena](#)
- [Cambio de la versión del motor con la AWS CLI](#)
- [Especificación de la versión del motor cuando crea un grupo de trabajo](#)
- [Pruebas de consultas antes de una actualización de la versión del motor](#)
- [Solución de problemas de consultas que producen un error](#)

## Búsqueda de la versión del motor de consultas para un grupo de trabajo

Puede utilizar la página de Workgroups (Grupos de trabajo) para encontrar la versión actual del motor para cualquier grupo de trabajo.

Para buscar la versión actual del motor para cualquier grupo de trabajo

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.



3. En el panel de navegación de la consola de Athena, elija Workgroups (Grupos de trabajo).
4. En la página Workgroups, busque el grupo de trabajo que quiera. La columna Query engine version (Versión del motor de consultas) del grupo de trabajo muestra la versión del motor de consultas.



## Cambio de versión del motor en la consola de Athena

Cuando haya disponible una nueva versión del motor, podrá optar por dejar que Athena decida cuándo actualizar el grupo de trabajo o puede especificar manualmente la versión del motor Athena que utiliza el grupo de trabajo. Si solo hay una versión disponible actualmente, no es posible especificar manualmente otra versión.

### Note

Para cambiar la versión del motor de un grupo de trabajo, debe tener permiso para ejecutar la acción `athena:ListEngineVersions` en el grupo de trabajo. Para ver ejemplos de política de IAM, consulte [Ejemplos de políticas de grupos de trabajo](#).

Para dejar que Athena decida cuándo actualizar el grupo de trabajo

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.
3. En el panel de navegación de la consola, elija Workgroups.
4. En la lista de grupos de trabajo, elija el enlace del grupo de trabajo que quiere configurar.
5. Elija Editar.
6. En la sección Query engine version, en Update query engine (Actualizar motor de consultas), elija Automatic (Automático) para dejar que Athena elija cuándo actualizar el grupo de trabajo. Este es el valor predeterminado.
7. Elija Guardar cambios.

En la lista de grupos de trabajo, la opción Query engine update status (Estado de actualización del motor de consultas) del grupo de trabajo muestra Automatic (Automático).

Para elegir manualmente una versión de motor

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.
3. En el panel de navegación de la consola, elija Workgroups.

4. En la lista de grupos de trabajo, elija el enlace del grupo de trabajo que quiere configurar.
5. Elija Editar.
6. En la sección Query engine version, en Update query engine, elija Manual para elegir manualmente una versión del motor
7. Use la opción Query engine version para elegir la versión del motor que quiere que utilice el grupo de trabajo. Si no está disponible otra versión del motor, no se puede especificar una versión del motor diferente.
8. Elija Guardar cambios.

En la lista de grupos de trabajo, la opción Query engine update status (Estado de actualización del motor de consultas) del grupo de trabajo muestra Manual.

## Cambio de la versión del motor con la AWS CLI

Para cambiar la versión del motor con la AWS CLI, utilice la sintaxis del ejemplo siguiente.

```
aws athena update-work-group --work-group workgroup-name --configuration-updates EngineVersion={SelectedEngineVersion='Athena engine version 3'}
```

## Especificación de la versión del motor cuando crea un grupo de trabajo

Al crear un grupo de trabajo, puede especificar la versión del motor que utiliza el grupo de trabajo o puede dejar que Athena decida cuándo actualizar el grupo de trabajo. Si está disponible una nueva versión del motor, una práctica recomendada es crear un grupo de trabajo para probar el nuevo motor antes de actualizar los otros grupos de trabajo. Para especificar la versión del motor de un grupo de trabajo, debe tener el permiso de `athena:ListEngineVersions` en el grupo de trabajo. Para ver ejemplos de política de IAM, consulte [Ejemplos de políticas de grupos de trabajo](#).

Para especificar la versión del motor al crear un grupo de trabajo

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.
3. En el panel de navegación de la consola, elija Workgroups.
4. En el panel Grupos de trabajo, elija Crear grupo de trabajo.
5. En la página Create workgroup (Crear grupo de trabajo), en la sección Query engine version, lleve a cabo una de las siguientes acciones:

- Elija Automatic para dejar que Athena elija cuándo actualizar el grupo de trabajo. Este es el valor predeterminado.
  - Elija Manual para elegir manualmente otra versión del motor si hay una disponible.
6. Ingrese la información para los demás campos, según sea necesario. Para obtener información acerca de los campos, consulte [Creación de un grupo de trabajo](#).
  7. Elija Crear grupo de trabajo.

## Pruebas de consultas antes de una actualización de la versión del motor

Cuando un grupo de trabajo se actualiza a una nueva versión del motor, algunas de sus consultas pueden invalidarse debido a incompatibilidades. Para asegurarse de que la actualización de la versión del motor se efectúe sin inconvenientes, puede probar las consultas con anticipación.

Para probar las consultas antes de una actualización de la versión del motor

1. Verifique la versión del motor del grupo de trabajo que está utilizando. La versión del motor que está utilizando se muestra en la página Workgroups en la columna Query engine version del grupo de trabajo. Para obtener más información, consulte [Búsqueda de la versión del motor de consultas para un grupo de trabajo](#).
2. Cree un grupo de trabajo de prueba que utilice la nueva versión del motor. Para obtener más información, consulte [Especificación de la versión del motor cuando crea un grupo de trabajo](#).
3. Utilice el nuevo grupo de trabajo para ejecutar las consultas que desea probar.
4. Si se produce un error en una consulta, utilice [Referencia de la versión del motor Athena](#) para verificar si hay cambios sustanciales que podrían estar afectando a la consulta. Es posible que algunos cambios requieran que actualice la sintaxis de las consultas.
5. Si las consultas continúan produciendo un error, póngase en contacto con AWS Support para obtener ayuda. En la AWS Management Console, elija Support (Asistencia), Support Center (Centro de asistencia) o haga una pregunta en [AWS re:Post](#) con la etiqueta Amazon Athena.

## Solución de problemas de consultas que producen un error

Si una consulta falla después de una actualización de la versión del motor de, utilice [Referencia de la versión del motor Athena](#) para verificar si hay cambios sustanciales, incluidos los que pueden afectar la sintaxis de las consultas.

Si las consultas continúan produciendo un error, póngase en contacto con AWS Support para obtener ayuda. En la AWS Management Console, elija Support (Asistencia), Support Center (Centro de asistencia) o haga una pregunta en [AWS re:Post](#) con la etiqueta Amazon Athena.

## Referencia de la versión del motor Athena

En esta sección, se enumeran los cambios efectuados en el motor de consultas Athena.

### Temas

- [Versión 3 del motor Athena](#)
- [Versión 2 del motor Athena](#)

### Versión 3 del motor Athena

Para la versión 3 del motor, Athena presenta un enfoque de integración continua para la administración del software de código abierto que mejora la aceptación de los proyectos de [Trino](#) y [Presto](#) de modo que pueda acceder más rápido a las mejoras de la comunidad, integradas y ajustadas en el motor de Athena.

El lanzamiento de la versión 3 del motor Athena admite todas las características de la versión 2 del motor Athena. Este documento destaca las diferencias clave entre la versión 2 y la versión 3 del motor Athena. Para obtener más información, consulte el artículo del Blog de macrodatos de AWS [Actualizar a la versión 3 del motor de Athena para aumentar el rendimiento de las consultas y acceder a más características de análisis](#).

- [Introducción](#)
- [Mejoras y nuevas características](#)
  - [Características agregadas](#)
  - [Funciones agregadas](#)
  - [Mejoras en el rendimiento](#)
  - [Mejoras de la fiabilidad](#)
  - [Mejoras de la sintaxis de las consultas](#)
  - [Mejoras en el formato y el tipo de datos](#)
- [Cambios bruscos](#)
  - [Cambios en la sintaxis de las consultas](#)
  - [Cambios en el procesamiento de datos](#)

- [Cambios en la marca de tiempo](#)
- [Limitaciones](#)

## Introducción

Para empezar, cree un nuevo grupo de trabajo de Athena que utilice la versión 3 del motor Athena o configure un grupo de trabajo existente para que utilice la versión 3. Cualquier grupo de trabajo de Athena puede actualizar de la versión 2 a la versión 3 del motor sin interrumpir su capacidad de enviar consultas.

Para obtener más información, consulte [Cambio de las versiones del motor Athena](#).

## Mejoras y nuevas características

Las características y actualizaciones enumeradas incluyen mejoras de Athena y de la funcionalidad incorporada desde Trino de código abierto. Para obtener una lista exhaustiva de las funciones y los operadores de consultas SQL, consulte la [documentación de Trino](#).

## Características agregadas

### Compatibilidad con el algoritmo de agrupación en buckets de Apache

Athena puede leer los buckets generados por el algoritmo hash de Spark. Para especificar que, originalmente, el algoritmo hash de Spark escribió los datos, ingrese ( 'bucketing\_format' = 'spark' ) en la cláusula TBLPROPERTIES de la instrucción CREATE TABLE. Si no se especifica esta propiedad, se utiliza el algoritmo hash de Hive.

```
CREATE EXTERNAL TABLE `spark_bucket_table`(  
  `id` int,  
  `name` string  
)  
CLUSTERED BY (`name`)  
INTO 8 BUCKETS  
STORED AS PARQUET  
LOCATION  
  's3://DOC-EXAMPLE-BUCKET/to/bucketed/table/'  
TBLPROPERTIES ('bucketing_format'='spark')
```

## Funciones agregadas

Las funciones de esta sección se introdujeron en la versión 3 del motor Athena.

## Funciones de agregación

`listagg(x, separator)`: devuelve los valores de entrada concatenados, separados por la cadena de separación.

```
SELECT listagg(value, ',') WITHIN GROUP (ORDER BY value) csv_value
FROM (VALUES 'a', 'c', 'b') t(value);
```

## Funciones de matriz

`contains_sequence(x, seq)`: devuelve True (Verdadero) si la matriz x contiene toda la secuencia de matrices como un subconjunto secuencial (todos los valores en el mismo orden consecutivo).

```
SELECT contains_sequence(ARRAY [1,2,3,4,5,6], ARRAY[1,2]);
```

## Funciones binarias

`murmur3(binary)`: calcula el hash MurmurHash3 de 128 bits del binario.

```
SELECT murmur3(from_base64('aaaaaa'));
```

## Funciones de conversión

`format_number(number)`: devuelve una cadena formateada con un símbolo de unidad.

```
SELECT format_number(123456); -- '123K'
```

```
SELECT format_number(1000000); -- '1M'
```

## Funciones de fecha y hora

`timezone_hour(timestamp)`: devuelve la hora de diferencia de zona horaria con respecto a la marca de tiempo.

```
SELECT EXTRACT(TIMEZONE_HOUR FROM TIMESTAMP '2020-05-10 12:34:56 +08:35');
```

`timezone_minute(timestamp)`: devuelve los minutos de diferencia de la zona horaria con respecto a la marca de tiempo.

```
SELECT EXTRACT(TIMEZONE_MINUTE FROM TIMESTAMP '2020-05-10 12:34:56 +08:35');
```

## Funciones geoespaciales

`to_encoded_polyline(Geometry)`: codifica una cadena lineal o un multipunto en una polilínea.

```
SELECT to_encoded_polyline(ST_GeometryFromText(
  'LINESTRING (-120.2 38.5, -120.95 40.7, -126.453 43.252)');
```

`from_encoded_polyline(varchar)`: decodifica una polilínea en una cadena de líneas.

```
SELECT ST_AsText(from_encoded_polyline('_p~iF~ps|U_uLLnnqC_mqNvxq`@'));
```

`to_geojson_geometry(SphericalGeography)`: devuelve la geografía esférica especificada en formato GeoJSON.

```
SELECT to_geojson_geometry(to_spherical_geography(ST_GeometryFromText(
  'LINESTRING (0 0, 1 2, 3 4)')));
```

`from_geojson_geometry(varchar)`: devuelve el objeto de tipo geográfico esférico de la representación de GeoJSON después de eliminar las claves o los valores no geométricos. `Feature` y `FeatureCollection` no son compatibles.

```
SELECT
  from_geojson_geometry(to_geojson_geometry(to_spherical_geography(ST_GeometryFromText(
    'LINESTRING (0 0, 1 2, 3 4)'))));
```

`geometry_nearest_points(Geometry, Geometry)`: devuelve los puntos de cada geometría que están más cerca unos de otros. Si la geometría está vacía, devuelve un valor NULL. De lo contrario, devuelve una fila de dos objetos `Point` que tienen la distancia mínima de dos puntos en las geometrías. El primer punto proviene del primer argumento `Geometry`, y el segundo proviene del segundo argumento `Geometry`. Si hay varios pares con la misma distancia mínima, se elige un par arbitrariamente.

```
SELECT geometry_nearest_points(ST_GeometryFromText(
  'LINESTRING (50 100, 50 200)'), ST_GeometryFromText(
  'LINESTRING (10 10, 20 20)');
```

## Configuración de funciones Digest

`make_set_digest(x)`: compone todos los valores de entrada de `x` en un `setdigest`.

```
SELECT make_set_digest(value) FROM (VALUES 1, 2, 3) T(value);
```

## Funciones de cadena

`soundex(char)`: devuelve una cadena de caracteres que contiene la representación fonética de `char`.

```
SELECT name
FROM nation
WHERE SOUNDEX(name) = SOUNDEX('CHYNA'); -- CHINA
```

`concat_ws(string0, string1, ..., stringN)`: devuelve la concatenación de `string1`, `string2`, ..., `stringN` con `string0` como separador. Si `string0` es `NULL`, el valor devuelto es `NULL`. Se omiten todos los valores nulos proporcionados en los argumentos después del separador.

```
SELECT concat_ws(',', 'def', 'pqr', 'mno');
```

## Funciones de ventana

**GROUPS**: agrega compatibilidad para marcos de ventanas basados en grupos.

```
SELECT array_agg(a) OVER(
  ORDER BY a ASC NULLS FIRST GROUPS BETWEEN 1 PRECEDING AND 2 FOLLOWING)
FROM (VALUES 3, 3, 3, 2, 2, 1, null, null) T(a);
```

## Mejoras en el rendimiento

En la versión 3 del motor Athena, se ha mejorado el rendimiento de las siguientes características:

- Recuperación más rápida de los metadatos de las tablas de AWS Glue: las consultas con varias tablas reducirán el tiempo de planificación de las consultas.
- Filtrado dinámico para **RIGHT JOIN**: el filtrado dinámico ahora está habilitado para las operaciones **RIGHT JOIN** que tienen condiciones de combinación de igualdad, como en el siguiente ejemplo.

```
SELECT *
```



```
FROM lineitem RIGHT JOIN tpch.tiny.supplier
ON lineitem.suppkey = supplier.suppkey
WHERE supplier.name = 'abc';
```

- Instrucciones preparadas de gran tamaño: se ha aumentado el tamaño predeterminado del encabezado de solicitud y respuesta HTTP a 2 MB para permitir instrucciones preparadas de gran tamaño.
- `approx_percentile()`: la función `approx_percentile` ahora utiliza `tdigest` en lugar de `qdigest` para recuperar valores cuantiles aproximados de las distribuciones. Esto se traduce en un mayor rendimiento y un menor uso de memoria. Tenga en cuenta que, como resultado de este cambio, la función arroja resultados diferentes a los que arrojaba en la versión 2 del motor de Athena. Para obtener más información, consulte [La función `approx\_percentile` devuelve resultados diferentes](#).

### Mejoras de la fiabilidad

Se han mejorado el uso general de la memoria del motor y el seguimiento en la versión 3 del motor Athena. Las consultas de gran tamaño son menos susceptibles de fallar por caídas de nodos.

### Mejoras de la sintaxis de las consultas

**INTERSECT ALL:** se agregó compatibilidad con `INTERSECT ALL`.

```
SELECT * FROM (VALUES 1, 2, 3, 4) INTERSECT ALL SELECT * FROM (VALUES 3, 4);
```

**EXCEPT ALL:** se agregó compatibilidad con `EXCEPT ALL`.

```
SELECT * FROM (VALUES 1, 2, 3, 4) EXCEPT ALL SELECT * FROM (VALUES 3, 4);
```

**RANGE PRECEDING:** se agregó compatibilidad con `RANGE PRECEDING` en funciones de ventana.

```
SELECT sum(x) over (order by x range 1 preceding)
FROM (values (1), (1), (2), (2)) t(x);
```

**MATCH\_RECOGNIZE:** se agregó compatibilidad con la coincidencia de patrones de filas, como en el siguiente ejemplo.

```
SELECT m.id AS row_id, m.match, m.val, m.label
FROM (VALUES(1, 90),(2, 80),(3, 70),(4, 70)) t(id, value)
```

```
MATCH_RECOGNIZE (
  ORDER BY id
  MEASURES match_number() AS match,
  RUNNING LAST(value) AS val,
  classifier() AS label
  ALL ROWS PER MATCH
  AFTER MATCH SKIP PAST LAST ROW
  PATTERN (() | A) DEFINE A AS true
) AS m;
```

## Mejoras en el formato y el tipo de datos

La versión 3 del motor Athena incluye las siguientes mejoras de formato y tipo de datos:

- LZ4 y ZSTD: se agregó compatibilidad con la lectura de datos de Parquet comprimidos con LZ4 y ZSTD. Se agregó compatibilidad con la lectura de datos de ORC comprimidos con ZSTD.
- Tablas basadas en enlaces simbólicos: se agregó compatibilidad con la lectura de tablas basadas en enlaces simbólicos en archivos de Avro. Ejemplo:

```
CREATE TABLE test_avro_symlink
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.avro.AvroSerDe'
...
INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
```

- SphericalGeography: el tipo SphericalGeography proporciona soporte nativo para las características espaciales representadas en coordenadas geográficas (a veces denominadas coordenadas geodésicas, lat/lon o lon/lat). Las coordenadas geográficas son coordenadas esféricas expresadas en unidades angulares (grados).

La función `to_spherical_geography` devuelve coordenadas geográficas (esféricas) a partir de coordenadas geométricas (planas), como en el siguiente ejemplo.

```
SELECT to_spherical_geography(ST_GeometryFromText(
  'LINESTRING (-40.2 28.9, -40.2 31.9, -37.2 31.9)'));
```

## Cambios bruscos

Al migrar de la versión 2 del motor Athena a la versión 3, ciertos cambios pueden afectar al esquema de la tabla, la sintaxis o el uso del tipo de datos. En esta sección, se enumeran los mensajes de error asociados y se sugieren soluciones alternativas.

## Cambios en la sintaxis de las consultas

IGNORE NULLS no se puede utilizar con funciones de ventana sin valores

Mensaje de error: No se puede especificar una cláusula de tratamiento nulo para la función `bool_or`.

Causa: ahora solo se puede usar IGNORE NULLS con las [funciones de valor](#) `first_value`, `last_value`, `nth_value`, `lead` y `lag`. Este cambio se hizo para cumplir con la especificación ANSI SQL.

Solución sugerida: eliminar IGNORE NULLS de las funciones de ventana sin valores en las cadenas de consulta.

La función CONCAT debe tener dos o más argumentos

Mensaje de error: INVALID\_FUNCTION\_ARGUMENT: debe haber dos o más argumentos de concatenación

Causa: anteriormente, la función de cadena CONCAT aceptaba un solo argumento. En la versión 3 del motor Athena, la función CONCAT requiere un mínimo de dos argumentos.

Solución sugerida: cambie las ocurrencias de `CONCAT(str)` por `CONCAT(str, '')`.

En la versión 3 del motor de Athena, las funciones no pueden tener más de 127 argumentos. Para obtener más información, consulte [Hay demasiados argumentos para llamar a la función](#).

La función `approx_percentile` devuelve resultados diferentes

La función `approx_percentile` arroja resultados diferentes en la versión 3 que en la versión 2 del motor de Athena.

Mensaje de error: ninguno.

Causa: la función `approx_percentile` está sujeta a cambios de versión.

### Important

Como los resultados de la función `approx_percentile` son aproximaciones y las aproximaciones están sujetas a cambios de una versión a otra, no debe confiar en la función `approx_percentile` para aplicaciones críticas.

Solución sugerida: para aproximar el comportamiento de `approx_percentile` de la versión 2 del motor de Athena, puede utilizar un conjunto diferente de funciones en la versión 3 del motor de Athena. Por ejemplo, supongamos que tiene la siguiente consulta en la versión 2 del motor de Athena:

```
SELECT approx_percentile(somecol, 2E-1)
```

Para obtener aproximadamente el mismo resultado en la versión 3 del motor de Athena, puede probar las funciones `qdigest_agg` y `value_at_quantile`, como en el siguiente ejemplo. Tenga en cuenta que, incluso con esta solución alternativa, no se garantiza el mismo comportamiento.

```
SELECT value_at_quantile(qdigest_agg(somecol, 1), 2E-1)
```

La función geoespacial no admite la entrada `varbinary`

Mensaje de error: `FUNCTION_NOT_FOUND` para `st_XXX`

Causa: algunas funciones geoespaciales ya no admiten el tipo de entrada `VARBINARY` heredado o las firmas de funciones relacionadas con el texto.

Solución sugerida: utilice funciones geoespaciales para convertir los tipos de entrada en los tipos compatibles. Los tipos de entrada admitidos se indican en el mensaje de error.

Las columnas anidadas deben ir entre comillas dobles en las cláusulas `GROUP BY`

Mensaje de error: `"column_name".nested_column` debe ser una expresión agregada o aparecer en la cláusula `GROUP BY`

Causa: la versión 3 del motor de Athena requiere que los nombres de las columnas anidadas en las cláusulas `GROUP BY` estén entre comillas dobles. Por ejemplo, la siguiente consulta produce el error porque, en la cláusula `GROUP BY`, `user.name` no aparece entre comillas dobles.

```
SELECT "user"."name" FROM dataset
GROUP BY user.name
```

Solución sugerida: coloque comillas dobles alrededor de los nombres de las columnas anidadas en las cláusulas `GROUP BY`, como en el siguiente ejemplo.

```
SELECT "user"."name" FROM dataset
GROUP BY "user"."name"
```

## Error inesperado de FilterNode al utilizar OPTIMIZE en una tabla Iceberg

Mensaje de error: Se encontró un FilterNode inesperado en el plan; probablemente el conector no pudo gestionar la expresión WHERE proporcionada.

Causa: la instrucción OPTIMIZE que se ejecutó en la tabla Iceberg utilizaba una cláusula WHERE que incluía una columna que no era de partición en su expresión de filtro.

Solución sugerida: la instrucción OPTIMIZE solo admite el filtrado por particiones. Cuando ejecute OPTIMIZE en las tablas particionadas, incluya solo las columnas de partición en la cláusula WHERE. Si ejecuta OPTIMIZE en una tabla no particionada, no especifique ninguna cláusula WHERE.

## Orden de los argumentos de la función Log()

En la versión 2 del motor Athena, el orden de los argumentos de la función `log()` era `log(value, base)`. En la versión 3 del motor Athena, esto ha cambiado a `log(base, value)` para ajustarse a los estándares de SQL.

## La función minute() no admite intervalos de un año a un mes

Mensaje de error: Parámetros inesperados (intervalo de un año a un mes) para la función minute. Expected: minute(timestamp with time zone), minute(time with time zone), minute(timestamp), minute(time), minute(interval day to second) (Valores esperados: minute[marca de tiempo con zona horaria], minute[hora con zona horaria], minute[marca de tiempo], minute[hora], minute[intervalo de días a segundos]).

Causa: en la versión 3 del motor Athena, las comprobaciones de tipo se han hecho más precisas para EXTRACT de acuerdo con la especificación ANSI SQL.

Solución sugerida: actualice las consultas para asegurarse de que los tipos coincidan con las firmas de funciones sugeridas.

## Las expresiones ORDER BY deben aparecer en la lista SELECT

Mensaje de error: En el caso de SELECT DISTINCT, las expresiones ORDER BY deben aparecer en la lista SELECT

Causa: en una cláusula SELECT, se utiliza un alias de tabla incorrecto.

Solución sugerida: compruebe que todas las columnas de la expresión ORDER BY tengan las referencias adecuadas en la cláusula SELECT DISTINCT.

## Error en la consulta al comparar varias columnas devueltas por una subconsulta

Ejemplo de mensaje de error: la expresión del valor y el resultado de la subconsulta deben ser del mismo tipo: row(varchar, varchar) vs. row(row (varchar, varchar)).

Causa: debido a una actualización de sintaxis en la versión 3 del motor de Athena, este error se produce cuando una consulta intenta comparar varios valores devueltos por una subconsulta y la instrucción SELECT de la subconsulta incluye su lista de columnas entre paréntesis, como en el siguiente ejemplo.

```
SELECT *
FROM table1
WHERE (t1_col1, t1_col2)
IN (SELECT (t2_col1, t2_col2) FROM table2)
```

Solución: en la versión 3 del motor de Athena, elimine los paréntesis que rodean la lista de columnas de la instrucción SELECT de la subconsulta, como en la siguiente consulta de ejemplo actualizada.

```
SELECT *
FROM table1
WHERE (t1_col1, t1_col2)
IN (SELECT t2_col1, t2_col2 FROM table2)
```

SKIP es una palabra reservada para las consultas DML

La palabra SKIP ahora es una palabra reservada para consultas DML como SELECT. Para utilizar SKIP como identificador en una consulta DML, escríbala entre comillas dobles.

Para obtener más información acerca de palabras reservadas en Athena, consulte [Palabras clave reservadas](#).

Las cláusulas SYSTEM\_TIME y SYSTEM\_VERSION están obsoletas para los desplazamientos en el tiempo

Mensaje de error: mismatched input 'SYSTEM\_TIME'. Expecting: 'TIMESTAMP', 'VERSION' (La entrada "SYSTEM\_TIME" no coincide. Valor esperado: "TIMESTAMP", "VERSION").

Causa: en la versión 2 del motor Athena, las tablas de Iceberg utilizaban las cláusulas FOR SYSTEM\_TIME AS OF y FOR SYSTEM\_VERSION AS OF para la marca de tiempo y el desplazamiento en el tiempo de la versión. La versión 3 del motor Athena utiliza las cláusulas FOR TIMESTAMP AS OF y FOR VERSION AS OF.

Solución sugerida: actualice la consulta SQL para utilizar las cláusulas `TIMESTAMP AS OF` y `VERSION AS OF` para las operaciones de desplazamiento en el tiempo, como en los siguientes ejemplos.

Desplazamiento en el tiempo por marca de tiempo:

```
SELECT * FROM TABLE FOR TIMESTAMP AS OF (current_timestamp - interval '1' day)
```

Desplazamiento en el tiempo por versión:

```
SELECT * FROM TABLE FOR VERSION AS OF 949530903748831860
```

Demasiados argumentos para un constructor de matrices

Mensaje de error: `TOO_MANY_ARGUMENTS`: hay demasiados argumentos para el constructor de matrices.

Causa: el número máximo de elementos en un constructor de matrices ahora está establecido en 254.

Solución sugerida: divida los elementos en varias matrices que tengan 254 o menos elementos cada una y utilice la función `CONCAT` para concatenar las matrices, como en el siguiente ejemplo.

```
CONCAT(  
  ARRAY[x1, x2, x3 . . . x254],  
  ARRAY[y1, y2, y3 . . . y254],  
  . . .  
)
```

No se permite el identificador delimitado por longitud cero

Mensaje de error: No se permite el identificador delimitado por longitud cero.

Causa: una consulta utilizaba una cadena vacía como alias de columna.

Solución sugerida: actualice la consulta para usar un alias que no esté vacío en la columna.

Cambios en el procesamiento de datos

Validación de buckets

Mensaje de error: `HIVE_INVALID_BUCKET_FILES`: la tabla de Hive está dañada.

**Causa:** es posible que la tabla esté dañada. La versión 3 del motor de Athena permite una validación adicional en las tablas agrupadas en buckets para garantizar la corrección de las consultas y evitar errores inesperados en el tiempo de ejecución.

**Solución sugerida:** vuelva a crear la tabla con la versión 3 del motor de Athena.

Cuando se convierte una estructura a JSON ahora se devuelven los nombres de los campos

Cuando se convierte una estructura `struct` a JSON en una consulta `SELECT` en la versión 3 del motor Athena, la conversión ahora devuelve tanto los nombres de los campos como los valores; por ejemplo, `"useragent":null` en lugar de solo los valores (como `null`).

Cambio de aplicación de medidas de seguridad de nivel de columna de tablas de Iceberg

**Mensaje de error:** Acceso denegado: no se puede hacer una selección en las columnas

**Causa:** la tabla de Iceberg no se creó con Athena y utiliza una versión del [SDK de Apache Iceberg](#) anterior a la 0.13.0. Como las versiones anteriores del SDK no rellenan las columnas en AWS Glue, Lake Formation no pudo determinar las columnas autorizadas para el acceso.

**Solución sugerida:** realice una actualización mediante la instrucción [ALTER TABLE SET PROPERTIES](#) de Athena o utilice el SDK de Iceberg más reciente para corregir la tabla y actualizar la información de la columna en AWS Glue.

Los valores nulos en los tipos de datos de la lista ahora se propagan a las UDF.

**Mensaje de error:** Excepción de puntero nulo

**Causa:** este problema puede afectarle si usa el conector de la UDF y ha implementado una función de Lambda definida por el usuario.

La versión 2 del motor Athena filtró los valores nulos de los tipos de datos de la lista que se pasaron a una función definida por el usuario. En la versión 3 del motor Athena, los valores nulos ahora se conservan y se transfieren a la UDF. Esto puede provocar una excepción de puntero nulo si la UDF intenta anular la referencia al elemento nulo sin comprobarlo.

Por ejemplo, si tiene los datos `[null, 1, null, 2, 3, 4]` en un origen de datos, como DynamoDB, se pasa lo siguiente a la función de Lambda definida por el usuario:

Versión 2 del motor Athena: `[1, 2, 3, 4]`

Versión 3 del motor Athena: `[null, 1, null, 2, 3, 4]`



Solución sugerida: asegúrese de que la función de Lambda definida por el usuario gestione los elementos nulos en los tipos de datos de la lista.

Las subcadenas de las matrices de caracteres ya no contienen espacios rellenos

Mensaje de error: no se produce ningún error, pero la cadena devuelta ya no contiene espacios rellenos. Por ejemplo, `substr(char[20], 1, 100)` devuelve ahora una cadena con una longitud de 20 en lugar de 100.

Solución sugerida: no se requiere hacer nada.

Coacción de tipo columna decimal no compatible

Mensajes de error: HIVE\_CURSOR\_ERROR: No se pudo leer el archivo Parquet: s3://DOC-EXAMPLE-BUCKET/*path/file\_name*.parquet o Tipo de columna no admitido (varchar) para la columna Parquet (*column\_name*)

Causa: la versión 2 del motor de Athena tenía éxito en ocasiones (pero fallaba con frecuencia) al intentar coercionar tipos de datos de `varchar` a decimal. Como la versión 3 del motor de Athena tiene una validación de tipos que comprueba que el tipo sea compatible antes de intentar leer el valor, estos intentos de coerción ahora siempre fallan.

Solución sugerida: tanto para la versión 2 como para la versión 3 del motor de Athena, modifique el esquema en AWS Glue a fin de utilizar un tipo de datos numéricos en lugar de `varchar` para columnas decimales en los archivos Parquet. Vuelva a rastrear los datos y asegúrese de que el nuevo tipo de datos de la columna sea de tipo decimal o vuelva a crear manualmente la tabla en Athena y utilice la sintaxis `decimal(precision, scale)` a fin de especificar un tipo de datos [decimal](#) para la columna.

Los valores NaN flotante o doble ya no se pueden convertir a `bigint`

Mensaje de error: INVALID\_CAST\_ARGUMENT: no se puede convertir NaN real/doble a `bigint`

Causa: en la versión 3 del motor de Athena, ya no se puede convertir NaN en 0 como `bigint`.

Solución sugerida: asegúrese de que los valores NaN no estén presentes en las columnas `float` o `double` al realizar la conversión a `bigint`.

Cambio en el tipo de devolución de la función `uuid()`

El siguiente problema afecta tanto a las tablas como a las vistas.

Mensaje de error: Unsupported Hive type: uuid (Tipo de Hive no compatible: uuid).

Causa: en la versión 2 del motor de Athena, la función `uuid()` devolvía una cadena, pero en la versión 3 del motor de Athena, devuelve un UUID generado de forma pseudoaleatoria (tipo 4). Como Athena no admite el tipo de datos de la columna UUID, la función `uuid()` ya no se puede utilizar directamente en las consultas CTAS para generar columnas UUID en la versión 3 del motor de Athena.

Por ejemplo, la siguiente instrucción `CREATE TABLE` se completa correctamente en la versión 2 del motor de Athena, pero devuelve `NOT_SUPPORTED: tipo de Hive no compatible: uuid` en la versión 3 del motor de Athena:

```
CREATE TABLE uuid_table AS
  SELECT uuid() AS myuuid
```

Por ejemplo, la siguiente instrucción `CREATE VIEW` se completa correctamente en la versión 2 del motor de Athena, pero devuelve `Invalid column type for column myuuid: Unsupported Hive type: uuid (Tipo de columna no válido para la columna myuuid: tipo de Hive no compatible: uuid)` en la versión 3 del motor de Athena:

```
CREATE VIEW uuid_view AS
  SELECT uuid() AS myuuid
```

Cuando se consulta una vista creada de esta forma con la versión 2 del motor de Athena en la versión 3, se produce un error como el siguiente:

`VIEW_IS_STALE: línea 1:15: la vista "awsdatacatalog.mydatabase.uuid_view" está obsoleta o en un estado no válido: la columna [myuuid] de tipo uuid proyectada desde la vista de consulta en la posición 0 no se puede convertir en la columna [myuuid] de tipo varchar almacenada en la definición de la vista`

Solución sugerida: al crear la tabla o vista, utilice la función `cast()` para convertir la salida de `uuid()` a `varchar`, como en los siguientes ejemplos:

```
CREATE TABLE uuid_table AS
  SELECT CAST(uuid() AS VARCHAR) AS myuuid
```

```
CREATE VIEW uuid_view AS
```

```
SELECT CAST(uuid() AS VARCHAR) AS myuuid
```

## Problemas de coerción de CHAR y VARCHAR

Utilice las soluciones alternativas de esta sección si tiene problemas de coerción con `varchar` y `char` en la versión 3 del motor de Athena. Si no puede utilizar estas soluciones alternativas, póngase en contacto con AWS Support.

### Error en la función CONCAT con entradas CHAR y VARCHAR mixtas

Problema: la siguiente consulta se realiza correctamente en la versión 2 del motor de Athena.

```
SELECT concat(CAST('abc' AS VARCHAR(20)), '12', CAST('a' AS CHAR(1)))
```

Sin embargo, en la versión 3 del motor de Athena, la misma consulta falla y ocurre lo siguiente:

Mensaje de error: FUNCTION\_NOT\_FOUND: line 1:8: Unexpected parameters (varchar(20), varchar(2), char(1)) for function concat. Expected: concat(char(x), char(y)), concat(array(E), E) E, concat(E, array(E)) E, concat(array(E)) E, concat(varchar), concat(varbinary) [FUNCTION\_NOT\_FOUND: línea 1:8: parámetros inesperados (varchar(20), varchar(2), char(1)) para la función concat. Parámetros esperados: concat(char(x), char(y)), concat(array(E), E) E, concat(E, array(E)) E, concat(array(E)) E, concat(varchar), concat(varbinary)].

Solución sugerida: al usar la función `concat`, cámbiela a `char` o `varchar`, pero no a una combinación de ambas.

### Error de concatenación de || SQL con las entradas CHAR y VARCHAR

En la versión 3 del motor de Athena, el operador de concatenación `||` de barras verticales dobles requiere `varchar` como entradas. Las entradas no pueden ser una combinación de los tipos `varchar` y `char`.

Mensaje de error: TYPE\_NOT\_FOUND: line 1:26: Unknown type: char(65537) [TYPE\_NOT\_FOUND: línea 1:26: tipo desconocido: char(65537)].

Causa: una consulta que usa `||` para concatenar `char` y `varchar` puede producir el error, como en el siguiente ejemplo.

```
SELECT CAST('a' AS CHAR) || CAST('b' AS VARCHAR)
```

Solución sugerida: concatenar `varchar` con `varchar`, como en el siguiente ejemplo.

```
SELECT CAST('a' AS VARCHAR) || CAST('b' AS VARCHAR)
```

## Error de consulta CHAR y VARCHAR UNION

Mensaje de error: NOT\_SUPPORTED: Unsupported Hive type: char(65536). Supported CHAR types: CHAR(<=255) [NOT\_SUPPORTED: tipo de Hive no compatible: char(65536). Tipos de CHAR compatibles: CHAR(<=255)].

Causa: consulta que intenta combinar `char` y `varchar`, como en el siguiente ejemplo:

```
CREATE TABLE t1 (c1) AS SELECT CAST('a' as CHAR) as c1 UNION ALL SELECT CAST('b' AS VARCHAR) AS c1
```

Solución sugerida: en la consulta de ejemplo, utilice `'a'` como `varchar` en lugar de `char`.

## Espacios vacíos no deseados tras la coerción de CHAR o VARCHAR

En la versión 3 del motor de Athena, cuando los datos `char(X)` y `varchar` se convierten en un solo tipo al formar una matriz o una sola columna, `char(65535)` es el tipo objetivo y cada campo contiene muchos espacios finales no deseados.

Causa: la versión 3 del motor de Athena convierte `varchar` y `char(X)` en `char(65535)` y luego rellena los datos con espacios.

Solución sugerida: convertir cada campo de forma explícita en `varchar`.

## Cambios en la marca de tiempo

Conversión de una marca de tiempo con zona horaria para cambiar el comportamiento de `varchar`

En la versión 2 del motor Athena, al convertir una `Timestamp` con zona horaria a `varchar`, se modificaron algunos literales de zona horaria (por ejemplo, `US/Eastern` cambió a `America/New_York`). Este comportamiento no se produce en la versión 3 del motor Athena.

El desbordamiento de la marca de tiempo de fecha y hora devuelve un error

Mensaje de error: Desbordamiento de milisegundos: XXX

Causa: dado que en la versión 2 del motor Athena no se comprobó la existencia de desbordamientos en las fechas ISO 8601, algunas fechas produjeron una marca de tiempo negativa. La versión 3 del motor Athena comprueba este desbordamiento y devuelve una excepción.

Solución sugerida: asegúrese de que la marca de tiempo esté dentro del intervalo.

No se admiten zonas horarias políticas con TIME

Mensaje de error: LITERAL NO VÁLIDO

Causa: consultas como `SELECT TIME '13:21:32.424 America/Los_Angeles'`.

Solución sugerida: evite usar zonas horarias políticas con TIME.

La falta de coincidencia de precisión en las columnas de marca de tiempo provoca un error de serialización

Mensaje de error: `SERIALIZATION_ERROR`: no se pudo serializar la columna "`COLUMNZ`" de tipo "timestamp(3)" en la posición `X:Y`

`COLUMNZ` es el nombre de salida de la columna que causa el problema. Los números `X: Y` indican la posición de la columna en la salida.

Causa: la versión 3 del motor Athena comprueba que la precisión de las marcas de tiempo de los datos sea la misma que la precisión especificada para el tipo de datos de la columna en la especificación de la tabla. Actualmente, esta precisión es siempre de 3. Si los datos tienen una precisión superior a esta, las consultas fallan y se produce el error observado.

Solución sugerida: compruebe los datos para asegurarse de que las marcas de tiempo tengan una precisión de milisegundos.

La precisión de la marca de tiempo es incorrecta en las consultas UNLOAD y CTAS para las tablas de Iceberg

Mensaje de error: Precisión de la marca de tiempo incorrecta para timestamp (6); la precisión configurada es `MILLISECONDS`.

Causa: la versión 3 del motor Athena comprueba que la precisión de las marcas de tiempo de los datos sea la misma que la precisión especificada para el tipo de datos de la columna en la especificación de la tabla. Actualmente, esta precisión es siempre de 3. Si los datos tienen una precisión superior a esta (por ejemplo, microsegundos en lugar de milisegundos), las consultas pueden fallar y arrojar el error mencionado.

Solución: a fin de solucionar este problema, primero utilice `CAST` para la precisión de la marca de tiempo en 6, como en el siguiente ejemplo de CTAS, en el que se crea una tabla de Iceberg. Tenga

en cuenta que la precisión debe especificarse como 6 en lugar de 3 para evitar el error La precisión de la marca de tiempo (3) no es compatible con Iceberg.

```
CREATE TABLE my_iceberg_ctas
WITH (table_type = 'ICEBERG', location = 's3://DOC-EXAMPLE-BUCKET/table_ctas/',
format = 'PARQUET')
AS SELECT id, CAST(dt AS timestamp(6)) AS "dt"
FROM my_iceberg
```

A continuación, dado que Athena no admite la marca de tiempo 6, vuelva a convertir el valor en marca de tiempo (por ejemplo, en una vista). En el siguiente ejemplo se crea una vista de la tabla `my_iceberg_ctas`.

```
CREATE OR REPLACE VIEW my_iceberg_ctas_view AS
SELECT cast(dt AS timestamp) AS dt
FROM my_iceberg_ctas
```

Ahora se produce un error de archivo de ORC con formato incorrecto al leer el tipo Long como marca de tiempo o viceversa en los archivos de ORC

Mensaje de error: Error al abrir el archivo de ORC con formato incorrecto "FILE (SPLIT POSITION)" dividido en Hive. No se puede leer la marca de tiempo de tipo SQL del flujo de ORC .long\_type de tipo LONG

Causa: la versión 3 del motor de Athena ahora rechaza la coerción implícita del tipo de datos Long a Timestamp o de Timestamp a Long. Anteriormente, los valores Long se convertían implícitamente en marcas de tiempo como si fueran milisegundos de epoch.

Solución sugerida: utilice la función `from_unixtime` para convertir la columna de forma explícita o utilice la función `from_unixtime` a fin de crear una columna adicional para consultas futuras.

No se admiten los intervalos de tiempo a mes

Mensaje de error: EL TIPO NO COINCIDE

Causa: la versión 3 del motor Athena no admite el tiempo ni el intervalo de un año a otro (por ejemplo, `SELECT TIME '01:00' + INTERVAL '3' MONTH`).

Desbordamiento de marca de tiempo para el formato int96 de Parquet

Mensaje de error: TimeOfDayNanos no válido

Causa: un desbordamiento de marca de tiempo para el formato de Parquet `int96`.

Solución sugerida: identifique los archivos específicos que tienen el problema. A continuación, vuelva a generar el archivo de datos con una biblioteca de Parquet actualizada y conocida, o utilice Athena CTAS. Si el problema persiste, contacte con el servicio de asistencia de Athena y explíquenos cómo se generan los archivos de datos.

Se requiere espacio entre los valores de fecha y hora al convertir una cadena en una marca de tiempo

Mensaje de error: `INVALID_CAST_ARGUMENT`: el valor no se puede convertir en una marca de tiempo.

Causa: la versión 3 del motor de Athena ya no acepta un guion como separador válido entre los valores de fecha y hora de la cadena de entrada para `cast`. Por ejemplo, la siguiente consulta funciona en la versión 2 del motor de Athena, pero no en la versión 3:

```
SELECT CAST('2021-06-06-23:38:46' AS timestamp) AS this_time
```

Solución sugerida: en la versión 3 del motor de Athena, sustituya el guion entre la fecha y la hora por un espacio, como en el siguiente ejemplo.

```
SELECT CAST('2021-06-06 23:38:46' AS timestamp) AS this_time
```

Cambio en el valor de devolución de la marca de tiempo `to_iso8601()`

Mensaje de error: ninguno.

Causa: en la versión 2 del motor de Athena, la función `to_iso8601` devuelve una marca de tiempo con la zona horaria incluso si el valor pasado a la función no incluye la zona horaria. En la versión 3 del motor de Athena, la función `to_iso8601` devuelve una marca de tiempo con la zona horaria solo cuando el argumento pasado incluye la zona horaria.

Por ejemplo, la siguiente consulta pasa la fecha actual a la función `to_iso8601` dos veces: primero como marca de tiempo con zona horaria y, después, como marca de tiempo.

```
SELECT TO_ISO8601(CAST(CURRENT_DATE AS TIMESTAMP WITH TIME ZONE)),  
       TO_ISO8601(CAST(CURRENT_DATE AS TIMESTAMP))
```

En el siguiente ejemplo se muestra el resultado de la consulta en cada motor.

Versión 2 del motor de Athena:

#	_col0	_col1
1	2023-02-24T00:00:00.000Z	2023-02-24T00:00:00.000Z

Versión 3 del motor de Athena:

#	_col0	_col1
1	2023-02-24T00:00:00.000Z	2023-02-24T00:00:00.000

Solución sugerida: para replicar el comportamiento anterior, puede pasar el valor de la marca de tiempo a la función `with_timezone` antes de pasarlo a `to_iso8601`, como en el siguiente ejemplo:

```
SELECT to_iso8601(with_timezone(TIMESTAMP '2023-01-01 00:00:00.000', 'UTC'))
```

Resultado

#	_col0
1	2023-01-01T00:00:00.000Z

El primer parámetro de `at_timezone()` debe especificar una fecha

Problema: en la versión 3 del motor de Athena, la función `at_timezone` no puede tomar un valor `time_with_timezone` como primer parámetro.

Causa: sin información de fecha, no se puede determinar si el valor transferido corresponde al horario diurno o a la hora estándar. Por ejemplo, `at_timezone('12:00:00 UTC', 'America/Los_Angeles')` es ambiguo, ya que no hay forma de determinar si el valor transferido es la hora diurna del Pacífico (PDT) o la hora estándar del Pacífico (PST).



## Limitaciones

La versión 3 del motor Athena tiene las siguientes limitaciones:

- Rendimiento de las consultas: muchas consultas se ejecutan más rápido en la versión 3 del motor Athena, pero algunos planes de consultas pueden diferir de la versión 2 del motor Athena. Como resultado, algunas consultas pueden diferir en la latencia o el costo.
- Conectores de Trino y Presto: no se admiten los conectores de [Trino](#) ni de [Presto](#). Utilice la consulta federada de Amazon Athena para conectar orígenes de datos. Para obtener más información, consulte [Uso de consulta federada de Amazon Athena](#).
- Ejecución tolerante a errores: no se admite la [ejecución tolerante a errores](#) de Trino (Trino Tardigrade).
- Límite de parámetros de las funciones: las funciones no pueden aceptar más de 127 parámetros. Para obtener más información, consulte [Hay demasiados argumentos para llamar a la función](#).

En la versión 2 del motor Athena se introdujeron los siguientes límites para garantizar que las consultas no fallen debido a limitaciones de recursos. Los usuarios no pueden configurar estos límites.

- Número de elementos de resultado: el número de elementos de resultado  $n$  está restringido a 10 000 o menos para las siguientes funciones: `min(col, n)`, `max(col, n)`, `min_by(col1, col2, n)` y `max_by(col1, col2, n)`.
- Conjuntos de agrupación: el número máximo de sectores en un conjunto de agrupación es 2048.
- Longitud máxima de línea del archivo de texto: la longitud máxima de línea predeterminada para los archivos de texto es de 200 MB.
- Tamaño máximo del resultado de la función de secuencia: el tamaño máximo de resultado de una función de secuencia es 50 000 entradas. Por ejemplo, `SELECT sequence(0, 45000, 1)` se realiza correctamente, pero `SELECT sequence(0, 55000, 1)` falla con el mensaje de error El resultado de la función de secuencia no debe tener más de 50 000 entradas. Este límite se aplica a todos los tipos de entrada de funciones de secuencia, incluidas las marcas de tiempo.

## Versión 2 del motor Athena

La versión 2 del motor de Athena presenta los siguientes cambios.

- [Mejoras y nuevas características](#)

- [Mejoras de agrupación, unión y subconsulta](#)
- [Mejoras de los tipos de datos](#)
- [Funciones agregadas](#)
- [Mejoras en el rendimiento](#)
- [Mejoras relacionadas con JSON](#)
- [Cambios bruscos](#)
  - [Correcciones de errores](#)
  - [Cambios en las funciones geoespaciales](#)
  - [Conformidad con ANSI SQL](#)
  - [Funciones reemplazadas](#)
  - [Límites](#)

## Mejoras y nuevas características

- **EXPLAIN y EXPLAIN ANALYZE:** puede utilizar la instrucción EXPLAIN en Athena para ver el plan de ejecución de las consultas SQL. Utilice EXPLAIN ANALYZE para ver el plan de ejecución distribuido y el costo de cada operación de las consultas SQL. Para obtener más información, consulte [Uso de EXPLAIN y EXPLAIN ANALYZE en Athena](#).
- **Consultas federadas:** en la versión 2 del motor Athena se admiten las consultas federadas. Para obtener más información, consulte [Uso de consulta federada de Amazon Athena](#).
- **Funciones geoespaciales:** se han agregado más de 25 funciones geoespaciales. Para obtener más información, consulte [Nuevas funciones geoespaciales en la versión 2 del motor Athena](#).
- **Esquema anidado:** se ha agregado compatibilidad para leer el esquema anidado, lo que reduce el costo.
- **Instrucciones preparadas:** utilice instrucciones preparadas para la ejecución repetida de la misma consulta con parámetros de consulta diferentes. Una instrucción preparada contiene parámetros de marcador de posición cuyos valores se pasan en tiempo de ejecución. Las instrucciones preparadas ayudan a prevenir los ataques de inyección SQL. Para obtener más información, consulte [Uso de consultas parametrizadas](#).
- **Compatibilidad con la evolución del esquema:** se ha agregado compatibilidad con evolución de esquemas para datos en formato Parquet.
  - Se agregó compatibilidad para leer columnas de matriz, mapa o fila de particiones donde el esquema de partición es diferente del esquema de tabla. Esto puede suceder cuando el

esquema de tabla se actualizó después de crear la partición. Los tipos de columna modificados deben ser compatibles. Para los tipos de fila, es posible agregar o eliminar campos finales, pero los campos correspondientes (por ordinal) deben tener el mismo nombre.

- Los archivos ORC ahora pueden tener columnas de estructura con campos faltantes. Esto permite cambiar el esquema de tabla sin volver a escribir los archivos ORC.
- Las columnas de estructura ORC ahora se asignan por nombre en lugar de ordinal. Esto maneja adecuadamente los campos de estructura faltantes o adicionales en el archivo ORC.
- SQL OFFSET: la cláusula OFFSET de SQL ahora es compatible con las instrucciones SELECT. Para obtener más información, consulte [SELECT](#).
- Instrucción UNLOAD: puede utilizar la instrucción UNLOAD para escribir la salida de una consulta SELECT en los formatos PARQUET, ORC, AVRO y JSON. Para obtener más información, consulte [UNLOAD](#).

### Mejoras de agrupación, unión y subconsulta

- Agrupación compleja: se ha agregado compatibilidad con operaciones de agrupación complejas.
- Subconsultas correlacionadas: se agregó compatibilidad para subconsultas correlacionadas en IN y para subconsultas correlacionadas que requieren coerciones.
- Combinación cruzada: se ha agregado compatibilidad para CROSS JOIN en tablas derivadas LATERAL.
- Conjuntos de agrupación: se ha agregado compatibilidad para cláusulas ORDER BY en agregaciones para consultas que utilizan GROUPING SETS.
- Expresiones de Lambda: se agregó compatibilidad para desreferenciar campos de fila en expresiones Lambda.
- Valores nulos en semiuniones: se agregó compatibilidad para valores nulos en el lado izquierdo de una semiunión (es decir, un predicado IN con subconsultas).
- Combinaciones espaciales: se agregó compatibilidad para uniones espaciales de difusión y uniones espaciales izquierdas.
- Vertido en disco: para operaciones INNER JOIN y LEFT JOIN de uso intensivo de memoria, Athena descarga los resultados de la operación intermedia en el disco. Esto permite la ejecución de consultas que requieren grandes cantidades de memoria.

## Mejoras de los tipos de datos

- **INT para INTEGER:** se ha agregado compatibilidad para INT como alias para el tipo de datos INTEGER.
- **Tipos de intervalo:** se ha agregado compatibilidad para conversión a tipos INTERVAL.
- **IPADDRESS:** se ha agregado un nuevo tipo IPADDRESS para representar direcciones IP en consultas DML. Se ha agregado compatibilidad de conversión entre tipo VARBINARY y IPADDRESS. El tipo IPADDRESS no se reconoce en las consultas DDL.
- **Es diferente de:** se ha agregado compatibilidad IS DISTINCT FROM con tipos JSON y IPADDRESS.
- **Verificaciones de igualdad nulas:** ya son compatibles las verificaciones de igualdad para valores nulos en estructuras de datos ARRAY, MAP y ROW. Por ejemplo, la expresión `ARRAY ['1', '3', null] = ARRAY ['1', '2', null]` devuelve `false`. Anteriormente, un elemento nulo devolvía el mensaje de error Comparación no compatible.
- **Coerción de tipo de fila:** ahora se permite la coerción entre tipos de filas independientemente de los nombres de campo. Anteriormente, un tipo de fila era coercible a otro solo si el nombre del campo del tipo de origen coincidía con el tipo de destino o cuando el tipo de destino tenía un nombre de campo anónimo.
- **Resta de tiempo:** resta implementada para todos los tipos TIME y TIMESTAMP.
- **Unicode:** se agregó compatibilidad para secuencias Unicode de escape en literales de cadena.
- **Concatenación VARBINARY:** se ha agregado compatibilidad con concatenación de valores VARBINARY.

Funciones de valor de ventana: las funciones de valor de ventana ahora son compatibles con IGNORE NULLS y RESPECT NULLS.

## Tipos de entrada adicionales para funciones

Las siguientes funciones ahora aceptan tipos de entrada adicionales. Para obtener más información acerca de cada función, visite el enlace correspondiente a la documentación de Presto.

- `approx_distinct()`: la función [approx\\_distinct\(\)](#) admite ahora los siguientes tipos: INTEGER, SMALLINT, TINYINT, DECIMAL, REAL, DATE, TIMESTAMP, TIMESTAMP WITH TIME ZONE, TIME, TIME WITH TIME ZONE, IPADDRESS y CHAR.
- `Avg()`, `sum()`: las funciones de agregación [avg\(\)](#) y [sum\(\)](#) ahora son compatibles con el tipo de datos INTERVAL.

- `Lpad()`, `rpad()`: las funciones [lpad](#) y [rpad](#) ahora funcionan en entradas VARBINARY.
- `Min()`, `max()`: las funciones de agregación [min\(\)](#) y [max\(\)](#) ahora permiten tipos de entrada desconocidos en el momento del análisis de consultas para que pueda utilizar las funciones con literales NULL.
- `regexp_replace()`: variante de la función [regexp\\_replace\(\)](#) que puede ejecutar una función Lambda para cada reemplazo.
- `Sequence()`: se agregaron variantes DATE de la función [sequence\(\)](#), incluida la variante con un incremento implícito de paso de un día.
- `ST_Area()`: la función geoespacial [ST\\_Area\(\)](#) ahora admite todos los tipos de geometría.
- `Substr()`: la función [substr](#) ahora funciona en entradas VARBINARY.
- `zip_with()`: ahora se pueden usar las matrices de longitud no coincidente con [zip\\_with\(\)](#). Las posiciones faltantes se rellenan con nulo. Anteriormente, cuando se pasaban matrices de diferentes longitudes, se generaba un error. Este cambio puede dificultar la distinción entre los valores que originalmente eran nulos de los valores que se agregaron para rellenas las matrices con la misma longitud.

## Funciones agregadas

La siguiente lista contiene funciones nuevas desde la versión 2 del motor Athena. La lista no incluye funciones geoespaciales. Para obtener una lista de funciones geoespaciales, consulte [Nuevas funciones geoespaciales en la versión 2 del motor Athena](#).

Para obtener más información acerca de cada función, visite el enlace correspondiente a la documentación de Presto.

## Funciones de agregación

### [reduce\\_agg\(\)](#)

## Array functions and operators (Funciones y operadores de matriz)

[array\\_sort\(\)](#): variante de esta función agregada que toma una función Lambda como comparador.

### [ngrams\(\)](#)

## Funciones binarias y operadores

### [from\\_big\\_endian\\_32\(\)](#)

### [from\\_ieee754\\_32\(\)](#)

[from\\_ieee754\\_64\(\)](#)

[hmac\\_md5\(\)](#)

[hmac\\_sha1\(\)](#)

[hmac\\_sha256\(\)](#)

[hmac\\_sha512\(\)](#)

[spooky\\_hash\\_v2\\_32\(\)](#)

[spooky\\_hash\\_v2\\_64\(\)](#)

[to\\_big\\_endian\\_32\(\)](#)

[to\\_ieee754\\_32\(\)](#)

[to\\_ieee754\\_64\(\)](#)

Funciones y operadores de fecha y hora

[millisecond\(\)](#)

[parse\\_duration\(\)](#)

[to\\_milliseconds\(\)](#)

Funciones y operadores de mapa

[multimap\\_from\\_entries\(\)](#)

Funciones y operadores matemáticos

[inverse\\_normal\\_cdf\(\)](#)

[wilson\\_interval\\_lower\(\)](#)

[wilson\\_interval\\_upper\(\)](#)

Funciones de resumen de cuantiles

[Funciones de resumen de cuantiles](#) y el tipo de resumen de cuantil `qdigest` agregado.

Funciones y operadores de cadena

[hamming\\_distance\(\)](#)

## [split\\_to\\_multimap\(\)](#)

### Mejoras en el rendimiento

En la versión 2 del motor Athena, se ha mejorado el rendimiento de las siguientes características.

#### Rendimiento de las consultas

- Rendimiento de uniones de difusión: se mejoró el rendimiento de las uniones de difusión mediante la aplicación de la eliminación de particiones dinámica en el nodo de trabajo.
- Tablas en buckets: rendimiento mejorado para escribir en tablas en buckets cuando los datos que se escriben ya están particionados de forma adecuada (por ejemplo, cuando la salida es de una unión en bucket).
- DISTINCT: rendimiento mejorado para algunas consultas que utilizan DISTINCT.

Filtrado dinámico y poda de particiones: las mejoras aumentan el rendimiento y reducen la cantidad de datos analizados en las consultas.

- Operaciones de filtrado y proyección: ahora las operaciones de filtrado y proyección se procesan siempre por columnas, de ser posible. El motor aprovecha automáticamente las codificaciones de diccionario cuando es eficaz.
- Intercambios de recopilación: rendimiento mejorado para consultas con intercambio de recopilación.
- Agregaciones globales: rendimiento mejorado para algunas consultas que realizan agregaciones globales filtradas.
- Conjuntos de agrupación, cube, rollup: rendimiento mejorado para consultas que implican GROUPING SETS, CUBE o ROLLUP, que puede utilizar para agregar varios conjuntos de columnas en una sola consulta.
- Filtros altamente selectivos: se ha mejorado el rendimiento de las consultas con filtros altamente selectivos.
- Operaciones JOIN y AGGREGATE: se ha mejorado el rendimiento de las operaciones JOIN y AGGREGATE.
- Like: se ha mejorado el rendimiento de las consultas que utilizan predicados LIKE en las columnas de tablas `information_schema`.
- Ordenar por y límite: planes, rendimiento y uso de memoria mejorados para las consultas que implican ORDER BY y LIMIT para evitar intercambios de datos innecesarios.

- Ordenar por: las operaciones ORDER BY ahora se distribuyen de forma predeterminada, lo que permite el uso de cláusulas ORDER BY más extensas.
- Conversiones de tipo fila: rendimiento mejorado al convertir entre tipos de ROW.
- Tipos estructurales: rendimiento mejorado de consultas que procesan tipos estructurales y contienen análisis, uniones, agregaciones o escrituras de tablas.
- Análisis de tablas: se presentó una nueva regla de optimización para evitar análisis de tablas duplicados en determinados casos.
- Unión: se ha mejorado el rendimiento de consultas UNION.

### Rendimiento de planificación de consultas

- Rendimiento de la planificación: se ha mejorado el rendimiento de la planificación de consultas que unen varias tablas con un gran número de columnas.
- Evaluaciones de predicados: mejora del rendimiento de la evaluación de predicados durante la inserción de predicados en la planificación.
- Compatibilidad de inserción de predicados para conversión: compatibilidad de inserción de predicados para el predicado `<column> IN <values list>` donde los valores de la lista de valores requieren conversión para que coincidan con el tipo de columna.
- Inferencia e inserción de predicados: inferencia e inserción de predicados extendidos para consultas que utilizan un predicado `<symbol> IN <subquery>`.
- Tiempos de espera: se corrigió un error que, en raras ocasiones, podía provocar tiempos de espera de planificación de consultas.

### Rendimiento de uniones

- Uniones con columnas de mapa: se ha mejorado el rendimiento de las uniones y agregaciones que incluyen columnas de mapa.
- Uniones solo con condiciones de no igualdad: se ha mejorado el rendimiento de uniones solo con condiciones de no igualdad mediante el uso de una unión de bucle anidado en lugar de una unión hash.
- Uniones externas: el tipo de distribución de unión ahora se selecciona automáticamente para las consultas que involucran uniones externas.
- Uniones de rango sobre una función: rendimiento mejorado de uniones donde la condición es un rango sobre una función (por ejemplo, `a JOIN b ON b.x < f(a.x) AND b.x > g(a.x)`).



- Vertido en disco: se corrigieron errores relacionados con el vertido de disco y problemas de memoria para mejorar el rendimiento y reducir los errores de memoria en operaciones JOIN.

## Rendimiento de subconsultas

- Subconsultas correlacionadas: se ha mejorado el rendimiento de subconsultas EXISTS correlacionadas.
- Subconsultas correlacionadas con igualdad: se ha mejorado la compatibilidad para subconsultas correlacionadas que contienen predicados de igualdad.
- Subconsultas correlacionadas con desigualdades: rendimiento mejorado para subconsultas correlacionadas que contienen desigualdades.
- Agregaciones count(\*) sobre subconsultas: se ha mejorado el rendimiento de agregaciones count ( \* ) sobre subconsultas con cardinalidad constante conocida.
- Propagación de filtro de consulta externa: rendimiento mejorado de subconsultas correlacionadas cuando los filtros de la consulta externa se pueden propagar a la subconsulta.

## Rendimiento de funciones

- Funciones de agregación de ventana: mejora del rendimiento de las funciones de agregación de ventana.
- element\_at(): se ha mejorado el rendimiento de element\_at ( ) para que los mapas sean de tiempo constante en lugar de proporcionales al tamaño del mapa.
- Grouping(): rendimiento mejorado para consultas que implican grouping ( ).
- JSON casting: se ha mejorado el rendimiento de la conversión de JSON a tipos ARRAY o MAP.
- Funciones de devolución de mapas: rendimiento mejorado de las funciones que devuelven mapas.
- Conversión mapa a mapa: se ha mejorado el rendimiento de la conversión de mapa a mapa.
- Min() y max(): se han optimizado las funciones min ( ) y max ( ) para evitar la creación innecesaria de objetos, lo que reduce la sobrecarga de recopilación de elementos no utilizados.
- row\_number(): rendimiento y uso de memoria mejorados para las consultas que utilizan row\_number ( ) seguido de un filtro en los números de fila generados.
- Funciones de ventana: rendimiento mejorado de consultas que contienen funciones de ventana con cláusulas PARTITION BY y ORDER BY idénticas.
- Funciones de ventana: mejora del rendimiento de ciertas funciones de ventana (por ejemplo, LAG) que tienen especificaciones similares.

## Rendimiento geoespacial

- Serialización de geometría: se ha mejorado el rendimiento de serialización de los valores de geometría.
- Funciones geoespaciales: se ha mejorado el rendimiento de `ST_Intersects()`, `ST_Contains()`, `ST_Touches()`, `ST_Within()`, `ST_Overlaps()`, `ST_Disjoint()`, `transform_values()`, `ST_XMin()`, `ST_XMax()`, `ST_YMin()`, `ST_YMax()`, `ST_Crosses()` y `array_intersect()`.
- `ST_Distance()`: mejora del rendimiento de las consultas de unión que involucran la función `ST_Distance()`.
- `ST_Intersection()`: se ha optimizado la función `ST_Intersection()` para rectángulos alineados con ejes de coordenadas (por ejemplo, polígonos producidos por las funciones `ST_Envelope()` y `bing_tile_polygon()`).

## Mejoras relacionadas con JSON

### Funciones de mapa

- Se ha mejorado el rendimiento del subíndice de mapa de  $O(n)$  a  $O(1)$  en todos los casos. Anteriormente, únicamente los mapas producidos por ciertas funciones y lectores aprovechaban esta mejora.
- Se han agregado las funciones `map_from_entries()` y `map_entries()`.

### Conversión

- Se ha agregado la capacidad de convertir JSON desde REAL, TINYINT o SMALLINT.
- Ahora puede convertir JSON a ROW incluso si el JSON no contiene todos los campos del ROW.
- Se mejoró el rendimiento de `CAST(json_parse(...) AS ...)`.
- Se ha mejorado el rendimiento de la conversión de JSON a tipos ARRAY o MAP.

### Nuevas funciones JSON

- [is\\_json\\_scalar\(\)](#)

## Cambios bruscos

Los cambios sustanciales incluyen correcciones de errores, cambios en las funciones geoespaciales, funciones reemplazadas y la introducción de límites. Las mejoras en la conformidad de ANSI SQL pueden interrumpir las consultas que dependían del comportamiento no estándar.

### Correcciones de errores

Los siguientes cambios corrigen problemas de comportamiento que provocaron que las consultas se ejecutaran correctamente, pero con resultados inexactos.

- Las columnas `fixed_len_byte_array` de parquet ahora se aceptan como DECIMALES: las consultas sobre columnas de Parquet de tipo `fixed_len_byte_array` se realizan correctamente y devuelven valores correctos si se anotan como DECIMAL en el esquema de Parquet. Las consultas en columnas `fixed_len_byte_array` sin la anotación DECIMAL fallan con un error. Anteriormente, las consultas en columnas `fixed_len_byte_array` sin la anotación decimal se realizaban correctamente, pero devolvían valores incomprensibles.
- `json_parse()` ya no ignora los caracteres finales: anteriormente, las entradas como `[1, 2]abc` se analizaban con éxito como `[1, 2]`. El uso de caracteres finales ahora genera el mensaje de error `Cannot convert '[1, 2]abc' to JSON (No se puede convertir '[1, 2]abc' a JSON)`.
- Precisión decimal `round()` corregida: `round(x, d)` ahora redondea correctamente `x` cuando `x` es un DECIMAL o cuando `x` es un DECIMAL con escala 0 y `d` es un número entero negativo. Anteriormente, no había redondeado en estos casos.
- `round(x, d)` y `truncate(x, d)`: el parámetro `d` en la firma de funciones `round(x, d)` y `truncate(x, d)` ahora es de tipo INTEGER. Anteriormente, `d` podía ser de tipo BIGINT.
- `Map()` con claves duplicadas: `map()` ahora genera un error en claves duplicadas en lugar de producir silenciosamente un mapa dañado. Las consultas que actualmente construyen valores de mapa utilizando claves duplicadas ahora fallan con un error.
- `map_from_entries()` genera un error con entradas nulas: `map_from_entries()` ahora genera un error cuando la matriz de entrada contiene una entrada nula. Ahora, las consultas que construyen un mapa con NULL como un valor fallan.
- Tablas: ya no se pueden crear tablas que tienen tipos de partición no admitidos.
- Mejora de la estabilidad numérica en funciones estadísticas: se ha mejorado la estabilidad numérica de las funciones estadísticas `corr()`, `covar_samp()`, `regr_intercept()` y `regr_slope()`.

- La precisión `TIMESTAMP` definida en Parquet ahora se respeta: la precisión de `TIMESTAMP` y la precisión definida para la columna `TIMESTAMP` en el esquema Parquet ahora deben coincidir. Las precisiones que no coincidan generan marcas de tiempo incorrectas.
- Información de zona horaria: ahora, la información de zona horaria se calcula utilizando el paquete [java.time](#) del SDK de Java 1.8.
- Suma de los tipos de datos `INTERVAL_DAY_TO_SECOND` e `INTERVAL_YEAR_TO_MONTH`: ya no puede utilizar `SUM(NULL)` directamente. Para usar `SUM(NULL)`, debe convertir `NULL` a un tipo de datos como `BIGINT`, `DECIMAL`, `REAL`, `DOUBLE`, `INTERVAL_DAY_TO_SECOND` o `INTERVAL_YEAR_TO_MONTH`.

## Cambios en las funciones geoespaciales

Los cambios realizados en las funciones geoespaciales incluyen los siguientes.

- Cambios de nombre de función: algunos nombres de funciones han cambiado. Para obtener más información, consulte [Cambios de nombre de la función geoespacial en la versión 2 del motor Athena](#).
- Entrada `VARBINARY`: el tipo `VARBINARY` ya no se admite directamente para la entrada de funciones geoespaciales. Por ejemplo, para calcular el área de una geometría directamente, ahora la geometría se debe introducir en formato `VARCHAR` o `GEOMETRY`. La solución alternativa consiste en utilizar funciones de transformación, como en los siguientes ejemplos.
  - Para utilizar `ST_area()` para calcular el área de entrada `VARBINARY` en formato binario conocido (WKB, Well-Known Binary), pase la entrada a `ST_GeomFromBinary()` primero, por ejemplo:

```
ST_area(ST_GeomFromBinary(<wkb_varbinary_value>))
```

- Para utilizar `ST_area()` para calcular el área de entrada `VARBINARY` en formato binario heredado, pase la misma entrada a la función `ST_GeomFromLegacyBinary()` primero, por ejemplo:

```
ST_area(ST_GeomFromLegacyBinary(<legacy_varbinary_value>))
```

- `ST_ExteriorRing()` y `ST_Polygon()`: [ST\\_ExteriorRing\(\)](#) y [ST\\_Polygon\(\)](#) ahora aceptan solo polígonos como entradas. Anteriormente, estas funciones aceptaban de manera errónea otras geometrías.

- `ST_Distance()`: según lo requerido por la [Especificación SQL/MM](#), la función `ST_Distance()` ahora devuelve NULL si una de las entradas es una geometría vacía. Anteriormente, se devolvía NaN.

## Conformidad con ANSI SQL

Se han corregido los siguientes problemas de sintaxis y comportamiento para seguir el estándar ANSI SQL.

- Operaciones `cast()`: las operaciones de `cast()` de REAL o DOBLE a DECIMAL ahora se ajustan al estándar SQL. Por ejemplo, `cast (double '1000000000000000000000000000000000000' as decimal(38))` antes devolvía `1000000000000000005366162204393472`, pero ahora devuelve `100000000000000000000000000000000000`.
- `JOIN... USING`: `JOIN . . . USING` ahora se ajusta a la semántica SQL estándar. Anteriormente, `JOIN . . . USING` requería calificar el nombre de la tabla en columnas, y la columna de ambas tablas estaba presente en la salida. Ahora, las calificaciones de tabla no son válidas y la columna solo está presente una vez en la salida.
- Literales de tipo fila eliminados: el formato literal de tipo fila `ROW<int, int>(1, 2)` ya no es compatible. En su lugar, utilice la sintaxis `ROW(1 int, 2 int)`.
- Semántica de agregación agrupada: las agregaciones agrupadas utilizan semántica `IS NOT DISTINCT FROM` en lugar de semántica de igualdad. Las agregaciones agrupadas ahora devuelven resultados correctos y muestran un rendimiento mejorado al agrupar en valores de punto flotante NaN. Se admite la agrupación en tipos de mapa, lista y fila que contienen valores nulos.
- Los tipos con comillas ya no están permitidos: de acuerdo con el estándar ANSI SQL, ya no se pueden incluir entre comillas los tipos de datos. Por ejemplo, `SELECT "date" '2020-02-02'` ya no es una consulta válida. En su lugar, se debe utilizar la sintaxis `SELECT date '2020-02-02'`.
- Acceso al campo de fila anónimo: ya no se puede acceder a los campos de fila anónimos utilizando la sintaxis `[.field0, .field1, ...]`.
- Operaciones de agrupación complejas: las operaciones de agrupación complejas `GROUPING SETS`, `CUBE` y `ROLLUP` no admiten agrupación en expresiones compuestas de columnas de entrada. Solo se admiten nombres de columnas.

## Funciones reemplazadas

Las siguientes funciones ya no son compatibles y se han reemplazado por sintaxis que generan los mismos resultados.

- `information_schema.__internal_partitions__`: el uso de `__internal_partitions__` ya no es compatible con la versión 2 del motor de Athena. Para una sintaxis equivalente, utilice `SELECT * FROM "<table_name>$partitions"` o `SHOW PARTITIONS`. Para obtener más información, consulte [Enumeración de las particiones de una tabla específica](#).
- Funciones geoespaciales reemplazadas: para obtener una lista de funciones geoespaciales cuyos nombres han cambiado, consulte [Cambios de nombre de la función geoespacial en la versión 2 del motor Athena](#).

## Límites

En la versión 2 del motor Athena se introdujeron los siguientes límites para garantizar que las consultas no fallen debido a limitaciones de recursos. Los usuarios no pueden configurar estos límites.

- Número de elementos de resultado: el número de elementos de resultado `n` está restringido a 10 000 o menos para las siguientes funciones: `min(col, n)`, `max(col, n)`, `min_by(col1, col2, n)` y `max_by(col1, col2, n)`.
- Conjuntos de agrupación: el número máximo de sectores en un conjunto de agrupación es 2048.
- Longitud máxima de línea del archivo de texto: la longitud máxima de línea predeterminada para los archivos de texto es de 200 MB.
- Tamaño máximo del resultado de la función de secuencia: el tamaño máximo de resultado de una función de secuencia es 50 000 entradas. Por ejemplo, `SELECT sequence(0, 45000, 1)` se realiza correctamente, pero `SELECT sequence(0, 55000, 1)` falla con el mensaje de error El resultado de la función de secuencia no debe tener más de 50 000 entradas. Este límite se aplica a todos los tipos de entrada de funciones de secuencia, incluidas las marcas de tiempo.

## Referencia de SQL para Athena

Amazon Athena admite un subconjunto de instrucciones, funciones, operadores y tipos de datos del lenguaje de definición de datos (DDL, Data Definition Language) y el lenguaje de manipulación de datos (DML, Data Manipulation Language). Salvo algunas excepciones, DDL de Athena se basa en

[DDL de HiveQL](#) y DML de Athena se basa en [Trino](#). Para obtener más información acerca de las versiones de motor Athena, consulte [Control de versiones del motor Athena](#).

## Temas

- [Tipos de datos en Amazon Athena](#)
- [Consultas, funciones y operadores de DML](#)
- [Instrucciones DDL](#)
- [Consideraciones y limitaciones de las consultas SQL en Amazon Athena](#)

## Tipos de datos en Amazon Athena

Cuando ejecute `CREATE TABLE`, deberá especificar los nombres de las columnas y el tipo de datos que puede contener cada columna. Las tablas que cree se almacenan en el AWS Glue Data Catalog.

Para facilitar la interoperabilidad con otros motores de consultas, Athena usa nombres de tipos de datos de [Apache Hive](#) para instrucciones DDL como `CREATE TABLE`. Para consultas de DML como `SELECT`, `CTAS` y `INSERT INTO`, Athena usa nombres de tipos de datos de [Trino](#). En la siguiente tabla, se muestran los tipos de datos admitidos en Athena. Cuando los tipos DDL y DML difieren en el nombre, la disponibilidad o la sintaxis, se muestran en columnas independientes.

DDL	DML	Descripción
BOOLEAN		Los valores son <code>true</code> y <code>false</code> .
	TINYINT	Un número entero firmado de 8 bits en formato de complemento de dos, con un valor mínimo de $-2^7$ y un valor máximo de $2^7-1$ .
	SMALLINT	Un número entero firmado de 16 bits en formato de complemento de dos, con un valor mínimo de $-2^{15}$ y un valor máximo de $2^{15}-1$ .
	INT, INTEGER	Un firmado de 32 bits en formato de complemento de dos, con un valor mínimo de $-2^{31}$ y un valor máximo de $2^{31}-1$ .

DDL	DML	Descripción
BIGINT		Un número entero firmado de 64 bits en formato de complemento de dos, con un valor mínimo de $-2^{63}$ y un valor máximo de $2^{63}-1$ .
FLOAT	REAL	Un número firmado de punto flotante de precisión simple de 32 bits. El rango es de $1.40129846432481707e-45$ a $3.40282346638528860e+38$ , positivo o negativo. Sigue el Estándar IEEE para Aritmética de Punto Flotante (IEEE 754).
DOBLE		Un número firmado de punto flotante de doble precisión de 64 bits. El rango es de $4.94065645841246544e-324d$ a $1.79769313486231570e+308d$ , positivo o negativo. Sigue el Estándar IEEE para Aritmética de Punto Flotante (IEEE 754).
DECIMAL ( <i>precisión</i> , <i>escala</i> )		<i>precisión</i> es el número total de dígitos. <i>scale</i> (opcional) es el número de dígitos de la parte fraccional con un valor predeterminado de 0. Por ejemplo, utilice estas definiciones de tipo: <code>decimal(11,5)</code> , <code>decimal(15)</code> . El valor máximo para <i>precisión</i> es 38 y el valor máximo para <i>escala</i> también es 38.
CHAR, CHAR ( <i>longitud</i> )		Datos de caracteres de longitud fija, con una longitud especificada comprendida entre 1 y 255 como, por ejemplo, <code>car(10)</code> . Si se especifica la <i>longitud</i> , las cadenas se truncan en la longitud especificada cuando se leen. Si la cadena de datos subyacente es más larga, esta permanece sin cambios.  Para obtener más información, consulte <a href="#">Tipo de datos Hive CHAR</a> .
STRING	VARCHAR	Datos de caracteres de longitud variable.



DDL	DML	Descripción
VARCHAR ( <i>longitud</i> )		Datos de caracteres de longitud variable con una longitud máxima de lectura. Las cadenas se truncan con la longitud especificada cuando se leen. Si la cadena de datos subyacente es más larga, esta permanece sin cambios.
BINARIO	VARBINARY	Datos binarios de longitud variable.
HORA		Hora del día con precisión de milisegundos.
No disponible	HORA ( <i>precisión</i> )	Una hora del día con una precisión específica. TIME(3) es equivalente a TIME.
No disponible	HORA CON ZONA HORARIA	Hora del día en una zona horaria. Las zonas horarias deben especificarse como desfases con respecto a la UTC.
FECHA		Una fecha de calendario con el año, el mes y el día.
MARCA DE TIEMPO	MARCA DE TIEMPO, MARCA DE TIEMPO SIN ZONA HORARIA	Una fecha y hora del día del calendario con una precisión de milisegundos.
No disponible	MARCA DE TIEMPO ( <i>precisión</i> ), MARCA DE TIEMPO ( <i>precisión</i> ) SIN ZONA HORARIA	Una fecha y hora del día del calendario con una precisión específica. TIMESTAMP(3) es equivalente a TIMESTAMP .

DDL	DML	Descripción
No disponible	MARCA DE TIEMPO CON ZONA HORARIA	Fecha y hora del día del calendario en una zona horaria. Las zonas horarias se pueden especificar como desfases con respecto a UTC, como nombres de zonas horarias de la IANA, o mediante UTC, UT, Z o GMT.
No disponible	MARCA DE TIEMPO( <i>precisi</i> ) CON ZONA HORARIA	Una fecha y hora del día del calendario con una precisión específica, en una zona horaria.
No disponible	INTERVALO AÑO A MES	Un intervalo de uno o más meses enteros
No disponible	INTERVALO DÍA A SEGUNDO	Un intervalo de uno o más segundos, minutos, horas o días
ARRAY< <i>element_</i> <i>ype</i> >	ARRAY[ <i>element_</i> <i>ype</i> ]	Una matriz de valores. Todos los valores deben ser del mismo tipo.
MAP< <i>key_type</i> , <i>value_type</i> >	MAPA( <i>key_type</i> , <i>value_type</i> )	Un mapa en el que los valores se pueden buscar por clave. Todas las claves deben tener el mismo valor y todos los valores deben tener el mismo valor.
STRUCT< <i>field_</i> <i>e_1</i> <i>e_1</i> : <i>field_typ</i> <i>e_1</i> , <i>field_nam</i> <i>e_2</i> : <i>field_typ</i> <i>e_2</i> , ...>	FILA( <i>field_nam</i> <i>e_1</i> <i>field_typ</i> <i>e_1</i> , <i>field_nam</i> <i>e_2</i> <i>field_typ</i> <i>e_2</i> , ...)	Estructura de datos con campos con nombre y sus valores.
No disponible	JSON	Tipo de valor JSON, que puede ser un objeto JSON, una matriz JSON, un número JSON, una cadena JSON, true, false o null.

DDL	DML	Descripción
No disponible	UUID	Un UUID (identificador único universal).
No disponible	IPADDRESS	Una dirección IPv4 o IPv6.
No disponible	<a href="#">HyperLogLog</a> <a href="#">P4HyperLogLog</a> <a href="#">SetDigest</a> <a href="#">QDigest</a> <a href="#">TDigest</a>	Estos tipos de datos admiten funciones internas aproximadas. Para obtener más información acerca de cada tipo, visite el enlace a la entrada correspondiente en la documentación de Trino.

## Ejemplos de tipos de datos

En la siguiente tabla se muestran ejemplos de literales de los tipos de datos DML.

Tipo de datos	Ejemplos
BOOLEAN	true false
TINYINT	TINYINT '123'
SMALLINT	SMALLINT '123'
INT, INTEGER	123456790
BIGINT	BIGINT '1234567890' 2147483648
REAL	'123456.78'
DOBLE	1.234

Tipo de datos	Ejemplos
DECIMAL ( <i>precisión</i> , <i>escala</i> )	DECIMAL '123.456'
CHAR, CHAR ( <i>longitud</i> )	CHAR 'hello world', CHAR 'hello ''world''!'
VARCHAR, VARCHAR ( <i>longitud</i> )	VARCHAR 'hello world', VARCHAR 'hello ''world''!'
VARBINARY	X'00 01 02'
HORA, HORA ( <i>precisión</i> )	TIME '10:11:12' , TIME '10:11:12.345'
HORA CON ZONA HORARIA	TIME '10:11:12.345 -06:00'
FECHA	DATE '2024-03-25'
MARCA DE TIEMPO, MARCA DE TIEMPO SIN ZONA HORARIA, MARCA DE TIEMPO ( <i>precisión</i> ), MARCA DE TIEMPO ( <i>precisión</i> ) SIN ZONA HORARIA	TIMESTAMP '2024-03-25 11:12:13' , TIMESTAMP '2024-03-25 11:12:13.456'
MARCA DE TIEMPO CON ZONA HORARIA, MARCA DE TIEMPO ( <i>precisión</i> ) CON ZONA HORARIA	TIMESTAMP '2024-03-25 11:12:13.456 Europe/Berlin'
INTERVALO AÑO A MES	INTERVAL '3' MONTH
INTERVALO DÍA A SEGUNDO	INTERVAL '2' DAY
ARRAY[ <i>element_type</i> ]	ARRAY['one', 'two', 'three']

Tipo de datos	Ejemplos
MAPA( <i>key_type</i> , <i>value_type</i> )	MAP(ARRAY['one', 'two', 'three'], ARRAY[1, 2, 3])  Tenga en cuenta que los mapas se crean a partir de una matriz de claves y una matriz de valores.
FILA( <i>field_name_1</i> <i>field_type_1</i> , <i>field_name_2</i> <i>field_type_2</i> , ...)	ROW('one', 'two', 'three')  Tenga en cuenta que las filas creadas de esta manera no tienen nombres de columna. Puede usar CAST para nombrar las columnas, como en el siguiente ejemplo:  <pre>CAST(ROW(1, 2, 3) AS ROW(one INT, two INT, three INT))</pre>
JSON	JSON '{"one":1, "two": 2, "three": 3}'
UUID	UUID '12345678-90ab-cdef-1234-567890abcdef'
IPADDRESS	IPADDRESS '10.0.0.1'  IPADDRESS '2001:db8::1'

## Consideraciones sobre los tipos de datos

### Límites de tamaño

En el caso de los tipos de datos que no especifican un límite de tamaño, tenga en cuenta que existe un límite práctico de 32 MB para todos los datos de una sola fila. Para obtener más información, consulte [Row or column size limitation](#) en [Consideraciones y limitaciones de las consultas SQL en Amazon Athena](#).

### CHAR y VARCHAR

Un valor CHAR(*n*) siempre tiene un recuento de *n* caracteres. Por ejemplo, si cambia “abc” a CHAR(7), se añaden 4 espacios finales.

Las comparaciones de valores CHAR incluyen espacios iniciales y finales.

Si se especifica una longitud para CHAR o VARCHAR, las cadenas se truncan con la longitud especificada al leerlas. Si la cadena de datos subyacente es más larga, esta permanece sin cambios.

Para evitar una comilla simple en un CHAR o VARCHAR, utilice una comilla simple adicional.

Para convertir un tipo de datos que no sea una cadena en una consulta DML, cámbielo al tipo de datos VARCHAR.

Para utilizar la función `substr` para devolver una subcadena de longitud especificada desde un tipo de datos CHAR, primero debe convertir el valor de CHAR a VARCHAR. En el siguiente ejemplo, `col1` utiliza el tipo de datos CHAR.

```
substr(CAST(col1 AS VARCHAR), 1, 4)
```

## DECIMAL

Para especificar valores decimales como literales en consultas SELECT como, por ejemplo, al seleccionar filas con un valor decimal específico, puede especificar el tipo DECIMAL y enumerar el valor decimal como literal entre comillas simples en su consulta, como en este ejemplo:

```
SELECT * FROM my_table  
WHERE decimal_value = DECIMAL '0.12'
```

```
SELECT DECIMAL '44.6' + DECIMAL '77.2'
```

## Trabajo con datos de marca de tiempo

En esta sección se describen algunas consideraciones para trabajar con datos de marcas de tiempo en Athena.

### Note

El tratamiento de las marcas de tiempo ha cambiado un poco entre la versión 2 y 3 del motor de Athena. Para obtener información sobre los errores relacionados con la marca de tiempo que se pueden producir en la versión 3 del motor de Athena y las soluciones sugeridas, consulte [Cambios en la marca de tiempo](#) en la referencia de [Versión 3 del motor Athena](#).

## Formato para escribir datos de marcas de tiempo en objetos de Amazon S3

El formato en el que se deben escribir los datos de marcas de tiempo en los objetos de Amazon S3 depende tanto del tipo de datos de la columna como de la [biblioteca SerDe](#) que utilice.

- Si cuenta con una columna de tabla de tipo DATE, Athena espera que la columna o propiedad correspondiente de los datos sea una cadena en el formato ISO YYYY-MM-DD o un tipo de fecha integrado, como los de Parquet u ORC.
- Si cuenta con una columna de tabla de tipo TIME, Athena espera que la columna o propiedad correspondiente de los datos sea una cadena en el formato ISO HH:MM:SS o un tipo de hora integrado, como los de Parquet u ORC.
- Si cuenta con una columna de tabla de tipo TIMESTAMP, Athena espera que la columna o propiedad correspondiente de los datos sea una cadena en el formato YYYY-MM-DD HH:MM:SS.SSS (tenga en cuenta el espacio entre la fecha y la hora) o un tipo de hora integrado, como los de Parquet, ORC o Ion.

### Note

Las marcas de tiempo de OpenCSVSerDe son una excepción y se deben codificar como periodos UNIX con una resolución de milisegundos.

## Cómo garantizar que los datos particionados en el tiempo coincidan con el campo de marca de tiempo de un registro

El productor de los datos debe asegurarse de que los valores de la partición se alineen con los datos de la partición. Por ejemplo, si sus datos tienen una propiedad `timestamp` y utiliza Firehose para cargar los datos en Amazon S3, debe utilizar la [partición dinámica](#), ya que la partición predeterminada de Firehose se basa en un tiempo cronológico.

## Uso de una cadena como tipo de datos para las claves de partición

Por motivos de rendimiento, es preferible utilizar `STRING` como el tipo de datos para las claves de partición. Si bien Athena reconoce los valores de partición en el formato `YYYY-MM-DD` como fechas al utilizar el tipo `DATE`, esto puede provocar un rendimiento deficiente. Por este motivo, en su lugar le recomendamos que utilice el tipo de datos `STRING` para las claves de partición.

## Cómo escribir consultas para campos de marca de tiempo que también se encuentren particionados por tiempo

La forma en que se escriben las consultas para los campos de marca de tiempo que se encuentran particionados por tiempo dependen del tipo de tabla que se desee consultar.

### Tablas de Hive

Con las tablas de Hive más utilizadas en Athena, el motor de consultas no conoce las relaciones entre las columnas y las claves de partición. Por este motivo, siempre debe agregar predicados en las consultas tanto para la columna como para la clave de partición.

Por ejemplo, supongamos que tiene una columna `event_time` y una clave de partición `event_date` y desea consultar eventos entre las 23:00 y las 03:00. En este caso, debe incluir predicados en la consulta tanto para la columna como para la clave de partición, como en el siguiente ejemplo.

```
WHERE event_time BETWEEN start_time AND end_time
      AND event_date BETWEEN start_time_date AND end_time_date
```

### Tablas de Iceberg

Con las tablas de Iceberg, puede utilizar valores de partición calculados, lo que simplifica las consultas. Por ejemplo, supongamos que la tabla de Iceberg se creó con una cláusula `PARTITIONED BY` como la siguiente:

```
PARTITIONED BY (event_date month(event_time))
```

En este caso, el motor de consultas reduce de forma automática las particiones en función de los valores de los predicados `event_time`. Por este motivo, la consulta solo necesita especificar un predicado para `event_time`, como en el siguiente ejemplo.

```
WHERE event_time BETWEEN start_time AND end_time
```

Para obtener más información, consulte [Creación de tablas de Iceberg](#).

## Consultas, funciones y operadores de DML

El motor de consultas DML de Athena generalmente admite la sintaxis de Trino y Presto y agrega sus propias mejoras. Athena no es compatible con todas las características de Presto. Para obtener



más información, consulte los temas sobre instrucciones específicas de esta sección, así como [Consideraciones y limitaciones](#). Para obtener información sobre las funciones, consulte [Funciones en Amazon Athena](#). Para obtener más información acerca de las versiones de motor Athena, consulte [Control de versiones del motor Athena](#).

Para obtener información sobre las instrucciones DDL, consulte [Instrucciones DDL](#). Para ver una lista de las instrucciones DDL no admitidas, consulte [DDL no admitido](#).

## SELECT

Recupera filas de datos de cero o más tablas.

### Note

En este tema se proporciona un resumen de información de referencia. La información completa sobre el uso de SELECT y el lenguaje SQL está fuera del alcance de esta documentación. Para obtener información sobre el uso de SQL específico de Athena, consulte [Consideraciones y limitaciones de las consultas SQL en Amazon Athena](#) y [Ejecución de consultas SQL mediante Amazon Athena](#). Para ver un ejemplo de cómo crear una base de datos, crear una tabla y ejecutar una consulta SELECT en una tabla en Athena, consulte [Introducción](#).

## Sinopsis

```
[ WITH with_query [, ...] ]  
SELECT [ ALL | DISTINCT ] select_expression [, ...]  
[ FROM from_item [, ...] ]  
[ WHERE condition ]  
[ GROUP BY [ ALL | DISTINCT ] grouping_element [, ...] ]  
[ HAVING condition ]  
[ { UNION | INTERSECT | EXCEPT } [ ALL | DISTINCT ] select ]  
[ ORDER BY expression [ ASC | DESC ] [ NULLS FIRST | NULLS LAST] [, ...] ]  
[ OFFSET count [ ROW | ROWS ] ]  
[ LIMIT [ count | ALL ] ]
```

**Note**

Las palabras reservadas en las instrucciones SELECT de SQL deben ponerse entre comillas dobles. Para obtener más información, consulte [Lista de palabras reservadas en instrucciones SQL SELECT](#).

## Parámetros

[ WITH with\_query [, ...] ]

Puede utilizar WITH para aplanar consultas anidadas o para simplificar subconsultas.

Se admite el uso de la cláusula WITH para crear consultas recursivas con la versión 3 del motor de Athena. La profundidad máxima de recursión es 10.

La cláusula WITH precede a la lista SELECT de una consulta y define una o varias subconsultas para utilizarlas en la consulta SELECT.

Cada subconsulta define una tabla temporal, similar a la definición de una vista, a la que puede hacer referencia en la cláusula FROM. Las tablas se utilizan solo cuando se ejecuta la consulta.

La sintaxis de with\_query es:

```
subquery_table_name [ ( column_name [, ...] ) ] AS (subquery)
```

Donde:

- subquery\_table\_name es un nombre único para una tabla temporal que define los resultados de la subconsulta de cláusula WITH. Cada subquery debe tener un nombre de tabla al que se pueda hacer referencia en la cláusula FROM.
- column\_name [, ...] es una lista opcional de los nombres de columna de salida. La cantidad de nombres de columnas debe ser igual o menor que la cantidad de columnas definidas por la subquery.
- subquery es cualquier declaración de consulta.

[ ALL | DISTINCT ] select\_expression

select\_expression determina las filas que deben seleccionarse. Con la opción select\_expression se puede usar uno de los siguientes formatos:

```
expression [ [ AS ] column_alias ] [, ...]
```

```
row_expression.* [ AS ( column_alias [, ...] ) ]
```

```
relation.*
```

```
*
```

- La sintaxis `expression [ [ AS ] column_alias ]` especifica una columna de salida. La sintaxis opcional `[AS] column_alias` especifica un nombre de encabezado personalizado que se utilizará en la columna de la salida.
- Para `row_expression.* [ AS ( column_alias [, ...] ) ]`, `row_expression` es una expresión arbitraria de tipo ROW. Los campos de la fila definen las columnas de salida que se incluirán en el resultado.
- Para `relation.*`, las columnas de `relation` se incluyen en el resultado. Esta sintaxis no permite el uso de alias de columna.
- El asterisco `*` especifica que todas las columnas deben incluirse en el conjunto de resultados.
- En el conjunto de resultados, el orden de las columnas es el mismo que el orden de su especificación según la expresión seleccionada. Si una expresión seleccionada devuelve varias columnas, el orden de las columnas sigue el orden utilizado en la relación de origen o la expresión de tipo fila.
- Cuando se especifican alias de columna, estos sustituyen a los nombres de columna o campos de fila preexistentes. Si la expresión seleccionada no tiene nombres de columna, en la salida se muestran los nombres de columna anónimos indexados a cero (`_col0`, `_col1` y `_col2`, ...).
- ALL es el valor predeterminado. El uso de ALL se trata de la misma manera que si se hubiera omitido; todas las filas de todas las columnas se seleccionan y se conservan los duplicados.
- Utilice DISTINCT para devolver solo valores únicos cuando una columna contiene valores duplicados.

```
FROM from_item [, ...]
```

Indica la entrada de datos de la consulta, donde `from_item` puede ser una vista, una construcción JOIN o una subconsulta, tal y como se describe a continuación.

`from_item` puede ser:

- `table_name [ [ AS ] alias [ (column_alias [, ...]) ] ]`

Donde `table_name` es el nombre de la tabla de destino en la que se seleccionan las filas, `alias` es el nombre que se debe dar a la salida de la instrucción `SELECT` y `column_alias` define las columnas para el alias especificado.

- O BIEN -

- `join_type from_item [ ON join_condition | USING ( join_column [, ...] ) ]`

Donde `join_type` es uno de los valores siguientes:

- `[ INNER ] JOIN`
- `LEFT [ OUTER ] JOIN`
- `RIGHT [ OUTER ] JOIN`
- `FULL [ OUTER ] JOIN`
- `CROSS JOIN`
- `ON join_condition | USING (join_column [, ...])` Donde con `join_condition` puede especificar nombres de columnas para claves de combinación en varias tablas y el uso de `join_column` requiere que `join_column` exista en ambas tablas.

[ WHERE condition ]

Filtra los resultados de acuerdo con la `condition` que especifique, donde `condition` por lo general tiene la siguiente sintaxis.

```
column_name operator value [[[AND | OR] column_name operator value] ...]
```

El *operador* puede ser uno de los comparadores `=`, `>`, `<`, `>=`, `<=`, `<>`, `!=`.

Las siguientes expresiones de subconsulta también se pueden utilizar en la cláusula `WHERE`.

- `[NOT] BETWEEN integer_A AND integer_B`: especifica un rango entre dos enteros, como en el ejemplo siguiente. Si el tipo de datos de la columna es `varchar`, primero se debe convertir la columna a números enteros.

```
SELECT DISTINCT processid FROM "webdata"."impressions"
WHERE cast(processid as int) BETWEEN 1500 and 1800
ORDER BY processid
```

- [NOT] LIKE *value*: busca el patrón especificado. Utilice el signo de porcentaje (%) como carácter comodín, como en el ejemplo siguiente.

```
SELECT * FROM "webdata"."impressions"  
WHERE referrer LIKE '%.org'
```

- [NOT] IN (*value*[, *value*[, ...]]): especifica una lista de valores posibles para una columna, como en el ejemplo siguiente.

```
SELECT * FROM "webdata"."impressions"  
WHERE referrer IN ('example.com', 'example.net', 'example.org')
```

[ GROUP BY [ ALL | DISTINCT ] grouping\_expressions [, ...] ]

Divide la salida de la instrucción SELECT en filas con valores coincidentes.

ALL y DISTINCT determinan si los conjuntos de agrupación duplicados producen cada uno filas de salida diferenciadas. Si se omite, el sistema presupone que se utiliza ALL.

grouping\_expressions le permite realizar operaciones de agrupación complejas. Puede utilizar operaciones de agrupación complejas para realizar análisis que requieran la agregación de varios conjuntos de columnas en una sola consulta.

El elemento grouping\_expressions puede ser cualquier función, como SUM, AVG o COUNT, realizado en columnas de entrada.

Las expresiones GROUP BY pueden agrupar la salida por nombres de columna de entrada que no se muestran en la salida de la instrucción SELECT.

Todas las expresiones de salida deben ser funciones o columnas agregadas que existan en la cláusula GROUP BY.

Puede utilizar una sola consulta para realizar análisis que requieran la agregación de varios conjuntos de columnas.

Athena admite agregaciones complejas usando GROUPING SETS, CUBE y ROLLUP. GROUP BY GROUPING SETS especifica varias listas de columnas en las que se agruparán. GROUP BY CUBE genera todos los conjuntos de agrupación posibles para un conjunto determinado de columnas. GROUP BY ROLLUP genera todos los subtotales posibles para un conjunto determinado de columnas. Las operaciones de agrupación complejas no admiten agrupación en expresiones compuestas por columnas de entrada. Solo se admiten nombres de columnas.

A menudo puede utilizar `UNION ALL` para obtener los mismos resultados que estas operaciones `GROUP BY`, pero las consultas que utilizan `GROUP BY` tienen la ventaja de leer los datos una vez, mientras que `UNION ALL` lee los datos subyacentes tres veces y puede producir resultados incoherentes cuando el origen de los datos está sujeto a cambios.

[ `HAVING condition` ]

Se usa con funciones agregadas y la cláusula `GROUP BY`. Controla qué grupos se seleccionan, con lo que se eliminan los grupos que no cumplen el valor de `condition`. Este filtro se aplica después de calcular los grupos y los agregados.

[ { `UNION | INTERSECT | EXCEPT` } [ `ALL | DISTINCT` ] `union_query` ] ]

`UNION`, `INTERSECT` y `EXCEPT` combinan los resultados de más de una instrucción `SELECT` en una sola consulta. `ALL` o `DISTINCT` determinan el carácter único de las filas que se incluyen en el conjunto de resultados final.

`UNION` combina las filas resultantes de la primera consulta con las filas resultantes de la segunda consulta. Para eliminar duplicados, `UNION` construye una tabla hash, que consume memoria. Para un mejor rendimiento, considere utilizar `UNION ALL` si la consulta no requiere la eliminación de duplicados. Si hay varias cláusulas `UNION`, dichas cláusulas se procesan de izquierda a derecha a menos que utilice paréntesis para definir explícitamente el orden de procesamiento.

`INTERSECT` devuelve solo las filas que están presentes en los resultados de la primera y la segunda consulta.

`EXCEPT` devuelve las filas de los resultados de la primera consulta, y excluye las filas encontradas por la segunda consulta.

`ALL` hace que se incluyan todas las filas, incluso si dichas filas son idénticas.

`DISTINCT` hace que solo se incluyan filas únicas en el conjunto de resultados combinados.

[ `ORDER BY expression` [ `ASC | DESC` ] [ `NULLS FIRST | NULLS LAST` ] [, ...] ]

Ordena un conjunto de resultados por una o varias salidas `expression`.

Cuando la cláusula contiene varias expresiones, el conjunto de resultados se ordena de acuerdo con la primera `expression`. A continuación, se aplica la segunda `expression` a las filas que tienen valores coincidentes de la primera expresión y así sucesivamente.

Cada `expression` puede especificar columnas de salida de `SELECT` o un número ordinal para una columna de salida por posición, a partir de uno.

ORDER BY se evalúa como el último paso después de cualquier cláusula GROUP BY o HAVING. ASC y DESC determinan si los resultados se ordenan de forma ascendente o descendente.

El orden de los valores nulos predeterminado es NULLS LAST, sea cual sea el orden de clasificación, ascendente o descendente.

[ OFFSET count [ ROW | ROWS ] ]

Utilice la cláusula OFFSET para descartar varias filas iniciales del conjunto de resultados. Si la cláusula ORDER BY está presente, la cláusula OFFSET se evalúa sobre un conjunto de resultados ordenado. El conjunto permanece ordenado después de descartar las filas omitidas. Si la consulta no tiene la cláusula ORDER BY, es arbitrario qué filas se descartan. Si el recuento especificado por OFFSET es igual o superior al tamaño del conjunto de resultados, el resultado final estará vacío.

LIMIT [ count | ALL ]

Restringe el número de filas del conjunto de resultados a count. LIMIT ALL equivale a omitir la cláusula LIMIT. Si la consulta no contiene ninguna cláusula ORDER BY, los resultados son arbitrarios.

TABLESAMPLE [ BERNOULLI | SYSTEM ] (porcentaje)

Operador opcional para seleccionar filas de una tabla basado en un método de muestreo.

BERNOULLI selecciona cada fila que se muestra en la tabla con una probabilidad de percentage. Todos los bloques físicos de la tabla se analizan y determinadas filas se omiten basándose en una comparación entre el valor de percentage de muestra y un valor aleatorio calculado en el tiempo de ejecución.

Con SYSTEM, la tabla se divide en segmentos lógicos de datos y la tabla se muestrea con esta granularidad.

O bien se seleccionan todas las filas de un segmento en particular, o el segmento se omite basándose en la comparación del valor de percentage de la muestra con un valor aleatorio calculado en tiempo de ejecución. El muestreo de SYSTEM depende del conector. Este método no garantiza que las probabilidades de muestreo sean independientes.

[ UNNEST (array\_or\_map) [WITH ORDINALITY] ]

Amplía una matriz o un mapa a una relación. Las matrices se amplían a una sola columna. Los mapas se amplían a dos columnas (clave, valor).

Puede utilizar UNNEST con varios argumentos, que se amplían en varias columnas con tantas filas como el mayor argumento de cardinalidad.

Las demás columnas se rellenan con valores nulos.

La cláusula WITH ORDINALITY añade una columna de ordinalidad al final.

UNNEST suele utilizarse con JOIN y puede hacer referencia a columnas de relaciones del lado izquierdo de JOIN.

### Obtención de las ubicaciones de archivos para los datos de origen en Simple Storage Service (Amazon S3)

Para ver la ubicación del archivo de Amazon S3 para los datos de una fila de tabla, puede utilizar "\$path" en una consulta SELECT, como en el ejemplo siguiente:

```
SELECT "$path" FROM "my_database"."my_table" WHERE year=2019;
```

Esta consulta devuelve un resultado similar al siguiente:

```
s3://DOC-EXAMPLE-BUCKET/datasets_mytable/year=2019/data_file1.json
```

Para devolver una lista ordenada y única de las rutas de nombre de archivo S3 para los datos de una tabla, puede utilizar SELECT DISTINCT y ORDER BY, como en el ejemplo a continuación.

```
SELECT DISTINCT "$path" AS data_source_file  
FROM sampledb.elb_logs  
ORDER By data_source_file ASC
```

Para devolver solo los nombres de archivo sin la ruta, puede pasar "\$path" como parámetro a una función regexp\_extract, como en el siguiente ejemplo.

```
SELECT DISTINCT regexp_extract("$path", '[^/]+$') AS data_source_file  
FROM sampledb.elb_logs  
ORDER By data_source_file ASC
```

Para devolver los datos de un archivo específico, especifique el archivo en la cláusula WHERE, como en el siguiente ejemplo.



```
SELECT *,"$path" FROM my_database.my_table WHERE "$path" = 's3://DOC-EXAMPLE-BUCKET/
my_table/my_partition/file-01.csv'
```

Para obtener más información y ejemplos, consulte el artículo del Centro de conocimientos [¿Cómo puedo ver el archivo de origen de Amazon S3 de una fila en una tabla de Athena?](#).

#### Note

En Athena, las columnas de metadatos ocultas de Hive o Iceberg \$bucket, \$file\_modified\_time, \$file\_size y \$partition no son compatibles con las vistas.

#### Incluir las comillas simples en caracteres de escape

Para incluir una comilla simple en caracteres de escape, inserte delante otra comilla simple, como en el ejemplo siguiente. Esto no debe confundirse con las comillas dobles.

```
Select '0'Reilly'
```

#### Resultados

```
0'Reilly
```

#### Recursos adicionales de

Para obtener más información acerca del uso de instrucciones SELECT en Athena, consulte los siguientes recursos.

Para obtener información sobre este tema	Consulte
Ejecución de consultas en Athena	<a href="#">Ejecución de consultas SQL mediante Amazon Athena</a>
Uso de SELECT para crear una tabla	<a href="#">Creación de una tabla a partir de los resultados de una consulta (CTAS)</a>
Insertar datos desde una consulta SELECT en otra tabla	<a href="#">INSERT INTO</a>

Para obtener información sobre este tema	Consulte
Uso de funciones integradas en instrucciones SELECT	<a href="#">Funciones en Amazon Athena</a>
Uso de funciones definidas por el usuario en instrucciones SELECT	<a href="#">Consulta con funciones definidas por el usuario</a>
Consulta de metadatos del catálogo de datos	<a href="#">Consulta de AWS Glue Data Catalog</a>

## INSERT INTO

Inserta filas nuevas en una tabla de destino en función de una instrucción de consulta SELECT que se ejecuta en una tabla de origen o en función de un conjunto de VALUES proporcionados como parte de la instrucción. Cuando la tabla de origen se basa en datos subyacentes con un formato concreto, como CSV o JSON, y la tabla de destino emplea otro formato, como Parquet u ORC, puede utilizar consultas INSERT INTO para transformar los datos seleccionados en el formato de la tabla de destino.

### Consideraciones y limitaciones

Cuando utilice las consultas INSERT con Athena, tenga en cuenta lo siguiente.

- De forma predeterminada, cuando se ejecuta una consulta INSERT en una tabla con datos subyacentes cifrados en Amazon S3, los archivos de salida que escribe la consulta INSERT no se cifran. Le recomendamos que cifre los resultados de las consultas INSERT si se van a insertar en tablas con datos cifrados.

Para obtener más información acerca cómo cifrar los resultados de las consultas utilizando la consola, consulte [Cifrado de los resultados de las consultas de Athena en Amazon S3](#).


Para habilitar el cifrado mediante la AWS CLI o la API de Athena, utilice las propiedades EncryptionConfiguration de la acción [StartQueryExecution](#) para especificar las opciones de cifrado de Amazon S3 en función de sus necesidades.

- Para las instrucciones INSERT INTO, la configuración del propietario del bucket esperado no se aplica a la ubicación de la tabla de destino en Amazon S3. La configuración del propietario esperado del bucket se aplica solo a la ubicación de salida de Amazon S3 especificada para los resultados de las consultas de Athena. Para obtener más información, consulte [Especificación de una ubicación de resultados de consulta mediante la consola de Athena](#).

- En relación con las instrucciones `INSERT INTO` que cumplen con los requisitos de ACID, consulte la sección `INSERT INTO` de [Actualización de los datos de las tablas de Iceberg](#).

## Formatos y SerDes compatibles

Puede ejecutar una consulta `INSERT` en las tablas creadas a partir de datos con los siguientes formatos y SerDes.

Formato de los datos	SerDe
Avro	<code>org.apache.hadoop.hive.serde2.avro.AvroSerDe</code>
Ion	<code>com.amazon.ionhiveserde.IonHiveSerDe</code>
JSON	<code>org.apache.hive.hcatalog.data.JsonSerDe</code>
ORC	<code>org.apache.hadoop.hive ql.io.orc.OrcSerde</code>
Parquet	<code>org.apache.hadoop.hive ql.io.parquet.serde.ParquetHiveSerDe</code>
Archivo de texto	<code>org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe</code> <div data-bbox="315 1192 1507 1413" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Se admiten los archivos CSV, TSV y los que tienen delimitadores personalizados.</p> </div>

No se admiten tablas en buckets.

`INSERT INTO` no es compatible con las tablas en buckets. Para obtener más información, consulte [Creación de particiones y asignación de buckets en Athena](#).

## Consultas federadas no compatibles

`INSERT INTO` no es compatible con las consultas federadas. Si intenta hacerlo, puede aparecer el mensaje de error `This operation is currently not supported for external catalogs` (Esta operación no

es compatible actualmente con catálogos externos). Para obtener información sobre las consultas federadas, consulte [Uso de consulta federada de Amazon Athena](#).

## Particiones

Tenga en cuenta los puntos de esta sección al utilizar particiones con consultas `INSERT INTO` o `CREATE TABLE AS SELECT`.

## Límites

La instrucción `INSERT INTO` admite escribir un máximo de 100 particiones en la tabla de destino. Si ejecuta la cláusula `SELECT` en una tabla con más de 100 particiones, se produce un error en la consulta a menos que la consulta `SELECT` esté limitada a 100 particiones o menos.

Para obtener más información sobre cómo evitar esta limitación, consulte [Uso de CTAS e INSERT INTO para evitar el límite de 100 particiones](#).

## Ordenamiento de columnas

Las instrucciones `INSERT INTO` o `CREATE TABLE AS SELECT` esperan que la columna particionada sea la última de la lista de columnas proyectadas en una instrucción `SELECT`.

Si la tabla de origen no está particionada o está particionada en columnas diferentes en comparación con la tabla de destino, las consultas como `INSERT INTO destination_table SELECT * FROM source_table` consideran que los valores de la última columna de la tabla de origen son valores para una columna de partición en la tabla de destino. Tenga esto en cuenta al intentar crear una tabla particionada desde una tabla no particionada.

## Recursos

Para obtener más información acerca del uso de `INSERT INTO` con la partición, consulte los siguientes recursos.

- Para insertar datos particionados en una tabla particionada, consulte [Uso de CTAS e INSERT INTO para evitar el límite de 100 particiones](#).
- Para insertar datos no particionados en una tabla particionada, consulte [Uso de CTAS e INSERT INTO en ETL y análisis de datos](#).

## Archivos escritos en Simple Storage Service (Amazon S3)

Tras ejecutar el comando INSERT, Athena escribe archivos en las ubicaciones de datos de origen de Amazon S3. Cada operación INSERT crea un nuevo archivo, en lugar de anexar el resultado a un archivo existente. Las ubicaciones de los archivos dependen de la estructura de la tabla y la consulta SELECT, si procede. Athena genera un archivo de manifiesto de datos para cada consulta INSERT. El manifiesto realiza un seguimiento de los archivos que escribió la consulta. Se guarda en la ubicación del resultado de la consulta de Athena en Amazon S3. Para obtener más información, consulte [Identificación de archivos de salida de consultas](#).

### Evite las actualizaciones altamente transaccionales

Cuando utiliza INSERT INTO para añadir filas a una tabla en Amazon S3, Athena no reescribe ni modifica los archivos existentes. En su lugar, escribe las filas como uno o más archivos nuevos. Dado que las tablas con [muchos archivos pequeños reducen el rendimiento de las consultas](#) y que las operaciones de escritura lectura, como PutObject y GetObject, generan costes más altos de Amazon S3, tenga en cuenta las siguientes opciones cuando utiliza INSERT INTO:

- Ejecute las operaciones INSERT INTO con menos frecuencia en lotes más grandes de filas.
- Para grandes volúmenes de ingesta de datos, considere la posibilidad de utilizar un servicio como [Amazon Data Firehose](#).
- Evite por completo el uso de INSERT INTO. En su lugar, acumule filas en archivos más grandes y cárguelos directamente en Amazon S3, donde Athena podrá consultarlos.

### Localización de archivos huérfanos

Si una instrucción CTAS o INSERT INTO falla, los datos huérfanos se pueden dejar en la ubicación de los datos y se pueden leer en consultas posteriores. Para localizar archivos huérfanos para su inspección o eliminación, puede utilizar el archivo de manifiesto de datos que Athena proporciona para realizar un seguimiento de la lista de archivos que se van a escribir. Para obtener más información, consulte [Identificación de archivos de salida de consultas](#) y [DataManifestLocation](#).

### INSERT INTO...SELECT

Especifica la consulta que se va a ejecutar en una tabla, `source_table`, lo que determina las filas que se van a insertar en una segunda tabla, `destination_table`. Si la consulta SELECT especifica columnas de `source_table`, las columnas deben coincidir exactamente con las de `destination_table`.

Para obtener más información acerca de las consultas SELECT, consulte [SELECT](#).

## Sinopsis

```
INSERT INTO destination_table
SELECT select_query
FROM source_table_or_view
```

## Ejemplos

Seleccione todas las filas de la tabla `vancouver_pageviews` e insértelas en la tabla `canada_pageviews`:

```
INSERT INTO canada_pageviews
SELECT *
FROM vancouver_pageviews;
```

Seleccione solo las filas de la tabla `vancouver_pageviews` en las que la columna `date` tiene un valor comprendido entre `2019-07-01` y `2019-07-31`. Después, insértelas en `canada_july_pageviews`:

```
INSERT INTO canada_july_pageviews
SELECT *
FROM vancouver_pageviews
WHERE date
      BETWEEN date '2019-07-01'
             AND '2019-07-31';
```

En la tabla `cities_world`, seleccione los valores de las columnas `city` y `state` de aquellas filas que tengan el valor `usa` en la columna `country` e insértelos en las columnas `city` y `state` de la tabla `cities_usa`:

```
INSERT INTO cities_usa (city,state)
SELECT city,state
FROM cities_world
      WHERE country='usa'
```

## INSERT INTO...VALUES

Inserta filas en una tabla existente especificando columnas y valores. Las columnas especificadas y los tipos de datos asociados deben coincidir exactamente con las columnas y los tipos de datos de la tabla de destino.

### Important

No es recomendable utilizar VALUES para insertar las filas, ya que Athena genera archivos con cada operación INSERT. Esto puede provocar que se creen muchos archivos pequeños y reducir el rendimiento de las consultas de la tabla. Para identificar los archivos que crea una consulta INSERT, examine el archivo del manifiesto de datos. Para obtener más información, consulte [Trabajo con resultados de las consultas, consultas recientes y archivos de salida](#).

## Sinopsis

```
INSERT INTO destination_table [(col1,col2,...)]
VALUES (col1value,col2value,...)[,
      (col1value,col2value,...)][,
      ...]
```

## Ejemplos

En los siguientes ejemplos, la tabla de ciudades tiene varias columnas: `id`, `city`, `state` y `state_motto`. La columna `id` es de tipo `INT`, mientras que todas las demás columnas son de tipo `VARCHAR`.

Inserte una sola fila en la tabla `cities` con todos los valores de columna especificados:

```
INSERT INTO cities
VALUES (1,'Lansing','MI','Si quaeris peninsulam amoenam circumspice')
```

Inserte dos filas en la tabla `cities`:

```
INSERT INTO cities
VALUES (1,'Lansing','MI','Si quaeris peninsulam amoenam circumspice'),
      (3,'Boise','ID','Esto perpetua')
```

## DELETE

Elimina filas de una tabla de Apache Iceberg. DELETE es transaccional y solo se admite en tablas de Apache Iceberg.

### Sinopsis

Para eliminar las filas de una tabla de Iceberg, use la siguiente sintaxis.

```
DELETE FROM [db_name.]table_name [WHERE predicate]
```

Para obtener más información y ejemplos, consulte la sección de DELETE de [Actualización de los datos de las tablas de Iceberg](#).

## UPDATE

Actualice las filas de una tabla de Apache Iceberg. UPDATE es transaccional y solo se admite en tablas de Apache Iceberg.

### Sinopsis

Para actualizar las filas de una tabla de Iceberg, use la siguiente sintaxis.

```
UPDATE [db_name.]table_name SET xx=yy[,...] [WHERE predicate]
```

Para obtener más información y ejemplos, consulte la sección de UPDATE de [Actualización de los datos de las tablas de Iceberg](#).

## MERGE INTO

Actualiza, elimina o inserta filas de forma condicional en una tabla de Apache Iceberg. Una sola instrucción puede combinar acciones de actualización, eliminación e inserción.

### Note

MERGE INTO es transaccional y solo es compatible con las tablas de Apache Iceberg en la versión 3 del motor Athena.



## Sinopsis

Para actualizar, eliminar o insertar filas de forma condicional de una tabla de Iceberg, use la siguiente sintaxis.

```
MERGE INTO target_table [ [ AS ] target_alias ]
USING { source_table | query } [ [ AS ] source_alias ]
ON search_condition
when_clause [...]
```

La cláusula *when\_clause* es una de las siguientes:

```
WHEN MATCHED [ AND condition ]
  THEN DELETE
```

```
WHEN MATCHED [ AND condition ]
  THEN UPDATE SET ( column = expression [, ...] )
```

```
WHEN NOT MATCHED [ AND condition ]
  THEN INSERT ( column_name [, column_name ...] ) VALUES ( expression, ... )
```

MERGE admite un número arbitrario de cláusulas WHEN con diferentes condiciones MATCHED. Las cláusulas de condición ejecutan la operación DELETE, UPDATE o INSERT de la primera cláusula WHEN seleccionada por el estado MATCHED y la condición de coincidencia.

Para cada fila de origen, las cláusulas WHEN se procesan en orden. Solo se ejecuta la primera cláusula WHEN coincidente. Las demás cláusulas se ignoran. Se produce un error de usuario cuando una sola fila de la tabla de destino coincide con más de una fila de origen.

Si una fila de origen no coincide con ninguna cláusula WHEN y no hay ninguna cláusula WHEN NOT MATCHED, se ignora la fila de origen.

En las cláusulas WHEN que tienen operaciones UPDATE, las expresiones de valores de la columna pueden hacer referencia a cualquier campo del destino o del origen. En el caso de NOT MATCHED, las expresiones INSERT pueden referirse a cualquier campo del origen.

## Ejemplo

En el siguiente ejemplo, se combinan filas de la segunda tabla en la primera tabla si las filas no existen en la primera tabla. Tenga en cuenta que las columnas que aparecen en la cláusula VALUES

deben ir precedidas del alias de la tabla de origen. Las columnas de destino enumeradas en la cláusula INSERT no deben tener ese prefijo.

```
MERGE INTO iceberg_table_sample as ice1
USING iceberg2_table_sample as ice2
ON ice1.col1 = ice2.col1
WHEN NOT MATCHED
THEN INSERT (col1)
VALUES (ice2.col1)
```

Para obtener más ejemplos de MERGE INTO, consulte [Actualización de los datos de las tablas de Iceberg](#).

## OPTIMIZE

Optimiza las filas de una tabla de Apache Iceberg al reescribir los archivos de datos en un diseño más optimizado en función de su tamaño y del número de archivos de eliminación asociados.

### Note

OPTIMIZE es transaccional y solo se admite en tablas de Apache Iceberg.

## Sintaxis

En el siguiente resumen de sintaxis, se muestra cómo optimizar el diseño de datos de una tabla de Iceberg.

```
OPTIMIZE [db_name.]table_name REWRITE DATA USING BIN_PACK
[WHERE predicate]
```

### Note

Solo se permiten columnas de partición en el *predicado* de la cláusula WHERE. Si se especifica una columna que no es de partición, se producirá un error en la consulta.

La acción de compactación se cobra por la cantidad de datos analizados durante el proceso de reescritura. La acción REWRITE DATA usa predicados para seleccionar los archivos que contienen filas que coinciden. Si alguna fila del archivo coincide con el predicado, el archivo se selecciona

para la optimización. Así, para controlar el número de archivos afectados por la operación de compactación, puede especificar una cláusula WHERE.

### Configuración de las propiedades de compactación

Para controlar el tamaño de los archivos que se seleccionarán para la compactación y el tamaño de archivo resultante después de la compactación, puede usar parámetros de propiedades de tabla. Puede utilizar el comando [ALTER TABLE SET PROPERTIES](#) para configurar las siguientes [propiedades de la tabla](#) relacionadas.

Recursos adicionales de

### [Optimización de las tablas de Iceberg](#)

## VACUUM

La instrucción VACUUM realiza el mantenimiento de las tablas de Apache Iceberg al eliminar los archivos de datos que ya no se necesitan.

#### Note

VACUUM es transaccional y solo es compatible con las tablas de Apache Iceberg en la versión 3 del motor Athena.

Se recomienda ejecutar la instrucción VACUUM en las tablas de Iceberg para eliminar los archivos de datos que ya no sean relevantes y reducir el tamaño de los metadatos y el consumo de almacenamiento. Tenga en cuenta que, debido a que la instrucción VACUUM realiza llamadas de API a Amazon S3, se aplican cargos por las solicitudes asociadas a Amazon S3.

#### Warning

Si ejecuta una operación de vencimiento de instantáneas, ya no podrá viajar en el tiempo a instantáneas vencidas.

## Sinopsis

Para eliminar los archivos de datos que ya no se necesitan para una tabla de Iceberg, use la siguiente sintaxis.

```
VACUUM [database_name.]target_table
```

Para ejecutar VACUUM en una tabla cuyo nombre comience con un guión bajo (por ejemplo, `_mytable`), encierre el nombre de la tabla entre comillas inversas, como en el siguiente ejemplo. Si antepone al nombre de la tabla un nombre de base de datos, no encierre el nombre de la base de datos entre comillas inversas. Tenga en cuenta que las comillas dobles no funcionarán en lugar de las comillas inversas.

Este comportamiento es específico de VACUUM. Las instrucciones CREATE y INSERT INTO no requieren comillas inversas para los nombres de tabla que comienzan con guiones bajos.

```
VACUUM `_mytable`  
VACUUM my_database.`_mytable`
```

Tenga en cuenta también que VACUUM espera que los datos de Iceberg estén en una carpeta de Amazon S3 en lugar de en un bucket de Amazon S3. Por ejemplo, si los datos de Iceberg se encuentran en `s3://DOC-EXAMPLE-BUCKET/` en lugar de `s3://DOC-EXAMPLE-BUCKET/myicebergfolder/`, la instrucción VACUUM falla con el mensaje de error `GENERIC_INTERNAL_ERROR: Falta la ruta en la ubicación del sistema de archivos: s3://DOC-EXAMPLE-BUCKET`.

## Operaciones realizadas

VACUUM lleva a cabo las siguientes operaciones:

- Elimina las instantáneas que tengan una antigüedad superior a la cantidad de tiempo especificada en la propiedad de la tabla `vacuum_max_snapshot_age_seconds`. De forma predeterminada, esta propiedad se establece en 432 000 segundos (5 días).
- Elimina las instantáneas que no estén dentro del periodo de conservación y que superen el número especificado en la propiedad de la tabla `vacuum_min_snapshots_to_keep`. El valor predeterminado de es 1.

Puede especificar estas propiedades de la tabla en su instrucción CREATE TABLE. Después de crear la tabla, se puede usar la instrucción [ALTER TABLE SET PROPERTIES](#) para actualizarla.

- Elimina los metadatos y los archivos de datos a los que no se pueda acceder como resultado de la eliminación de la instantánea. Puede configurar la cantidad de archivos de metadatos antiguos que se van a retener al configurar la propiedad de la tabla `vacuum_max_metadata_files_to_keep`. El valor predeterminado es 100.

- Elimina los archivos huérfanos que tienen una antigüedad superior al tiempo especificado en la propiedad de la tabla `vacuum_max_snapshot_age_seconds`. Los archivos huérfanos son archivos del directorio de datos de la tabla que no forman parte del estado de la tabla.

Para obtener más información acerca de la creación y administración de tablas de Apache Iceberg en Athena, consulte [Creación de tablas de Iceberg](#) y [Administración de tablas de Iceberg](#).

## Uso de EXPLAIN y EXPLAIN ANALYZE en Athena

La instrucción EXPLAIN muestra el plan de ejecución lógico o distribuido de una instrucción SQL especificada, o valida la instrucción SQL. Puede generar los resultados en formato de texto o en formato de datos para representarlos en un gráfico.

### Note

Puede ver representaciones gráficas de planes lógicos y distribuidos para sus consultas en la consola de Athena sin usar la sintaxis EXPLAIN. Para obtener más información, consulte [Ver planes de ejecución para consultas SQL](#).

La instrucción EXPLAIN ANALYZE muestra el plan de ejecución distribuido de una instrucción SQL especificada y el costo computacional de cada operación en una consulta SQL. Puede generar los resultados en formato de texto o JSON.

## Consideraciones y limitaciones

Las instrucciones EXPLAIN y EXPLAIN ANALYZE de Athena tienen las siguientes limitaciones.

- Dado que las consultas EXPLAIN no escanean ningún dato, Athena no cobra por ellas. Sin embargo, como las consultas EXPLAIN realizan llamadas a AWS Glue para recuperar metadatos de la tabla, puede incurrir en cargos de Glue si las llamadas superan el [límite de nivel gratuito de Glue](#).
- Como se ejecutan las consultas EXPLAIN ANALYZE, analizan datos, de modo que Athena cobra en función de los datos escaneados.
- La información de filtrado de filas o celdas definida en Lake Formation y la información de estadísticas de consultas no se muestran en la salida de EXPLAIN y EXPLAIN ANALYZE.

## Sintaxis de EXPLAIN

```
EXPLAIN [ ( option [, ...] ) ] statement
```

La *opción* puede ser una de las siguientes:

```
FORMAT { TEXT | GRAPHVIZ | JSON }  
TYPE { LOGICAL | DISTRIBUTED | VALIDATE | IO }
```

Si no se especifica la opción FORMAT, la salida predeterminada es en formato TEXT. El tipo IO proporciona información acerca de las tablas y esquemas que lee la consulta. IO solo se admite en la versión 2 del motor Athena y solo se puede devolver en formato JSON.

## Sintaxis de EXPLAIN ANALYZE

Además de la salida incluida en EXPLAIN, la salida EXPLAIN ANALYZE también incluye estadísticas de tiempo de ejecución de la consulta especificada, como el uso de CPU, el número de filas de entrada y el número de filas de salida.

```
EXPLAIN ANALYZE [ ( option [, ...] ) ] statement
```

La *opción* puede ser una de las siguientes:

```
FORMAT { TEXT | JSON }
```

Si no se especifica la opción FORMAT, la salida predeterminada es en formato TEXT. Como todas las consultas para EXPLAIN ANALYZE son DISTRIBUTED, la opción TYPE no está disponible para EXPLAIN ANALYZE.

La *instrucción* puede ser una de las siguientes:

```
SELECT  
CREATE TABLE AS SELECT  
INSERT  
UNLOAD
```

## Ejemplos de EXPLAIN

Los siguientes ejemplos de EXPLAIN avanzan de los más sencillos a los más complejos.

## Ejemplo nro. 1 de EXPLAIN: uso de la instrucción EXPLAIN para mostrar un plan de consulta en formato de texto

En el siguiente ejemplo, EXPLAIN muestra el plan de ejecución de una consulta SELECT en registros de Elastic Load Balancing. El formato se establece de forma predeterminada en la salida de texto.

```
EXPLAIN
SELECT
  request_timestamp,
  elb_name,
  request_ip
FROM sampledb.elb_logs;
```

### Resultados

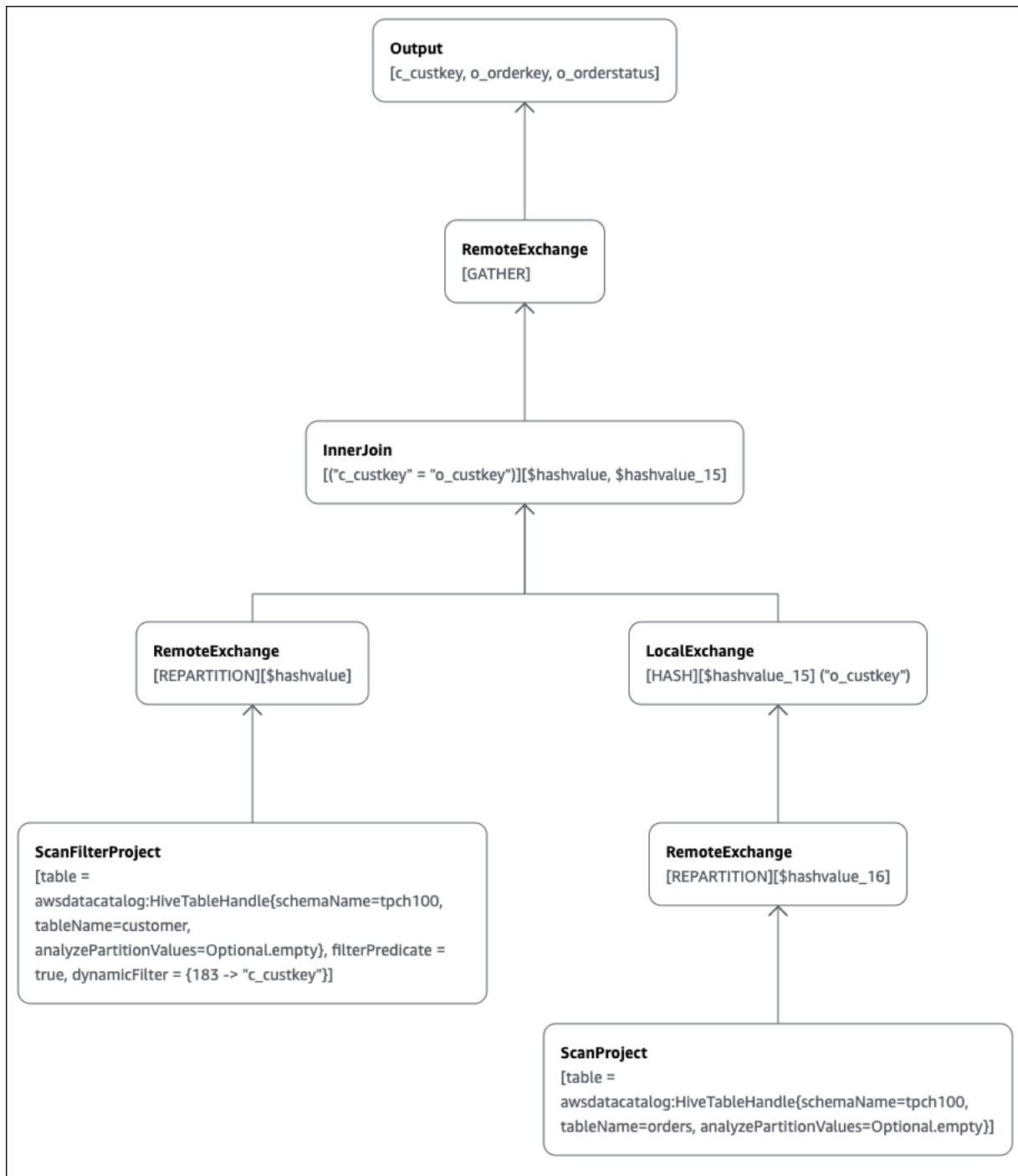
```
- Output[request_timestamp, elb_name, request_ip] => [[request_timestamp, elb_name, request_ip]]
  - RemoteExchange[GATHER] => [[request_timestamp, elb_name, request_ip]]
    - TableScan[awsdatacatalog:HiveTableHandle{schemaName=sampled,
      tableName=elb_logs,
      analyzePartitionValues=Optional.empty}] => [[request_timestamp, elb_name, request_ip]]
      LAYOUT: sampledb.elb_logs
      request_ip := request_ip:string:2:REGULAR
      request_timestamp := request_timestamp:string:0:REGULAR
      elb_name := elb_name:string:1:REGULAR
```

## Ejemplo nro. 2 de EXPLAIN: representación gráfica de un plan de consulta

Puede utilizar la consola de Athena para representar de manera gráfica un plan de consultas. Ingrese una instrucción SELECT como la siguiente en el editor de consultas de Athena y, a continuación, elija EXPLICAR.

```
SELECT
  c.c_custkey,
  o.o_orderkey,
  o.o_orderstatus
FROM tpch100.customer c
JOIN tpch100.orders o
  ON c.c_custkey = o.o_custkey
```

Se abrirá la página Explain (Explicar) del editor de consultas de Athena y se mostrará un plan distribuido y un plan lógico para la consulta. En el siguiente gráfico se muestra el plan lógico para el ejemplo.





**⚠ Important**

Actualmente, es posible que algunos filtros de partición no estén visibles en el gráfico de árbol de operadores anidado, aunque Athena los aplique a la consulta. Para comprobar el efecto de dichos filtros, ejecute EXPLAIN o EXPLAIN ANALYZE en la consulta y observe los resultados.

Para obtener más información sobre cómo usar las características de representación gráfica de planes de consulta en la consola de Athena, consulte [Ver planes de ejecución para consultas SQL](#).

Ejemplo nro. 3 de EXPLAIN: uso de la instrucción EXPLAIN para verificar la poda de particiones

Cuando se utiliza un predicado de filtrado en una clave particionada para consultar una tabla particionada, el motor de consultas aplica el predicado a la clave particionada para reducir la cantidad de datos leídos.

En el siguiente ejemplo se utiliza una consulta EXPLAIN para verificar la poda de partición para una consulta SELECT en una tabla particionada. Primero, una instrucción CREATE TABLE crea la tabla tpch100.orders\_partitioned. La tabla está particionada en la columna o\_orderdate.

```
CREATE TABLE `tpch100.orders_partitioned` (  
  `o_orderkey` int,  
  `o_custkey` int,  
  `o_orderstatus` string,  
  `o_totalprice` double,  
  `o_orderpriority` string,  
  `o_clerk` string,  
  `o_shippriority` int,  
  `o_comment` string)  
PARTITIONED BY (  
  `o_orderdate` string)  
ROW FORMAT SERDE  
  'org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe'  
STORED AS INPUTFORMAT  
  'org.apache.hadoop.hive.q1.io.parquet.MapredParquetInputFormat'  
OUTPUTFORMAT  
  'org.apache.hadoop.hive.q1.io.parquet.MapredParquetOutputFormat'  
LOCATION  
  's3://DOC-EXAMPLE-BUCKET/<your_directory_path>/'
```

La tabla `tpch100.orders_partitioned` tiene varias particiones en `o_orderdate`, como se muestra en el comando `SHOW PARTITIONS`.

```
SHOW PARTITIONS tpch100.orders_partitioned;

o_orderdate=1994
o_orderdate=2015
o_orderdate=1998
o_orderdate=1995
o_orderdate=1993
o_orderdate=1997
o_orderdate=1992
o_orderdate=1996
```

La siguiente consulta `EXPLAIN` comprueba la poda de partición en la instrucción `SELECT` especificada.

```
EXPLAIN
SELECT
  o_orderkey,
  o_custkey,
  o_orderdate
FROM tpch100.orders_partitioned
WHERE o_orderdate = '1995'
```

## Resultados

```
Query Plan
- Output[o_orderkey, o_custkey, o_orderdate] => [[o_orderkey, o_custkey, o_orderdate]]
  - RemoteExchange[GATHER] => [[o_orderkey, o_custkey, o_orderdate]]
    - TableScan[awsdatacatalog:HiveTableHandle{schemaName=tpch100,
      tableName=orders_partitioned,
      analyzePartitionValues=Optional.empty}] => [[o_orderkey, o_custkey, o_orderdate]]
      LAYOUT: tpch100.orders_partitioned
      o_orderdate := o_orderdate:string:-1:PARTITION_KEY
      :: [[1995]]
      o_custkey := o_custkey:int:1:REGULAR
      o_orderkey := o_orderkey:int:0:REGULAR
```

El texto en negrita del resultado muestra que se aplicó el predicado `o_orderdate = '1995'` en `PARTITION_KEY`.

## Ejemplo nro. 4 de EXPLAIN: uso de una consulta EXPLAIN para verificar el orden de unión y el tipo de unión

La siguiente consulta EXPLAIN verifica el orden y el tipo de unión de la instrucción SELECT. Utilice una consulta como esta para examinar el uso de la memoria de consulta de modo que pueda reducir las posibilidades de obtener un error EXCEEDED\_LOCAL\_MEMORY\_LIMIT.

```
EXPLAIN (TYPE DISTRIBUTED)
SELECT
  c.c_custkey,
  o.o_orderkey,
  o.o_orderstatus
FROM tpch100.customer c
JOIN tpch100.orders o
  ON c.c_custkey = o.o_custkey
WHERE c.c_custkey = 123
```

## Resultados

```
Query Plan
Fragment 0 [SINGLE]
  Output layout: [c_custkey, o_orderkey, o_orderstatus]
  Output partitioning: SINGLE []
  Stage Execution Strategy: UNGROUPED_EXECUTION
  - Output[c_custkey, o_orderkey, o_orderstatus] => [[c_custkey, o_orderkey,
o_orderstatus]]
    - RemoteSource[1] => [[c_custkey, o_orderstatus, o_orderkey]]

Fragment 1 [SOURCE]
  Output layout: [c_custkey, o_orderstatus, o_orderkey]
  Output partitioning: SINGLE []
  Stage Execution Strategy: UNGROUPED_EXECUTION
  - CrossJoin => [[c_custkey, o_orderstatus, o_orderkey]]
    Distribution: REPLICATED
    - ScanFilter[table = awsdatacatalog:HiveTableHandle{schemaName=tpch100,
tableName=customer, analyzePartitionValues=Optional.empty}, grouped = false,
filterPredicate = ("c_custkey" = 123)] => [[c_custkey]]
      LAYOUT: tpch100.customer
      c_custkey := c_custkey:int:0:REGULAR
    - LocalExchange[SINGLE] () => [[o_orderstatus, o_orderkey]]
      - RemoteSource[2] => [[o_orderstatus, o_orderkey]]
```

```

Fragment 2 [SOURCE]
  Output layout: [o_orderstatus, o_orderkey]
  Output partitioning: BROADCAST []
  Stage Execution Strategy: UNGROUPED_EXECUTION
  - ScanFilterProject[table = awsdatacatalog:HiveTableHandle{schemaName=tpch100,
tableName=orders, analyzePartitionValues=Optional.empty}, grouped = false,
filterPredicate = ("o_custkey" = 123)] => [[o_orderstatus, o_orderkey]]
    LAYOUT: tpch100.orders
    o_orderstatus := o_orderstatus:string:2:REGULAR
    o_custkey := o_custkey:int:1:REGULAR
    o_orderkey := o_orderkey:int:0:REGULAR

```

La consulta de ejemplo se optimizó en una combinación cruzada para obtener un mejor rendimiento. Los resultados muestran que `tpch100.orders` se distribuirá como el tipo de distribución `BROADCAST`. Esto implica que la tabla `tpch100.orders` se distribuirá a todos los nodos que realizan la operación de unión. El tipo de distribución `BROADCAST` requerirá que todos los resultados filtrados de la tabla `tpch100.orders` se ajusten en la memoria de cada nodo que realiza la operación de unión.

Sin embargo, la tabla `tpch100.customer` es más pequeña que `tpch100.orders`. Dado que `tpch100.customer` requiere menos memoria, puede reescribir la consulta en `BROADCAST tpch100.customer` en lugar de `tpch100.orders`. Esto reduce la posibilidad de que la consulta reciba el error `EXCEEDED_LOCAL_MEMORY_LIMIT`. Esta estrategia supone los siguientes puntos:

- La `tpch100.customer.c_custkey` es única en la tabla `tpch100.customer`.
- Existe una relación de asignación de uno a muchos entre `tpch100.customer` y `tpch100.orders`.

En el siguiente ejemplo, se muestra la consulta reescrita.

```

SELECT
  c.c_custkey,
  o.o_orderkey,
  o.o_orderstatus
FROM tpch100.orders o
JOIN tpch100.customer c -- the filtered results of tpch100.customer are distributed to
  all nodes.
  ON c.c_custkey = o.o_custkey
WHERE c.c_custkey = 123

```

## Ejemplo nro. 5 de EXPLAIN: uso de una consulta EXPLAIN para eliminar predicados que no tengan efecto

Puede utilizar una consulta EXPLAIN para verificar la eficacia de los predicados de filtrado. Puede utilizar los resultados para eliminar predicados que no tengan ningún efecto, como en el ejemplo a continuación.

```
EXPLAIN
  SELECT
    c.c_name
  FROM tpch100.customer c
  WHERE c.c_custkey = CAST(RANDOM() * 1000 AS INT)
  AND c.c_custkey BETWEEN 1000 AND 2000
  AND c.c_custkey = 1500
```

### Resultados

```
Query Plan
- Output[c_name] => [[c_name]]
  - RemoteExchange[GATHER] => [[c_name]]
    - ScanFilterProject[table =
awsdatacatalog:HiveTableHandle{schemaName=tpch100,
tableName=customer, analyzePartitionValues=Optional.empty},
filterPredicate = (("c_custkey" = 1500) AND ("c_custkey" =
CAST(("random"() * 1E3) AS int)))] => [[c_name]]
      LAYOUT: tpch100.customer
      c_custkey := c_custkey:int:0:REGULAR
      c_name := c_name:string:1:REGULAR
```

El `filterPredicate` en los resultados muestra que el optimizador fusionó los tres predicados originales en dos predicados y cambió su orden de aplicación.

```
filterPredicate = (("c_custkey" = 1500) AND ("c_custkey" = CAST(("random"() * 1E3) AS
int)))
```

Como los resultados muestran que el predicado `AND c.c_custkey BETWEEN 1000 AND 2000` no tiene ningún efecto, puede eliminar este predicado sin cambiar los resultados de la consulta.

Para obtener información sobre los términos utilizados en los resultados de las consultas EXPLAIN, consulte [Descripción de los resultados de la instrucción EXPLAIN de Athena](#).

## Ejemplos de EXPLAIN ANALYZE

En los siguientes ejemplos, se muestran consultas y salidas EXPLAIN ANALYZE.

Ejemplo nro. 1 de EXPLAIN ANALYZE: uso de EXPLAIN ANALYZE para mostrar un plan de consulta y los costos computacionales en formato de texto

En el siguiente ejemplo, EXPLAIN ANALYZE muestra el plan de ejecución y los costos computacionales de una consulta SELECT en los registros de CloudFront. El formato se establece de forma predeterminada en la salida de texto.

```
EXPLAIN ANALYZE SELECT FROM cloudfront_logs LIMIT 10
```

### Resultados

```
Fragment 1
  CPU: 24.60ms, Input: 10 rows (1.48kB); per task: std.dev.: 0.00, Output: 10 rows
(1.48kB)
  Output layout: [date, time, location, bytes, requestip, method, host, uri, status,
referrer,\
  os, browser, browserversion]
Limit[10] => [[date, time, location, bytes, requestip, method, host, uri, status,
referrer, os,\
  browser, browserversion]]
  CPU: 1.00ms (0.03%), Output: 10 rows (1.48kB)
  Input avg.: 10.00 rows, Input std.dev.: 0.00%
LocalExchange[SINGLE] () => [[date, time, location, bytes, requestip, method, host,
uri, status, referrer, os,\
  browser, browserversion]]
  CPU: 0.00ns (0.00%), Output: 10 rows (1.48kB)
  Input avg.: 0.63 rows, Input std.dev.: 387.30%
RemoteSource[2] => [[date, time, location, bytes, requestip, method, host, uri, status,
referrer, os,\
  browser, browserversion]]
  CPU: 1.00ms (0.03%), Output: 10 rows (1.48kB)
  Input avg.: 0.63 rows, Input std.dev.: 387.30%

Fragment 2
  CPU: 3.83s, Input: 998 rows (147.21kB); per task: std.dev.: 0.00, Output: 20 rows
(2.95kB)
  Output layout: [date, time, location, bytes, requestip, method, host, uri, status,
referrer, os,\
  browser, browserversion]
```

```

LimitPartial[10] => [[date, time, location, bytes, requestip, method, host, uri,
status, referrer, os,\
  browser, browserversion]]
      CPU: 5.00ms (0.13%), Output: 20 rows (2.95kB)
      Input avg.: 166.33 rows, Input std.dev.: 141.42%
TableScan[awsdatacatalog:HiveTableHandle{schemaName=default, tableName=cloudfront_logs,
\
  analyzePartitionValues=Optional.empty},
grouped = false] => [[date, time, location, bytes, requestip, method, host, uri, st
      CPU: 3.82s (99.82%), Output: 998 rows (147.21kB)
      Input avg.: 166.33 rows, Input std.dev.: 141.42%
      LAYOUT: default.cloudfront_logs
      date := date:date:0:REGULAR
      referrer := referrer:string:9:REGULAR
      os := os:string:10:REGULAR
      method := method:string:5:REGULAR
      bytes := bytes:int:3:REGULAR
      browser := browser:string:11:REGULAR
      host := host:string:6:REGULAR
      requestip := requestip:string:4:REGULAR
      location := location:string:2:REGULAR
      time := time:string:1:REGULAR
      uri := uri:string:7:REGULAR
      browserversion := browserversion:string:12:REGULAR
      status := status:int:8:REGULAR

```

Ejemplo nro. 2 de EXPLAIN ANALYZE: uso de EXPLAIN ANALYZE para mostrar un plan de consulta en formato JSON

En el siguiente ejemplo, se muestra el plan de ejecución y los costos computacionales de una consulta SELECT en los registros de CloudFront. En el ejemplo se especifica JSON como formato de salida.

```
EXPLAIN ANALYZE (FORMAT JSON) SELECT * FROM cloudfront_logs LIMIT 10
```

## Resultados

```
{
  "fragments": [{
    "id": "1",

    "stageStats": {
```

```

    "totalCpuTime": "3.31ms",
    "inputRows": "10 rows",
    "inputDataSize": "1514B",
    "stdDevInputRows": "0.00",
    "outputRows": "10 rows",
    "outputDataSize": "1514B"
  },
  "outputLayout": "date, time, location, bytes, requestip, method, host,\
    uri, status, referrer, os, browser, browserversion",

  "logicalPlan": {
    "1": [{
      "name": "Limit",
      "identifier": "[10]",
      "outputs": ["date", "time", "location", "bytes", "requestip", "method",
"host",\
        "uri", "status", "referrer", "os", "browser", "browserversion"],
      "details": "",
      "distributedNodeStats": {
        "nodeCpuTime": "0.00ns",
        "nodeOutputRows": 10,
        "nodeOutputDataSize": "1514B",
        "operatorInputRowsStats": [{
          "nodeInputRows": 10.0,
          "nodeInputRowsStdDev": 0.0
        }]
      }
    },
    "children": [{
      "name": "LocalExchange",
      "identifier": "[SINGLE] ()",
      "outputs": ["date", "time", "location", "bytes", "requestip",
"method", "host",\
        "uri", "status", "referrer", "os", "browser", "browserversion"],
      "details": "",
      "distributedNodeStats": {
        "nodeCpuTime": "0.00ns",
        "nodeOutputRows": 10,
        "nodeOutputDataSize": "1514B",
        "operatorInputRowsStats": [{
          "nodeInputRows": 0.625,
          "nodeInputRowsStdDev": 387.2983346207417
        }]
      }
    },
    "children": [{

```



```

        "name": "RemoteSource",
        "identifier": "[2]",
        "outputs": ["date", "time", "location", "bytes", "requestip",
"method", "host",\
        "uri", "status", "referrer", "os", "browser",
"browserversion"],
        "details": "",
        "distributedNodeStats": {
            "nodeCpuTime": "0.00ns",
            "nodeOutputRows": 10,
            "nodeOutputDataSize": "1514B",
            "operatorInputRowsStats": [{
                "nodeInputRows": 0.625,
                "nodeInputRowsStdDev": 387.2983346207417
            }]
        },
        "children": []
    ]
}
}, {
    "id": "2",
    "stageStats": {
        "totalCpuTime": "1.62s",
        "inputRows": "500 rows",
        "inputDataSize": "75564B",
        "stdDevInputRows": "0.00",
        "outputRows": "10 rows",
        "outputDataSize": "1514B"
    },
    "outputLayout": "date, time, location, bytes, requestip, method, host, uri,
status,\
referrer, os, browser, browserversion",
    "logicalPlan": {
        "1": [{
            "name": "LimitPartial",
            "identifier": "[10]",
            "outputs": ["date", "time", "location", "bytes", "requestip", "method",
"host", "uri",\
            "status", "referrer", "os", "browser", "browserversion"],
            "details": "",

```

```

        "distributedNodeStats": {
            "nodeCpuTime": "0.00ns",
            "nodeOutputRows": 10,
            "nodeOutputDataSize": "1514B",
            "operatorInputRowsStats": [{
                "nodeInputRows": 83.33333333333333,
                "nodeInputRowsStdDev": 223.60679774997897
            }]
        },
        "children": [{
            "name": "TableScan",
            "identifier": "[awsdatacatalog:HiveTableHandle{schemaName=default,\
analyzePartitionValues=Optional.empty},\
grouped = false]",
            "outputs": ["date", "time", "location", "bytes", "requestip",
"method", "host", "uri",\
"status", "referrer", "os", "browser", "browserversion"],
            "details": "LAYOUT: default.cloudfront_logs\ndate :=
date:date:0:REGULAR\nreferrer :=\
referrer: string:9:REGULAR\nos := os:string:10:REGULAR
\nmethod := method:string:5:\
REGULAR\nbytes := bytes:int:3:REGULAR\nbrowser :=
browser:string:11:REGULAR\nhost :=\
host:string:6:REGULAR\nrequestip := requestip:string:4:REGULAR
\nlocation :=\
location:string:2:REGULAR\ntime := time:string:1: REGULAR
\nuri := uri:string:7:\
REGULAR\nbrowserversion := browserversion:string:12:REGULAR
\nstatus :=\
status:int:8:REGULAR\n",
            "distributedNodeStats": {
                "nodeCpuTime": "1.62s",
                "nodeOutputRows": 500,
                "nodeOutputDataSize": "75564B",
                "operatorInputRowsStats": [{
                    "nodeInputRows": 83.33333333333333,
                    "nodeInputRowsStdDev": 223.60679774997897
                }]
            },
            "children": []
        }]
    ]
}

```

```
}]
}
```

## Recursos adicionales de

Para obtener información adicional, consulte los siguientes recursos.

- [Descripción de los resultados de la instrucción EXPLAIN de Athena](#)
- [Ver planes de ejecución para consultas SQL](#)
- [Visualización de estadísticas y detalles de ejecución de consultas completadas](#)
- Documentación de Trino sobre [EXPLAIN](#)
- Documentación de Trino sobre [EXPLAIN ANALYZE](#)
- [Optimice el rendimiento de las consultas federadas con EXPLAIN y EXPLAIN ANALYZE en Amazon Athena](#) en el Blog de macrodatos de AWS.

## Descripción de los resultados de la instrucción EXPLAIN de Athena

Este tema proporciona una breve guía de los términos operativos utilizados en los resultados de instrucciones EXPLAIN en Athena.

## Tipos de salida de instrucciones EXPLAIN

Las salidas de instrucciones EXPLAIN pueden ser de dos tipos:

- Plan lógico: muestra el plan lógico que utiliza el motor SQL para ejecutar una instrucción. La sintaxis para esta opción es EXPLAIN o EXPLAIN (TYPE LOGICAL).
- Plan distribuido: muestra un plan de ejecución en un entorno distribuido. La salida muestra fragmentos, que son etapas de procesamiento. Cada fragmento de plan es procesado por uno o más nodos. Los datos se pueden intercambiar entre los nodos que procesan los fragmentos. La sintaxis para esta opción es EXPLAIN (TYPE DISTRIBUTED).

En la salida de un plan distribuido, los fragmentos (etapas de procesamiento) están indicados por un Fragment *número* [*fragment\_type*], donde el *número* es un entero basado en cero y *fragment\_type* especifica cómo los nodos ejecutan el fragmento. En la tabla siguiente se describen los tipos de fragmentos, que proporcionan información del diseño del intercambio de datos.

## Tipos de fragmentos del plan distribuido

Tipo de fragmento	Descripción
SINGLE	El fragmento se ejecuta en un solo nodo.
HASH	El fragmento se ejecuta en un número de nodos fijo. Los datos de entrada se distribuyen utilizando una función hash.
ROUND_ROB IN	El fragmento se ejecuta en un número de nodos fijo. Los datos de entrada se distribuyen de forma rotativa.
BROADCAST	El fragmento se ejecuta en un número de nodos fijo. Los datos de entrada se transmiten a todos los nodos.
SOURCE	El fragmento se ejecuta en nodos donde se accede a las divisiones de entrada.

## Exchange

Los términos relacionados con el intercambio describen cómo se intercambian datos entre nodos de trabajo. Las transferencias pueden ser locales o remotas.

### LocalExchange [*exchange\_type*]

Transfiere datos de manera local dentro de los nodos de trabajo para diferentes etapas de una consulta. El valor de *exchange\_type* puede ser uno de los tipos de intercambio lógicos o distribuidos, como se describe más adelante en esta sección.

### RemoteExchange [*exchange\_type*]

Transfiere datos entre los nodos de trabajo para diferentes etapas de una consulta. El valor de *exchange\_type* puede ser uno de los tipos de intercambio lógicos o distribuidos, como se describe más adelante en esta sección.

## Tipos de intercambio lógico

Los siguientes tipos de intercambio describen las acciones efectuadas durante la fase de intercambio de un plan lógico.

- **GATHER:** un nodo de trabajo único recopila la salida de todos los demás nodos de trabajo. Por ejemplo, la última etapa de una consulta de selección reúne los resultados de todos los nodos y los escribe en Amazon S3.
- **REPARTITION:** envía los datos de fila a un empleado específico según el esquema de partición necesario para aplicar al siguiente operador.
- **REPLICATE:** copia los datos de fila en todos los empleados.

## Tipos de intercambio distribuido

Los siguientes tipos de intercambio indican el diseño de los datos cuando se intercambian entre nodos en un plan distribuido.

- **HASH:** el intercambio distribuye datos a varios destinos utilizando una función hash.
- **SINGLE:** el intercambio distribuye los datos a un único destino.

## Análisis

Los siguientes términos describen cómo se analizan los datos durante una consulta.

### TableScan

Analiza los datos de origen de una tabla desde Amazon S3 o un conector Apache Hive y aplica la poda de partición generada a partir del predicado del filtro.

### ScanFilter

Analiza los datos de origen de una tabla desde Amazon S3 o un conector Hive y aplica la poda de partición generada a partir del predicado del filtro y de predicados de filtro adicionales no aplicados a través de la poda de partición.

### ScanFilterProject

Primero, analiza los datos de origen de una tabla desde Amazon S3 o un conector Hive y aplica la poda de partición generada a partir del predicado del filtro y de predicados de filtro adicionales no aplicados a través de la poda de partición. A continuación, modifica el diseño de la memoria de los datos de salida en una nueva proyección para mejorar el rendimiento de etapas posteriores.

## Join

Une datos entre dos tablas. Las uniones se pueden clasificar por tipo de unión y por tipo de distribución.

### Tipos de combinación

Los tipos de unión definen la manera en que se produce la operación de unión.

**CrossJoin:** produce el producto cartesiano de las dos tablas unidas.

**InnerJoin:** selecciona registros que tienen valores coincidentes en ambas tablas.

**LeftJoin:** selecciona todos los registros de la tabla izquierda y los registros coincidentes de la tabla derecha. Si no se produce ninguna coincidencia, el resultado en el lado derecho es NULL.

**Right Join:** selecciona todos los registros de la tabla derecha y los registros coincidentes de la tabla izquierda. Si no se produce ninguna coincidencia, el resultado en el lado izquierdo es NULL.

**FullJoin:** selecciona todos los registros en los que hay una coincidencia en los registros de la tabla izquierda o derecha. La tabla combinada contiene todos los registros de las tablas y rellena NULL para las coincidencias que faltan en ambos lados.

### Note

Por razones de rendimiento, el motor de consultas puede reescribir una consulta de unión en un tipo de unión diferente para producir los mismos resultados. Por ejemplo, una consulta de unión interna con predicado en una tabla se puede volver a escribir en una `CrossJoin`. Esto empuja el predicado a la fase de escaneo de la tabla para que se escaneen menos datos.

### Tipos de distribución de unión

Los tipos de distribución definen la manera en que se intercambian los datos entre nodos de trabajo cuando se realiza la operación de unión.

**Particionada:** tanto la tabla izquierda como la derecha están particionadas en hash en todos los nodos de trabajo. La distribución particionada consume menos memoria en cada nodo. La distribución particionada puede ser mucho más lenta que las uniones replicadas. Las uniones particionadas son adecuadas cuando se unen dos tablas grandes.

Replicada: una tabla está particionada en hash en todos los nodos de trabajo y la otra tabla se replica en todos los nodos de trabajo para realizar la operación de unión. La distribución replicada puede ser mucho más rápida que las uniones particionadas, pero consume más memoria en cada nodo de trabajo. Si la tabla replicada es demasiado grande, el nodo de trabajo puede experimentar un error de memoria insuficiente. Las uniones replicadas son adecuadas cuando una de las tablas combinadas es pequeña.

## PREPARE

Creas una instrucción SQL con el nombre `statement_name` que se ejecutará en un momento posterior. La instrucción puede incluir parámetros representados por signos de interrogación. Para proporcionar valores para los parámetros y ejecutar la instrucción preparada, utilice [EXECUTE](#).

### Sinopsis

```
PREPARE statement_name FROM statement
```

En la tabla siguiente se describen estos parámetros.

Parámetro	Descripción
<code>statement_name</code>	Nombre de la instrucción a preparar. El nombre debe ser único dentro del grupo de trabajo.
<code>statement</code>	Una consulta SELECT, CTAS o INSERT INTO.

### Note

El número máximo de instrucciones preparadas en un grupo de trabajo es 1000.

### Ejemplos

El siguiente ejemplo prepara una consulta SELECT sin parámetros.

```
PREPARE my_select1 FROM  
SELECT * FROM nation
```

El siguiente ejemplo prepara una consulta SELECT que incluye parámetros. Los valores `productid` y `quantity` serán suministrados por la cláusula USING de una instrucción EXECUTE:

```
PREPARE my_select2 FROM
SELECT order FROM orders WHERE productid = ? and quantity < ?
```

En el siguiente ejemplo, se prepara una consulta de inserción.

```
PREPARE my_insert FROM
INSERT INTO cities_usa (city, state)
SELECT city, state
FROM cities_world
WHERE country = ?
```

Recursos adicionales de

[Consultas con instrucciones preparadas](#)

[EXECUTE](#)

[DEALLOCATE PREPARE](#)

[INSERT INTO](#)

EXECUTE

Ejecuta una instrucción preparada con el nombre `statement_name`. Los valores de los parámetros para los signos de interrogación de la instrucción preparada se definen en la cláusula USING en una lista separada por comas. Para crear una instrucción preparada, utilice [PREPARE](#).

Sinopsis

```
EXECUTE statement_name [ USING parameter1[, parameter2, ... ] ]
```

Ejemplos

El siguiente ejemplo prepara y ejecuta una consulta sin parámetros.

```
PREPARE my_select1 FROM
```



```
SELECT name FROM nation
EXECUTE my_select1
```

El siguiente ejemplo prepara y ejecuta una consulta con un único parámetro.

```
PREPARE my_select2 FROM
SELECT * FROM "my_database"."my_table" WHERE year = ?
EXECUTE my_select2 USING 2012
```

Equivale a:

```
SELECT * FROM "my_database"."my_table" WHERE year = 2012
```

El siguiente ejemplo prepara y ejecuta una consulta con dos parámetros.

```
PREPARE my_select3 FROM
SELECT order FROM orders WHERE productid = ? and quantity < ?
EXECUTE my_select3 USING 346078, 12
```

Recursos adicionales de

[Consultas con instrucciones preparadas](#)

[PREPARE](#)

[INSERT INTO](#)

[DEALLOCATE PREPARE](#)

Elimina la instrucción preparada con el nombre especificado de las instrucciones preparadas del grupo de trabajo actual.

Sinopsis

```
DEALLOCATE PREPARE statement_name
```

Ejemplos

En el siguiente ejemplo, se elimina la instrucción preparada `my_select1` del grupo de trabajo actual.

```
DEALLOCATE PREPARE my_select1
```

Recursos adicionales de

[Consultas con instrucciones preparadas](#)

[PREPARE](#)

**UNLOAD**

Escribe los resultados de la consulta desde una instrucción `SELECT` al formato de datos especificado. Los formatos admitidos para `UNLOAD` incluyen Apache Parquet, ORC, Apache Avro y JSON. CSV es el único formato de salida compatible con el comando `SELECT` de Athena; no obstante, pero puede usar el comando `UNLOAD`, que admite una variedad de formatos de salida, para incluir la consulta `SELECT` y reescribir su salida en uno de los formatos compatibles con `UNLOAD`.

Si bien puede utilizar la instrucción `CTAS` para generar datos en formatos distintos a CSV, esas instrucciones también requieren la creación de una tabla en Athena. La instrucción `UNLOAD` es útil cuando se desea generar los resultados de una consulta `SELECT` en un formato que no sea CSV, pero no se requiere la tabla asociada. Por ejemplo, una aplicación descendente puede requerir que los resultados de una consulta `SELECT` estén en formato JSON, y Parquet u ORC podrían proporcionar una ventaja de rendimiento sobre CSV si tiene la intención de utilizar los resultados de la consulta `SELECT` para un análisis adicional.

Consideraciones y limitaciones

Cuando utiliza la instrucción `UNLOAD` en Athena, tenga en cuenta los siguientes puntos:

- Sin orden global de archivos: los resultados de `UNLOAD` se escriben en varios archivos en paralelo. Si la consulta `SELECT` en la instrucción `UNLOAD` especifica un orden de clasificación, el contenido de cada archivo está en orden de clasificación, pero los archivos no están ordenados entre sí.
- Datos huérfanos no eliminados: en caso de error, Athena no intenta eliminar datos huérfanos. Este comportamiento es el mismo que en instrucciones `CTAS` y `INSERT INTO`.
- Particiones máximas: el número máximo de particiones que se pueden utilizar con `UNLOAD` es 100.
- Archivos de manifiesto y metadatos: Athena genera un archivo de metadatos y un archivo de manifiesto de datos para cada consulta `UNLOAD`. El manifiesto realiza un seguimiento de los archivos que escribió la consulta. Ambos archivos se guardan en la ubicación de resultados de consulta de Athena en Amazon S3. Para obtener más información, consulte [Identificación de archivos de salida de consultas](#).

- Cifrado: Los archivos de salida UNLOAD se cifran de acuerdo con la configuración de cifrado utilizada para Amazon S3. Para configurar la configuración de cifrado para cifrar el resultado UNLOAD, puede utilizar la [API EncryptionConfiguration](#).
- Instrucciones preparadas: UNLOAD se puede utilizar con instrucciones preparadas. Para obtener información acerca de las instrucciones preparadas en Athena, consulte [Uso de consultas parametrizadas](#).
- Cuotas de servicio: UNLOAD utiliza cuotas de consulta DML. Para obtener información sobre las cuotas, consulte [Service Quotas](#).
- Propietario esperado del bucket: la configuración del propietario esperado del bucket no se aplica a la ubicación de Amazon S3 de destino especificada en la consulta UNLOAD. La configuración del propietario esperado del bucket se aplica solo a la ubicación de salida de Amazon S3 especificada para los resultados de las consultas de Athena. Para obtener más información, consulte [Especificación de una ubicación de resultados de consulta mediante la consola de Athena](#).

## Sintaxis

La instrucción UNLOAD utiliza la siguiente sintaxis.

```
UNLOAD (SELECT col_name [, ...] FROM old_table)  
TO 's3://DOC-EXAMPLE-BUCKET/my_folder/'  
WITH ( property_name = 'expression' [, ...] )
```

Excepto cuando se escribe en particiones, el destino de T0 debe especificar una ubicación en Amazon S3 que no tenga datos. Antes de que la consulta UNLOAD escriba en la ubicación especificada, verifica que la ubicación del bucket está vacía. Como UNLOAD no escribe datos en la ubicación especificada si la ubicación ya tiene datos en ella, UNLOAD no sobrescribe los datos existentes. Para reutilizar una ubicación de bucket como destino para UNLOAD, elimine los datos de la ubicación del bucket y, a continuación, vuelva a ejecutar la consulta.

Tenga en cuenta que cuando UNLOAD escribe en particiones, este comportamiento es diferente. Si ejecuta la misma consulta UNLOAD varias veces con la misma instrucción SELECT, la misma ubicación T0 y las mismas particiones, cada consulta UNLOAD descargará los datos en Amazon S3 en la ubicación y las particiones especificadas.

## Parámetros


Los valores posibles para *property\_name* son los siguientes.

format = '**file\_format**'

Obligatorio. Especifica el formato de archivo de la salida. Los valores posibles para **file\_format** son ORC, PARQUET, AVRO, JSON o TEXTFILE.

compression = '**compression\_format**'

Opcional. Esta opción es específica de los formatos ORC y Parquet. Para ORC, el valor predeterminado es `zlib`, y para Parquet, el valor predeterminado es `gzip`. Para obtener información sobre los formatos de compresión compatibles, consulte [Compatibilidad con la compresión de Athena](#).

 Note

Esta opción no se aplica al formato AVRO. Athena utiliza `gzip` para los formatos JSON y TEXTFILE.

compression\_level = **compression\_level**

Opcional. El nivel de compresión que se debe utilizar para la compresión ZSTD. Esta propiedad solo se aplica a la compresión ZSTD. Para obtener más información, consulte [Uso de los niveles de compresión ZSTD en Athena](#).

field\_delimiter = '**delimiter**'


Opcional. Especifica un delimitador de campo de un solo carácter para archivos con formato CSV, TSV y otros formatos de texto. En el ejemplo siguiente se especifica un delimitador de coma.

```
WITH (field_delimiter = ',')
```

Actualmente, no se admiten delimitadores de campos de caracteres múltiples. Si no especifica un delimitador de campo, se utiliza el carácter octal `\001` (^A).

partitioned\_by = ARRAY[ **col\_name**[,...] ]

Opcional. Una matriz de columnas por las que se particiona la salida.

 Note

Asegúrese de que en su instrucción SELECT, los nombres de las columnas particionadas aparezcan en último lugar en la lista de columnas.

## Ejemplos

En el siguiente ejemplo se escribe el resultado de una consulta SELECT en la ubicación `s3://DOC-EXAMPLE-BUCKET/unload_test_1/` de Amazon S3 utilizando el formato JSON.

```
UNLOAD (SELECT * FROM old_table)
TO 's3://DOC-EXAMPLE-BUCKET/unload_test_1/'
WITH (format = 'JSON')
```

En el siguiente ejemplo se escribe el resultado de una consulta SELECT en formato Parquet mediante compresión Snappy.

```
UNLOAD (SELECT * FROM old_table)
TO 's3://DOC-EXAMPLE-BUCKET/'
WITH (format = 'PARQUET',compression = 'SNAPPY')
```

En el siguiente ejemplo se escriben cuatro columnas en formato de texto, con la salida particionada por la última columna.

```
UNLOAD (SELECT name1, address1, comment1, key1 FROM table1)
TO 's3://DOC-EXAMPLE-BUCKET/ partitioned/'
WITH (format = 'TEXTFILE', partitioned_by = ARRAY['key1'])
```

En el siguiente ejemplo se descargan los resultados de la consulta en la ubicación especificada mediante el formato de archivo Parquet, compresión ZSTD y el nivel 4 de compresión ZSTD.

```
UNLOAD (SELECT * FROM old_table)
TO 's3://DOC-EXAMPLE-BUCKET/'
WITH (format = 'PARQUET', compression = 'ZSTD', compression_level = 4)
```

## Recursos adicionales de

- [Simplificar las canalizaciones de ETL y ML con la característica UNLOAD de Amazon Athena](#) en el Blog de macrodatos de AWS.

## Funciones en Amazon Athena

Para obtener información acerca de los cambios en las funciones entre las versiones del motor de Athena, consulte [Referencia de la versión del motor Athena](#). Para obtener una lista de las zonas horarias que se pueden utilizar con el operador `AT TIME ZONE`, consulte [Zonas horarias admitidas](#).

## Versión 3 del motor Athena

Las funciones de la versión 3 del motor Athena se basan en Trino. Para obtener información sobre las funciones, los operadores y las expresiones de Trino, consulte [Funciones y operadores](#) y las siguientes subsecciones de la documentación de Trino.

- [Aggregate](#)
- [Array](#) (Matriz)
- [Binario](#)
- [Bitwise](#) (Bit a bit)
- [Color](#)
- [Comparación](#)
- [Condicional](#)
- [Conversion](#) (Conversión)
- [Fecha y hora](#)
- [Decimal](#) (Decimal)
- [Geospatial](#) (Geoespacial)
- [HyperLogLog](#)
- [IP Address](#) (Dirección IP)
- [JSON](#)
- [Lambda](#)
- [Logical](#) (Lógica)
- [Machine learning](#)
- [Map](#)
- [Math \(Matemática\)](#)
- [Quantile digest](#) (Resumen de cuantiles)
- [Expresión regular](#)
- [Session](#) (Sesión)
- [Set Digest](#) (Establecer resumen)
- [Cadena](#)
- [Tabla](#)
- [Teradata](#)

- [T-Digest](#)
- [URL](#)
- [UUID](#) (Identificador único universal)
- [Window](#)

## Versión 2 del motor Athena

Las funciones de la versión 2 del motor Athena se basan en [Presto 0.217](#). Para ver las funciones geoespaciales en la versión 2 del motor Athena, consulte [Funciones geoespaciales en la versión 2 del motor Athena](#).

### Note

La documentación específica de la versión sobre las funciones de Presto 0.217 ya no está disponible. Para obtener información sobre las funciones, los operadores y las expresiones actuales de Presto, consulte [Operadores y funciones de Presto](#) o consulte los enlaces de las subcategorías de esta sección.

- [Logical operators](#) (Operadores lógicos)
- [Comparison functions and operators](#) (Funciones y operadores de comparación)
- [Conditional expressions](#) (Expresiones condicionales)
- [Conversion functions](#) (Funciones de conversión)
- [Mathematical functions and operators](#) (Funciones y operadores matemáticos)
- [Bitwise functions](#) (Funciones Bitwise)
- [Decimal functions and operators](#) (Funciones y operadores decimales)
- [String functions and operators](#) (Funciones y operadores de cadena)
- [Binary functions](#) (Funciones binarias)
- [Date and time functions and operators](#) (Funciones y operadores de fecha y hora)
- [Regular expression functions](#) (Funciones de expresión regular)
- [JSON functions and operators](#) (Funciones y operadores JSON)
- [URL functions](#) (Funciones de URL)
- [Aggregate functions](#) (Funciones de agregación)
- [Window functions](#) (Funciones de ventana)

- [Color functions](#) (Funciones de color)
- [Array functions and operators](#) (Funciones y operadores de matriz)
- [Map functions and operators](#) (Funciones y operadores de mapas)
- [Lambda expressions and functions](#) (Expresiones y funciones de Lambda)
- [Teradata functions](#) (Funciones de teradatos)

## Zonas horarias admitidas

Puede utilizar el operador AT TIME ZONE en una instrucción SELECT timestamp para especificar la zona horaria de la marca de tiempo que se devuelve, como en el ejemplo siguiente:

```
SELECT timestamp '2012-10-31 01:00 UTC' AT TIME ZONE 'America/Los_Angeles' AS la_time;
```

## Resultados

**la\_time**

```
2012-10-30 18:00:00.000 America/Los_Angeles
```

La siguiente lista contiene las zonas horarias que se pueden utilizar con el operador AT TIME ZONE en Athena. Para obtener más funciones y ejemplos relacionados con zona horaria, consulte

[Funciones y ejemplos de zona horaria](#).

```
Africa/Abidjan  
Africa/Accra  
Africa/Addis_Ababa  
Africa/Algiers  
Africa/Asmara  
Africa/Asmera  
Africa/Bamako  
Africa/Bangui  
Africa/Banjul  
Africa/Bissau  
Africa/Blantyre  
Africa/Brazzaville  
Africa/Bujumbura  
Africa/Cairo  
Africa/Casablanca  
Africa/Ceuta
```



Africa/Conakry  
Africa/Dakar  
Africa/Dar\_es\_Salaam  
Africa/Djibouti  
Africa/Douala  
Africa/El\_Aaiun  
Africa/Freetown  
Africa/Gaborone  
Africa/Harare  
Africa/Johannesburg  
Africa/Juba  
Africa/Kampala  
Africa/Khartoum  
Africa/Kigali  
Africa/Kinshasa  
Africa/Lagos  
Africa/Libreville  
Africa/Lome  
Africa/Luanda  
Africa/Lubumbashi  
Africa/Lusaka  
Africa/Malabo  
Africa/Maputo  
Africa/Maseru  
Africa/Mbabane  
Africa/Mogadishu  
Africa/Monrovia  
Africa/Nairobi  
Africa/Ndjamena  
Africa/Niamey  
Africa/Nouakchott  
Africa/Ouagadougou  
Africa/Porto-Novo  
Africa/Sao\_Tome  
Africa/Timbuktu  
Africa/Tripoli  
Africa/Tunis  
Africa/Windhoek  
America/Adak  
America/Anchorage  
America/Anguilla  
America/Antigua  
America/Araguaina  
America/Argentina/Buenos\_Aires

America/Argentina/Catamarca  
America/Argentina/ComodRivadavia  
America/Argentina/Cordoba  
America/Argentina/Jujuy  
America/Argentina/La\_Rioja  
America/Argentina/Mendoza  
America/Argentina/Rio\_Gallegos  
America/Argentina/Salta  
America/Argentina/San\_Juan  
America/Argentina/San\_Luis  
America/Argentina/Tucuman  
America/Argentina/Ushuaia  
America/Aruba  
America/Asuncion  
America/Atikokan  
America/Atka  
America/Bahia  
America/Bahia\_Banderas  
America/Barbados  
America/Belem  
America/Belize  
America/Blanc-Sablon  
America/Boa\_Vista  
America/Bogota  
America/Boise  
America/Buenos\_Aires  
America/Cambridge\_Bay  
America/Campo\_Grande  
America/Cancun  
America/Caracas  
America/Catamarca  
America/Cayenne  
America/Cayman  
America/Chicago  
America/Chihuahua  
America/Coral\_Harbour  
America/Cordoba  
America/Costa\_Rica  
America/Creston  
America/Cuiaba  
America/Curacao  
America/Danmarkshavn  
America/Dawson  
America/Dawson\_Creek

America/Denver  
America/Detroit  
America/Dominica  
America/Edmonton  
America/Eirunepe  
America/El\_Salvador  
America/Ensenada  
America/Fort\_Nelson  
America/Fort\_Wayne  
America/Fortaleza  
America/Glace\_Bay  
America/Godthab  
America/Goose\_Bay  
America/Grand\_Turk  
America/Grenada  
America/Guadeloupe  
America/Guatemala  
America/Guayaquil  
America/Guyana  
America/Halifax  
America/Havana  
America/Hermosillo  
America/Indiana/Indianapolis  
America/Indiana/Knox  
America/Indiana/Marengo  
America/Indiana/Petersburg  
America/Indiana/Tell\_City  
America/Indiana/Vevay  
America/Indiana/Vincennes  
America/Indiana/Winamac  
America/Indianapolis  
America/Inuvik  
America/Iqaluit  
America/Jamaica  
America/Jujuy  
America/Juneau  
America/Kentucky/Louisville  
America/Kentucky/Monticello  
America/Knox\_IN  
America/Kralendijk  
America/La\_Paz  
America/Lima  
America/Los\_Angeles  
America/Louisville

America/Lower\_Princes  
America/Maceio  
America/Managua  
America/Manaus  
America/Marigot  
America/Martinique  
America/Matamoros  
America/Mazatlan  
America/Mendoza  
America/Menominee  
America/Merida  
America/Metlaktla  
America/Mexico\_City  
America/Miquelon  
America/Moncton  
America/Monterrey  
America/Montevideo  
America/Montreal  
America/Montserrat  
America/Nassau  
America/New\_York  
America/Nipigon  
America/Nome  
America/Noronha  
America/North\_Dakota/Beulah  
America/North\_Dakota/Center  
America/North\_Dakota/New\_Salem  
America/Ojinaga  
America/Panama  
America/Pangnirtung  
America/Paramaribo  
America/Phoenix  
America/Port-au-Prince  
America/Port\_of\_Spain  
America/Porto\_Acre  
America/Porto\_Velho  
America/Puerto\_Rico  
America/Punta\_Arenas  
America/Rainy\_River  
America/Rankin\_Inlet  
America/Recife  
America/Regina  
America/Resolute  
America/Rio\_Branco

America/Rosario  
America/Santa\_Isabel  
America/Santarem  
America/Santiago  
America/Santo\_Domingo  
America/Sao\_Paulo  
America/Scoresbysund  
America/Shiprock  
America/Sitka  
America/St\_Barthelemy  
America/St\_Johns  
America/St\_Kitts  
America/St\_Lucia  
America/St\_Thomas  
America/St\_Vincent  
America/Swift\_Current  
America/Tegucigalpa  
America/Thule  
America/Thunder\_Bay  
America/Tijuana  
America/Toronto  
America/Tortola  
America/Vancouver  
America/Virgin  
America/Whitehorse  
America/Winnipeg  
America/Yakutat  
America/Yellowknife  
Antarctica/Casey  
Antarctica/Davis  
Antarctica/DumontDUrville  
Antarctica/Macquarie  
Antarctica/Mawson  
Antarctica/McMurdo  
Antarctica/Palmer  
Antarctica/Rothera  
Antarctica/South\_Pole  
Antarctica/Syowa  
Antarctica/Troll  
Antarctica/Vostok  
Arctic/Longyearbyen  
Asia/Aden  
Asia/Almaty  
Asia/Amman

Asia/Anadyr  
Asia/Aqtau  
Asia/Aqtobe  
Asia/Ashgabat  
Asia/Ashkhabad  
Asia/Atyrau  
Asia/Baghdad  
Asia/Bahrain  
Asia/Baku  
Asia/Bangkok  
Asia/Barnaul  
Asia/Beirut  
Asia/Bishkek  
Asia/Brunei  
Asia/Calcutta  
Asia/Chita  
Asia/Choibalsan  
Asia/Chongqing  
Asia/Chungking  
Asia/Colombo  
Asia/Dacca  
Asia/Damascus  
Asia/Dhaka  
Asia/Dili  
Asia/Dubai  
Asia/Dushanbe  
Asia/Gaza  
Asia/Harbin  
Asia/Hebron  
Asia/Ho\_Chi\_Minh  
Asia/Hong\_Kong  
Asia/Hovd  
Asia/Irkutsk  
Asia/Istanbul  
Asia/Jakarta  
Asia/Jayapura  
Asia/Jerusalem  
Asia/Kabul  
Asia/Kamchatka  
Asia/Karachi  
Asia/Kashgar  
Asia/Kathmandu  
Asia/Katmandu  
Asia/Khandyga

Asia/Kolkata  
Asia/Krasnoyarsk  
Asia/Kuala\_Lumpur  
Asia/Kuching  
Asia/Kuwait  
Asia/Macao  
Asia/Macau  
Asia/Magadan  
Asia/Makassar  
Asia/Manila  
Asia/Muscat  
Asia/Nicosia  
Asia/Novokuznetsk  
Asia/Novosibirsk  
Asia/Omsk  
Asia/Oral  
Asia/Phnom\_Penh  
Asia/Pontianak  
Asia/Pyongyang  
Asia/Qatar  
Asia/Qyzylorda  
Asia/Rangoon  
Asia/Riyadh  
Asia/Saigon  
Asia/Sakhalin  
Asia/Samarkand  
Asia/Seoul  
Asia/Shanghai  
Asia/Singapore  
Asia/Srednekolymsk  
Asia/Taipei  
Asia/Tashkent  
Asia/Tbilisi  
Asia/Tehran  
Asia/Tel\_Aviv  
Asia/Thimbu  
Asia/Thimphu  
Asia/Tokyo  
Asia/Tomsk  
Asia/Ujung\_Pandang  
Asia/Ulaanbaatar  
Asia/Ulan\_Bator  
Asia/Urumqi  
Asia/Ust-Nera

Asia/Vientiane  
Asia/Vladivostok  
Asia/Yakutsk  
Asia/Yangon  
Asia/Yekaterinburg  
Asia/Yerevan  
Atlantic/Azores  
Atlantic/Bermuda  
Atlantic/Canary  
Atlantic/Cape\_Verde  
Atlantic/Faeroe  
Atlantic/Faroe  
Atlantic/Jan\_Mayen  
Atlantic/Madeira  
Atlantic/Reykjavik  
Atlantic/South\_Georgia  
Atlantic/St\_Helena  
Atlantic/Stanley  
Australia/ACT  
Australia/Adelaide  
Australia/Brisbane  
Australia/Broken\_Hill  
Australia/Canberra  
Australia/Currie  
Australia/Darwin  
Australia/Eucla  
Australia/Hobart  
Australia/LHI  
Australia/Lindeman  
Australia/Lord\_Howe  
Australia/Melbourne  
Australia/NSW  
Australia/North  
Australia/Perth  
Australia/Queensland  
Australia/South  
Australia/Sydney  
Australia/Tasmania  
Australia/Victoria  
Australia/West  
Australia/Yancowinna  
Brazil/Acre  
Brazil/DeNoronha  
Brazil/East



Brazil/West  
CET  
CST6CDT  
Canada/Atlantic  
Canada/Central  
Canada/Eastern  
Canada/Mountain  
Canada/Newfoundland  
Canada/Pacific  
Canada/Saskatchewan  
Canada/Yukon  
Chile/Continental  
Chile/EasterIsland  
Cuba  
EET  
EST5EDT  
Egypt  
Eire  
Europe/Amsterdam  
Europe/Andorra  
Europe/Astrakhan  
Europe/Athens  
Europe/Belfast  
Europe/Belgrade  
Europe/Berlin  
Europe/Bratislava  
Europe/Brussels  
Europe/Bucharest  
Europe/Budapest  
Europe/Busingen  
Europe/Chisinau  
Europe/Copenhagen  
Europe/Dublin  
Europe/Gibraltar  
Europe/Guernsey  
Europe/Helsinki  
Europe/Isle\_of\_Man  
Europe/Istanbul  
Europe/Jersey  
Europe/Kaliningrad  
Europe/Kiev  
Europe/Kirov  
Europe/Lisbon  
Europe/Ljubljana

Europe/London  
Europe/Luxembourg  
Europe/Madrid  
Europe/Malta  
Europe/Mariehamn  
Europe/Minsk  
Europe/Monaco  
Europe/Moscow  
Europe/Nicosia  
Europe/Oslo  
Europe/Paris  
Europe/Podgorica  
Europe/Prague  
Europe/Riga  
Europe/Rome  
Europe/Samara  
Europe/San\_Marino  
Europe/Sarajevo  
Europe/Simferopol  
Europe/Skopje  
Europe/Sofia  
Europe/Stockholm  
Europe/Tallinn  
Europe/Tirane  
Europe/Tiraspol  
Europe/Ulyanovsk  
Europe/Uzhgorod  
Europe/Vaduz  
Europe/Vatican  
Europe/Vienna  
Europe/Vilnius  
Europe/Volgograd  
Europe/Warsaw  
Europe/Zagreb  
Europe/Zaporozhye  
Europe/Zurich  
GB  
GB-Eire  
Hongkong  
Iceland  
Indian/Antananarivo  
Indian/Chagos  
Indian/Christmas  
Indian/Cocos

Indian/Comoro  
Indian/Kerguelen  
Indian/Mahe  
Indian/Maldives  
Indian/Mauritius  
Indian/Mayotte  
Indian/Reunion  
Iran  
Israel  
Jamaica  
Japan  
Kwajalein  
Libya  
MET  
MST7MDT  
Mexico/BajaNorte  
Mexico/BajaSur  
Mexico/General  
NZ  
NZ-CHAT  
Navajo  
PRC  
PST8PDT  
Pacific/Apia  
Pacific/Auckland  
Pacific/Bougainville  
Pacific/Chatham  
Pacific/Chuuk  
Pacific/Easter  
Pacific/Efate  
Pacific/Enderbury  
Pacific/Fakaofu  
Pacific/Fiji  
Pacific/Funafuti  
Pacific/Galapagos  
Pacific/Gambier  
Pacific/Guadalcanal  
Pacific/Guam  
Pacific/Honolulu  
Pacific/Johnston  
Pacific/Kiritimati  
Pacific/Kosrae  
Pacific/Kwajalein  
Pacific/Majuro

Pacific/Marquesas  
Pacific/Midway  
Pacific/Nauru  
Pacific/Niue  
Pacific/Norfolk  
Pacific/Noumea  
Pacific/Pago\_Pago  
Pacific/Palau  
Pacific/Pitcairn  
Pacific/Pohnpei  
Pacific/Ponape  
Pacific/Port\_Moresby  
Pacific/Rarotonga  
Pacific/Saipan  
Pacific/Samoa  
Pacific/Tahiti  
Pacific/Tarawa  
Pacific/Tongatapu  
Pacific/Truk  
Pacific/Wake  
Pacific/Wallis  
Pacific/Yap  
Poland  
Portugal  
ROK  
Singapore  
Turkey  
US/Alaska  
US/Aleutian  
US/Arizona  
US/Central  
US/East-Indiana  
US/Eastern  
US/Hawaii  
US/Indiana-Starke  
US/Michigan  
US/Mountain  
US/Pacific  
US/Pacific-New  
US/Samoa  
W-SU  
WET

## Funciones y ejemplos de zona horaria

A continuación, se muestran más funciones y ejemplos relacionados con zona horaria.

- `at_timezone` (***timestamp***, ***zone***): devuelve el valor de ***timestamp*** (marca temporal) en la hora local correspondiente para ***zone*** (zona).

### Ejemplo

```
SELECT at_timezone(timestamp '2021-08-22 00:00 UTC', 'Canada/Newfoundland')
```

### Resultado

```
2021-08-21 21:30:00.000 Canada/Newfoundland
```

- `timezone_hour` (***timestamp***): devuelve la hora de diferencia de la zona horaria con respecto al valor `timestamp` como `bigint`.

### Ejemplo

```
SELECT timezone_hour(timestamp '2021-08-22 04:00 UTC' AT TIME ZONE 'Canada/Newfoundland')
```

### Resultado

```
-2
```

- `timezone_minute` (***timestamp***): devuelve los minutos de diferencia de la zona horaria con respecto a ***timestamp*** como `bigint`.

### Ejemplo

```
SELECT timezone_minute(timestamp '2021-08-22 04:00 UTC' AT TIME ZONE 'Canada/Newfoundland')
```

### Resultado

```
-30
```

- `with_timezone` (***timestamp***, ***zone***): devuelve una marca temporal con zona horaria de los valores ***timestamp*** y ***zone*** especificados.

### Ejemplo

```
SELECT with_timezone(timestamp '2021-08-22 04:00', 'Canada/Newfoundland')
```

### Resultado

```
2021-08-22 04:00:00.000 Canada/Newfoundland
```

## Instrucciones DDL

Utilice las siguientes instrucciones DDL directamente en Athena.

El motor de consultas de Athena se basa en parte en el [DDL de HiveQL](#).

Athena no admite todas las instrucciones DDL y existen algunas diferencias entre el DDL de HiveQL y el DDL de Athena. Para obtener más información, consulte los temas de referencia de esta sección y [DDL no admitido](#).

### Temas

- [DDL no admitido](#)
- [ALTER DATABASE SET DBPROPERTIES](#)
- [ALTER TABLE ADD COLUMNS](#)
- [ALTER TABLE ADD PARTITION](#)
- [ALTER TABLE DROP PARTITION](#)
- [ALTER TABLE RENAME PARTITION](#)
- [ALTER TABLE REPLACE COLUMNS](#)
- [ALTER TABLE SET LOCATION](#)
- [ALTER TABLE SET TBLPROPERTIES](#)
- [CREATE DATABASE](#)
- [CREATE TABLE](#)
- [CREATE TABLE AS](#)

- [CREATE VIEW](#)
- [DESCRIBE](#)
- [DESCRIBE VIEW](#)
- [DROP DATABASE](#)
- [DROP TABLE](#)
- [DROP VIEW](#)
- [MSCK REPAIR TABLE](#)
- [SHOW COLUMNS](#)
- [SHOW CREATE TABLE](#)
- [SHOW CREATE VIEW](#)
- [SHOW DATABASES](#)
- [SHOW PARTITIONS](#)
- [SHOW TABLES](#)
- [SHOW TBLPROPERTIES](#)
- [SHOW VIEWS](#)

## DDL no admitido

Athena no admite las siguientes instrucciones DDL:

- ALTER INDEX
- ALTER TABLE *table\_name* ARCHIVE PARTITION
- ALTER TABLE *table\_name* CLUSTERED BY
- ALTER TABLE *table\_name* EXCHANGE PARTITION
- ALTER TABLE *table\_name* NOT CLUSTERED
- ALTER TABLE *table\_name* NOT SKEWED
- ALTER TABLE *table\_name* NOT SORTED
- ALTER TABLE *table\_name* NOT STORED AS DIRECTORIES
- ALTER TABLE *table\_name* partitionSpec CHANGE COLUMNS
- ALTER TABLE *table\_name* partitionSpec COMPACT

- ALTER TABLE *table\_name* partitionSpec CONCATENATE
- ALTER TABLE *table\_name* partitionSpec SET FILEFORMAT
- ALTER TABLE *table\_name* SET SERDEPROPERTIES
- ALTER TABLE *table\_name* SET SKEWED LOCATION
- ALTER TABLE *table\_name* SKEWED BY
- ALTER TABLE *table\_name* TOUCH
- ALTER TABLE *table\_name* UNARCHIVE PARTITION
- COMMIT
- CREATE INDEX
- CREAT ROL
- CREATE TABLE *table\_name* LIKE *existing\_table\_name*
- CREATE TEMPORARY MACRO
- DELETE FROM
- DESCRIBE DATABASE
- DFS
- DROP INDEX
- DROP ROLE
- DROP TEMPORARY MACRO
- EXPORT TABLE
- GRANT ROLE
- IMPORT TABLE
- LOCK DATABASE
- LOCK TABLE
- REVOKE ROLE
- ROLLBACK
- SHOW COMPACTIONS
- SHOW CURRENT ROLES
- SHOW GRANT
- SHOW INDEXES



- SHOW LOCKS
- SHOW PRINCIPALS
- SHOW ROLE GRANT
- SHOW ROLES
- MOSTRAR ESTADÍSTICAS
- SHOW TRANSACTIONS
- START TRANSACTION
- UNLOCK DATABASE
- UNLOCK TABLE

## ALTER DATABASE SET DBPROPERTIES

Crea una o varias propiedades de una base de datos. DATABASE y SCHEMA son intercambiables; significan lo mismo.

### Sinopsis

```
ALTER {DATABASE|SCHEMA} database_name
  SET DBPROPERTIES ('property_name'='property_value' [, ...] )
```

### Parámetros

```
SET DBPROPERTIES ('property_name'='property_value' [, ...]
```

Especifica una o varias propiedades de la base de datos con el nombre `property_name` y establece como valor de cada propiedad `property_value`. Si `property_name` ya existe, se sobrescribe el valor antiguo con `property_value`.

### Ejemplos

```
ALTER DATABASE jd_datasets
  SET DBPROPERTIES ('creator'='John Doe', 'department'='applied mathematics');
```

```
ALTER SCHEMA jd_datasets
  SET DBPROPERTIES ('creator'='Jane Doe');
```

## ALTER TABLE ADD COLUMNS

Agrega una o más columnas a una tabla existente. Cuando se utiliza la sintaxis `PARTITION` opcional, actualiza los metadatos de la partición.

### Sinopsis

```
ALTER TABLE table_name
  [PARTITION
    (partition_col1_name = partition_col1_value
    [,partition_col2_name = partition_col2_value][,...])]
  ADD COLUMNS (col_name data_type)
```

### Parámetros

`PARTITION (partition_col_name = partition_col_value [,...])`

Crea una partición con las combinaciones de nombre y valor de columna que especifica. Escriba `partition_col_value` entre comillas solo si el tipo de datos de la columna es una cadena.

`ADD COLUMNS (col_name data_type [,col_name data_type,...])`

Agrega columnas después de las columnas existentes pero antes de las columnas de partición.

### Ejemplos

```
ALTER TABLE events ADD COLUMNS (eventowner string)
```

```
ALTER TABLE events PARTITION (awsregion='us-west-2') ADD COLUMNS (event string)
```

```
ALTER TABLE events PARTITION (awsregion='us-west-2') ADD COLUMNS (eventdescription
string)
```

### Notas

- Para ver una nueva columna de tabla en el panel de navegación del Editor de consultas de Athena después de ejecutar `ALTER TABLE ADD COLUMNS`, actualice manualmente la lista de tablas en el editor y vuelva a expandirla.
- `ALTER TABLE ADD COLUMNS` no funciona para columnas con el tipo de datos `date`. Para solucionar este problema, utilice el tipo de datos `timestamp` en su lugar.

## ALTER TABLE ADD PARTITION

Crea una o varias columnas de partición para la tabla. Cada partición se compone de una o varias combinaciones diferenciadas de nombre y valor de columna. Se crea un directorio de datos independiente por cada combinación especificada, lo que puede mejorar el rendimiento de la consulta en algunas circunstancias. Las columnas con particiones no existen en la tabla de datos en sí, de modo que si utiliza un nombre de columna que tiene el mismo nombre que una columna de la tabla, obtendrá un error. Para obtener más información, consulte [Particiones de datos en Athena](#).

En Athena, una tabla y sus particiones deben utilizar los mismos formatos de datos, pero sus esquemas pueden ser diferentes. Para obtener más información, consulte [Actualizaciones en tablas con particiones](#).

Para obtener información sobre los permisos de nivel de recursos requeridos en las políticas de IAM (incluido `glue:CreatePartition`), consulte [Permisos de la API de AWS Glue: Referencia de recursos y acciones](#) y [Acceso detallado a las bases de datos y las tablas en AWS Glue Data Catalog](#). Para obtener información sobre la solución de problemas de permisos al utilizar Athena, consulte la sección [Permisos](#) del tema [Solución de problemas en Athena](#).

### Sinopsis

```
ALTER TABLE table_name ADD [IF NOT EXISTS]
PARTITION
(partition_col1_name = partition_col1_value
[,partition_col2_name = partition_col2_value]
[,...])
[LOCATION 'location1']
[PARTITION
(partition_colA_name = partition_colA_value
[,partition_colB_name = partition_colB_value
[,...]])]
[LOCATION 'location2']
[,...]
```

### Parámetros

Cuando agrega una partición, especifica uno o más pares de nombre y valor de columna para la partición y la ruta de acceso de Amazon S3 donde residen los archivos de datos de esa partición.

#### [IF NOT EXISTS]

Hace que se suprima el error si ya existe una partición con la misma definición.

**PARTITION** (*partition\_col\_name* = *partition\_col\_value* [,...])

Crea una partición con las combinaciones de nombre y valor de columna que especifica. Ponga *partition\_col\_value* entre caracteres de cadena solo si el tipo de datos de la columna es una cadena.

[LOCATION '*location*']

Especifica el directorio en el que se va a almacenar la partición definida mediante la instrucción anterior. La cláusula LOCATION es opcional cuando los datos están particionados al estilo Hive (*pk1=v1/pk2=v2/pk3=v3*). Con la partición al estilo Hive, el URI completo de Amazon S3 se crea automáticamente a partir de la ubicación de la tabla, los nombres de las claves de partición y los valores de las claves de partición. Para obtener más información, consulte [Particiones de datos en Athena](#).

## Consideraciones

Amazon Athena no le impone un límite específico al número de particiones que se pueden añadir en una sola instrucción DDL ALTER TABLE ADD PARTITION. Sin embargo, si necesita añadir un número significativo de particiones, considere la posibilidad de dividir la operación en lotes más pequeños para evitar posibles problemas de rendimiento. En el siguiente ejemplo, se utilizan comandos sucesivos para añadir particiones de forma individual y se utiliza IF NOT EXISTS para evitar añadir duplicados.

```
ALTER TABLE table_name ADD IF NOT EXISTS PARTITION (ds='2023-01-01')
ALTER TABLE table_name ADD IF NOT EXISTS PARTITION (ds='2023-01-02')
ALTER TABLE table_name ADD IF NOT EXISTS PARTITION (ds='2023-01-03')
```

Cuando trabaje con particiones en Athena, también tenga en cuenta los siguientes puntos:

- Aunque Athena admite consultas de tablas de AWS Glue que tienen 10 millones de particiones, no puede leer más de 1 millón de particiones en un solo escaneo.
- Para optimizar sus consultas y reducir el número de particiones escaneadas, considere estrategias como la eliminación de particiones o el uso de índices de particiones.
- Si no utiliza AWS Glue Data Catalog en este momento, el número máximo de particiones predeterminado por tabla es de 20 000. Puede solicitar un aumento de cuota.

Para obtener información adicional sobre el trabajo con particiones en Athena, consulte [Particiones de datos en Athena](#).

## Ejemplos

En el siguiente ejemplo, se agrega una sola partición a una tabla para datos particionados al estilo Hive.

```
ALTER TABLE orders ADD
PARTITION (dt = '2016-05-14', country = 'IN');
```

En el siguiente ejemplo, se agregan varias particiones a una tabla para datos particionados al estilo Hive.

```
ALTER TABLE orders ADD
PARTITION (dt = '2016-05-31', country = 'IN')
PARTITION (dt = '2016-06-01', country = 'IN');
```

Si la tabla no es para datos particionados al estilo Hive, la cláusula `LOCATION` es obligatoria y debe ser el URI completo de Amazon S3 del prefijo que contiene los datos de la partición.

```
ALTER TABLE orders ADD
PARTITION (dt = '2016-05-31', country = 'IN') LOCATION 's3://DOC-EXAMPLE-BUCKET/path/to/INDIA_31_May_2016/'
PARTITION (dt = '2016-06-01', country = 'IN') LOCATION 's3://DOC-EXAMPLE-BUCKET/path/to/INDIA_01_June_2016/';
```

Para ignorar los errores cuando la partición ya existe, utilice la cláusula `IF NOT EXISTS`, como en el siguiente ejemplo.

```
ALTER TABLE orders ADD IF NOT EXISTS
PARTITION (dt = '2016-05-14', country = 'IN');
```

## Archivos `_$folder_` sin bytes

Si ejecuta una declaración `ALTER TABLE ADD PARTITION` y especifica erróneamente una partición que ya existe y una ubicación de Amazon S3 incorrecta, se crean archivos de marcador de posición de cero bytes del formato `partition_value_$folder` en Amazon S3. Debe eliminar esos archivos en forma manual.

Para evitar que esto ocurra, utilice la sintaxis `ADD IF NOT EXISTS` de la instrucción `ALTER TABLE ADD PARTITION`, como se muestra en el siguiente ejemplo.

```
ALTER TABLE table_name ADD IF NOT EXISTS PARTITION [...]
```

## ALTER TABLE DROP PARTITION

Elimina una o varias particiones especificadas de la tabla indicada.

### Sinopsis

```
ALTER TABLE table_name DROP [IF EXISTS] PARTITION (partition_spec) [, PARTITION (partition_spec)]
```

### Parámetros

#### [IF EXISTS]

Suprime el mensaje de error si la partición especificada no existe.

#### PARTITION (partition\_spec)

Cada valor de `partition_spec` especifica una combinación de columna nombre/valor con la forma `partition_col_name = partition_col_value [, ...]`.

### Ejemplos

```
ALTER TABLE orders
DROP PARTITION (dt = '2014-05-14', country = 'IN');
```

```
ALTER TABLE orders
DROP PARTITION (dt = '2014-05-14', country = 'IN'), PARTITION (dt = '2014-05-15',
country = 'IN');
```

### Notas

La instrucción `ALTER TABLE DROP PARTITION` no proporciona una sintaxis única para eliminar todas las particiones a la vez ni admite criterios de filtro para especificar un rango de particiones que se eliminarán.

Como alternativa, puede utilizar las acciones de la API de AWS Glue [GetPartitions](#) y [BatchDeletePartition](#) durante el scripting. La acción `GetPartitions` admite expresiones de filtro complejas como las de una expresión `WHERE` de SQL. Después de usar `GetPartitions` para crear una lista filtrada de particiones para eliminar, puede utilizar la acción `BatchDeletePartition` para eliminar las particiones en lotes de 25.

### Important

Debido a un problema conocido, cuando se especifica una partición no válida para la instrucción `ALTER TABLE DROP PARTITION`, se eliminan todas las particiones de la tabla en AWS Glue. Por ejemplo, la siguiente instrucción eliminará todas las particiones de la tabla `my_table` aunque la partición especificada no exista. Como solución alternativa, asegúrese de introducir correctamente la información de la partición antes de ejecutar la instrucción `ALTER TABLE DROP PARTITION`.

```
ALTER TABLE my_table DROP IF EXISTS PARTITION(zzz='');
```

## ALTER TABLE RENAME PARTITION

Cambia el nombre de un valor de partición.

### Note

`ALTER TABLE RENAME PARTITION` no cambia el nombre de las columnas de partición. Para cambiar el nombre de una columna de partición, puede utilizar la consola de AWS Glue. Para obtener más información, consulte [Cambiar el nombre de una columna de partición en AWS Glue](#) más adelante en este documento.

## Sinopsis

En el caso de la tabla llamada `table_name`, cambia el nombre del valor de la partición especificado por `partition_spec` al valor especificado por `new_partition_spec`.

```
ALTER TABLE table_name PARTITION (partition_spec) RENAME TO PARTITION  
(new_partition_spec)
```

## Parámetros

### PARTITION (partition\_spec)

Cada valor de `partition_spec` especifica una combinación de columna nombre/valor con la forma `partition_col_name = partition_col_value [, ...]`.

## Ejemplos

```
ALTER TABLE orders
PARTITION (dt = '2014-05-14', country = 'IN') RENAME TO PARTITION (dt = '2014-05-15',
country = 'IN');
```

## Cambiar el nombre de una columna de partición en AWS Glue

Utilice el siguiente procedimiento para cambiar el nombre de las columnas de partición en la consola de AWS Glue.

Para cambiar el nombre de una columna de partición de tabla en la consola de AWS Glue

1. Inicie sesión en la AWS Management Console y abra la consola de AWS Glue en <https://console.aws.amazon.com/glue/>.
2. En el panel de navegación, elija Tablas.
3. En la página Tablas, utilice el cuadro de búsqueda Filtrar tablas para encontrar la tabla que desea modificar.
4. En la columna Nombre, elija el enlace de la tabla que desea modificar.
5. En la página de detalles de la tabla, en la sección Esquema, realice una de las siguientes acciones:
  - Para realizar el cambio de nombre en formato JSON, seleccione Editar esquema como JSON.
  - Para cambiar el nombre directamente, elija Editar esquema. Este procedimiento elige Editar esquema.
6. Seleccione la casilla de verificación de la columna particionada cuyo nombre desee cambiar y, a continuación, elija Editar.
7. En el cuadro de diálogo Editar entrada de esquema, en Nombre, ingrese el nuevo nombre de la columna de partición.



8. Elija Guardar como nueva versión de la tabla. Esta acción actualiza el nombre de columna de la partición y conserva el historial de evolución del esquema sin crear una copia física separada de los datos.
9. Para comparar versiones de tablas, en la página de detalles de la tabla, elija Acciones y, a continuación, seleccione Comparar versiones.

#### Recursos adicionales de

Para obtener más información sobre la creación de particiones, consulte [Particiones de datos en Athena](#).

## ALTER TABLE REPLACE COLUMNS

Elimina todas las columnas existentes de una tabla creada con [LazySimpleSerDe](#) y las reemplaza con el conjunto de columnas especificado. Cuando se utiliza la sintaxis `PARTITION` opcional, actualiza los metadatos de la partición. También puede utilizar `ALTER TABLE REPLACE COLUMNS` para eliminar columnas al especificar solo las columnas que desea conservar.

#### Sinopsis

```
ALTER TABLE table_name
  [PARTITION
    (partition_col1_name = partition_col1_value
    [,partition_col2_name = partition_col2_value][,...])]
  REPLACE COLUMNS (col_name data_type [, col_name data_type, ...])
```

#### Parámetros

`PARTITION (partition_col_name = partition_col_value [...])`

Especifica una partición con las combinaciones de nombre y valor de columna que especifica. Escriba `partition_col_value` entre comillas solo si el tipo de datos de la columna es una cadena.

`REPLACE COLUMNS (col_name data_type [,col_name data_type,...])`

Reemplaza las columnas existentes por los nombres de columna y los tipos de datos especificados.

## Notas

- Para ver el cambio de las columnas de la tabla en el panel de navegación del Editor de consultas de Athena después de ejecutar `ALTER TABLE REPLACE COLUMNS`, es posible que tenga que actualizar manualmente la lista de tablas en el editor y volver a expandirla.
- `ALTER TABLE REPLACE COLUMNS` no funciona para columnas con el tipo de datos `date`. Para solucionar este problema, utilice el tipo de datos `timestamp` en la tabla en su lugar.
- Tenga en cuenta que, incluso si está reemplazando una sola columna, la sintaxis debe ser `ALTER TABLE table-name REPLACE COLUMNS`, con `columns` (columnas) en plural. Debe especificar no solo la columna que desea reemplazar, sino también las columnas que desea conservar; de lo contrario, se eliminarán las columnas que no especifique. Esta sintaxis y este comportamiento se derivan del DDL de Apache Hive. Como referencia, consulte [Add/Replace Columns](#) (Agregar o reemplazar columnas) en la documentación de Apache.

## Ejemplo

En el siguiente ejemplo, la tabla `names_cities`, que se creó con [LazySimpleSerDe](#), tiene tres columnas denominadas `col1`, `col2` y `col3`. Todas las columnas son de tipo `string`. Para mostrar las columnas de la tabla, el siguiente comando utiliza la instrucción [SHOW COLUMNS](#).

```
SHOW COLUMNS IN names_cities
```

Resultado de la consulta:

```
col1  
col2  
col3
```

El siguiente comando `ALTER TABLE REPLACE COLUMNS` reemplaza los nombres de las columnas por `first_name`, `last_name` y `city`. Los datos de origen subyacentes no se ven afectados.

```
ALTER TABLE names_cities  
REPLACE COLUMNS (first_name string, last_name string, city string)
```

Para probar el resultado, `SHOW COLUMNS` se ejecuta de nuevo.

```
SHOW COLUMNS IN names_cities
```

Resultado de la consulta:

```
first_name
last_name
city
```

Otra forma de mostrar los nuevos nombres de las columnas es [obtener una vista previa de la tabla](#) en el Editor de consultas de Athena o ejecutar su propia consulta SELECT.

## ALTER TABLE SET LOCATION

Cambia la ubicación de la tabla denominada `table_name` y, de forma opcional, una partición con `partition_spec`.

Sinopsis

```
ALTER TABLE table_name [ PARTITION (partition_spec) ] SET LOCATION 'new location'
```

Parámetros

`PARTITION (partition_spec)`

Especifica la partición con parámetros `partition_spec` cuya ubicación desea cambiar. La `partition_spec` especifica una combinación de nombre de columna y valor con la forma `partition_col_name = partition_col_value`.

`SET LOCATION 'nueva ubicación'`

Especifica la nueva ubicación, que debe ser una ubicación de Amazon S3. Para obtener información sobre la sintaxis, consulte [Ubicación de las tablas en Amazon S3](#).

Ejemplos

```
ALTER TABLE customers PARTITION (zip='98040', state='WA') SET LOCATION 's3://DOC-EXAMPLE-BUCKET/custdata/';
```

## ALTER TABLE SET TBLPROPERTIES

Agrega propiedades de metadatos personalizadas o predefinidas a una tabla y establece sus valores asignados. Para ver las propiedades de una tabla, utilice el comando [SHOW TBLPROPERTIES](#).

No se admiten las [tablas administradas](#) por Apache Hive, por lo que el valor 'EXTERNAL' = 'FALSE' no tiene ningún efecto.

## Sinopsis

```
ALTER TABLE table_name SET TBLPROPERTIES ('property_name' = 'property_value' [ , ... ])
```

## Parámetros

```
SET TBLPROPERTIES ('property_name' = 'property_value' [ , ... ])
```

Especifica las propiedades de metadatos que se añaden como `property_name` y el valor de cada una como `property_value`. Si `property_name` ya existe, su valor está establecido en el `property_value` recién especificado.

Las siguientes propiedades de tabla predefinidas tienen usos especiales.

Propiedad predefinida	Descripción
<code>classification</code>	Indica el tipo de datos para AWS Glue. Los valores posibles son <code>csv</code> , <code>parquet</code> , <code>orc</code> , <code>avro</code> o <code>json</code> . Las tablas creadas para Athena en la consola de CloudTrail añaden <code>cloudtrail</code> como valor a la propiedad <code>classification</code> . Para obtener más información, consulte la sección TBLPROPERTIES de <a href="#">CREATE TABLE</a> .
<code>has_encrypted_data</code>	Indica si el conjunto de datos especificado por LOCATION está cifrado. Para obtener más información, consulte la sección TBLPROPERTIES de <a href="#">CREATE TABLE</a> y <a href="#">Creación de tablas basadas en conjuntos de datos cifrados en Amazon S3</a> .
<code>orc.compress</code>	Especifica un formato de compresión para los datos en formato ORC. Para obtener más información, consulte <a href="#">El SerDe de ORC</a> .
<code>parquet.compression</code>	Especifica un formato de compresión para los datos en formato Parquet. Para obtener más información, consulte <a href="#">El SerDe de Parquet</a> .

Propiedad predefinida	Descripción
<code>write.compression</code>	Especifica un formato de compresión para datos en formato de archivo de texto o JSON. Para los formatos Parquet y ORC, utilice las propiedades <code>parquet.compression</code> y <code>orc.compress</code> , respectivamente.
<code>compression_level</code>	Especifica el nivel de compresión que se debe utilizar. Esta propiedad solo se aplica a la compresión ZSTD. Los valores posibles están comprendidos entre 1 y 22. El valor predeterminado es 3. Para obtener más información, consulte <a href="#">Uso de los niveles de compresión ZSTD en Athena</a> .
<code>projection.*</code>	Las propiedades personalizadas utilizadas en la proyección de particiones permiten a Athena saber qué patrones de partición esperar cuando ejecuta una consulta en la tabla. Para obtener más información, consulte <a href="#">Proyección de particiones con Amazon Athena</a> .
<code>skip.header.line.count</code>	Omite los encabezados de los datos cuando se define una tabla. Para obtener más información, consulte <a href="#">Ignorar encabezados</a> .
<code>storage.location.template</code>	Especifica una plantilla de ruta de Amazon S3 personalizada para las particiones proyectadas. Para obtener más información, consulte <a href="#">Configuración de la proyección de particiones</a> .

## Ejemplos

En el siguiente ejemplo se agrega una nota de comentario a las propiedades de tabla.

```
ALTER TABLE orders
SET TBLPROPERTIES ('notes'="Please don't drop this table.");
```

En el siguiente ejemplo se modifica la tabla `existing_table` para que utilice el formato de archivo Parquet con compresión ZSTD y nivel 4 de compresión ZSTD.

```
ALTER TABLE existing_table
SET TBLPROPERTIES ('parquet.compression' = 'ZSTD', 'compression_level' = 4)
```

## CREATE DATABASE

Creando una base de datos. El uso de DATABASE y SCHEMA es intercambiable. Significan lo mismo.

### Note

Para ver un ejemplo de cómo crear una base de datos, crear una tabla y ejecutar una consulta SELECT en una tabla en Athena, consulte [Introducción](#).

## Sinopsis

```
CREATE {DATABASE|SCHEMA} [IF NOT EXISTS] database_name
  [COMMENT 'database_comment']
  [LOCATION 'S3_loc']
  [WITH DBPROPERTIES ('property_name' = 'property_value') [, ...]]
```

## Parámetros

### [IF NOT EXISTS]

Hace que se suprima el error si ya existe una base de datos denominada database\_name.

### [COMMENT database\_comment]

Establece el valor de metadato de la propiedad de metadatos integrada denominada comment y el valor que proporciona para database\_comment. En AWS Glue, los contenidos COMMENT se escriben en el campo Description de las propiedades de la base de datos.

### [LOCATION S3\_loc]

Especifica como S3\_loc la ubicación donde existirán los archivos de la base de datos y el metaalmacén. La ubicación tiene que ser una ubicación de Amazon S3.

### [WITH DBPROPERTIES ('property\_name' = 'property\_value') [, ...]]

Le permite especificar las propiedades de metadatos personalizados para la definición de la base de datos.

## Ejemplos

```
CREATE DATABASE clickstreams;
```

```
CREATE DATABASE IF NOT EXISTS clickstreams
COMMENT 'Site Foo clickstream data aggregates'
LOCATION 's3://DOC-EXAMPLE-BUCKET/clickstreams/'
WITH DBPROPERTIES ('creator'='Jane D.', 'Dept.'='Marketing analytics');
```

## Visualización de las propiedades de la base de datos

Para ver las propiedades de la base de datos que se crea en AWSDataCatalog mediante CREATE DATABASE, puede utilizar el comando [aws glue get-database](#) de la AWS CLI, como en el siguiente ejemplo:

```
aws glue get-database --name <your-database-name>
```

En la salida JSON, el resultado tiene el siguiente aspecto:

```
{
  "Database": {
    "Name": "<your-database-name>",
    "Description": "<your-database-comment>",
    "LocationUri": "s3://DOC-EXAMPLE-BUCKET",
    "Parameters": {
      "<your-database-property-name>": "<your-database-property-value>"
    },
    "CreateTime": 1603383451.0,
    "CreateTableDefaultPermissions": [
      {
        "Principal": {
          "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
        },
        "Permissions": [
          "ALL"
        ]
      }
    ]
  }
}
```

Para obtener más información acerca de la AWS CLI, consulte la [Guía del usuario de AWS Command Line Interface](#).

## CREATE TABLE

Creación de una tabla con el nombre y los parámetros que especifica.

### Note

Esta página contiene información de referencia resumida. Para obtener más información sobre la creación de tablas en Athena y una declaración de CREATE TABLE de ejemplo, consulte [Creación de tablas en Athena](#). Para ver un ejemplo de cómo crear una base de datos, crear una tabla y ejecutar una consulta SELECT en una tabla en Athena, consulte [Introducción](#).

### Sinopsis

```
CREATE EXTERNAL TABLE [IF NOT EXISTS]
  [db_name.]table_name [(col_name data_type [COMMENT col_comment] [, ...] )]
  [COMMENT table_comment]
  [PARTITIONED BY (col_name data_type [COMMENT col_comment], ...)]
  [CLUSTERED BY (col_name, col_name, ...) INTO num_buckets BUCKETS]
  [ROW FORMAT row_format]
  [STORED AS file_format]
  [WITH SERDEPROPERTIES (...)]
  [LOCATION 's3://DOC-EXAMPLE-BUCKET/[folder]/']
  [TBLPROPERTIES ( ['has_encrypted_data']='true | false',]
  ['classification']='aws_glue_classification',] property_name=property_value [, ...] ) ]
```

### Parámetros

#### EXTERNAL

Especifica que la tabla se basa en un archivo de datos subyacente que existe en Amazon S3, en la ubicación LOCATION que especifique. Utilice siempre la palabra clave EXTERNAL excepto al crear tablas de [Iceberg](#). Si utiliza CREATE TABLE sin la palabra clave EXTERNAL para tablas que no son Iceberg, Athena emite un error. Cuando cree una tabla externa, los datos a los que hace referencia deben respetar el formato predeterminado o el formato que especifique con las cláusulas ROW FORMAT, STORED AS y WITH SERDEPROPERTIES.



## [IF NOT EXISTS]

Este parámetro comprueba si ya existe una tabla con el mismo nombre. Si es así, el parámetro devuelve TRUE y Amazon Athena cancela la acción CREATE TABLE. Como la cancelación se produce antes de que Athena llame al catálogo de datos, no emite ningún evento AWS CloudTrail.

## [db\_name.]table\_name

Especifica un nombre para la tabla que se creará. El parámetro opcional db\_name especifica la base de datos donde existe la tabla. Si se omite, se presupondrá que se utiliza la base de datos actual. Si el nombre de la tabla incluye números, ponga table\_name entre comillas; por ejemplo "table123". Si table\_name comienza con un guion bajo, utilice acentos graves; por ejemplo: `\_mytable`. Los caracteres especiales, salvo el guion bajo, no se admiten.

En los nombres de tabla de Athena no se distingue entre mayúsculas y minúsculas. Sin embargo, si utiliza Apache Spark, este requiere que los nombres de las tablas estén en minúsculas.

## [ ( col\_name data\_type [COMMENT col\_comment] [, ...] ) ]

Especifica el nombre de cada columna que se va a crear, junto con el tipo de datos de la columna. Los nombres de columnas no permiten caracteres especiales que no sean guiones bajos (\_). Si col\_name comienza con un guion bajo, ponga el nombre de la columna entre acentos graves; por ejemplo `\_mycolumn`.


El valor data\_type puede ser cualquiera de los siguientes:

- **boolean**: los valores son true y false.
- **tinyint**: un número entero firmado de 8 bits en formato de complemento de dos, con un valor mínimo de  $-2^7$  y un valor máximo de  $2^7-1$ .
- **smallint**: un número entero firmado de 16 bits en formato de complemento de dos, con un valor mínimo de  $-2^{15}$  y un valor máximo de  $2^{15}-1$ .
- **int**: en las consultas de lenguaje de definición de datos (DDL), como CREATE TABLE, utilice la palabra clave int para representar un entero. En otras consultas, utilice la palabra clave integer, donde integer se representa como un valor firmado de 32 bits en formato de complemento a dos, con un valor mínimo de  $-2^{31}$  y un valor máximo de  $2^{31}-1$ . En el controlador JDBC se devuelve integer a fin de garantizar la compatibilidad con las aplicaciones de análisis empresarial.
- **bigint**: un número entero firmado de 64 bits en formato de complemento de dos, con un valor mínimo de  $-2^{63}$  y un valor máximo de  $2^{63}-1$ .

- `double`: un número firmado de punto flotante de doble precisión de 64 bits. El rango es de `4.94065645841246544e-324d` a `1.79769313486231570e+308d`, positivo o negativo. `double` sigue el estándar IEEE para la aritmética de punto flotante (IEEE 754).
- `float`: un número firmado de punto flotante de precisión simple de 32 bits. El rango es de `1.40129846432481707e-45` a `3.40282346638528860e+38`, positivo o negativo. `float` sigue el estándar IEEE para la aritmética de punto flotante (IEEE 754). Equivalente a `real` en Presto. En Athena, utilice `float` en instrucciones DDL como `CREATE TABLE` y `real` en funciones SQL como `SELECT CAST`. El rastreador de AWS Glue devuelve valores en `float`, y Athena traduce los tipos `real` y `float` internamente (consulte las notas de la versión [5 de junio de 2018](#)).
- `decimal [ (precision, scale) ]`, donde *precision* es el número total de dígitos y *scale* (opcional) es el número de dígitos con parte fraccional, el valor predeterminado es 0. Por ejemplo, utilice estas definiciones de tipo: `decimal(11,5)`, `decimal(15)`. El valor máximo para *precisión* es 38 y el valor máximo para *escala* también es 38.

Para especificar valores decimales como literales como, por ejemplo, al seleccionar filas con un valor decimal específico en una expresión DDL de consulta, especifique la definición de tipo `decimal` y enumere el valor decimal como literal (entre comillas simples) en su consulta, como en este ejemplo: `decimal_value = decimal '0.12'`.

- `char`: datos de caracteres de longitud fija, con una longitud especificada comprendida entre 1 y 255 como, por ejemplo, `char(10)`. Para obtener más información, consulte [Tipo de datos Hive CHAR](#).
- `varchar`: datos de caracteres de longitud variable, con una longitud especificada comprendida entre 1 y 65 535 como, por ejemplo, `varchar(10)`. Para obtener más información, consulte [Tipo de datos Hive VARCHAR](#).
- `string`: un literal de cadena entre comillas simples o dobles.

 Note

Los tipos de datos que no son de cadena no se pueden convertir en `string` en Athena; en su lugar, se pueden convertir en `varchar`.

- `binary`: (para los datos en formato Parquet)
- `date`: una fecha en formato ISO, como `YYYY-MM-DD`. Por ejemplo, `date '2008-09-15'`. Una excepción es `OpenCSVSerDe`, que utiliza el número de días transcurridos desde el 1 de

enero de 1970. Para obtener más información, consulte [OpenCSVSerDe para el procesamiento de CSV](#).

- `timestamp`: instante de fecha y hora en un formato compatible con [java.sql.Timestamp](#) hasta una resolución máxima de milisegundos, como `yyyy-MM-dd HH:mm:ss[.f...]`. Por ejemplo, `timestamp '2008-09-15 03:04:05.324'`. Una excepción es `OpenCSVSerDe`, que utiliza datos de `TIMESTAMP` en formato numérico UNIX (por ejemplo, `1579059880000`). Para obtener más información, consulte [OpenCSVSerDe para el procesamiento de CSV](#).
- `array < tipo_datos >`
- `map < tipo_primitiva, tipo_datos >`
- `struct < nombre_col : tipo_datos [comment comentario_col] [, ...] >`

[COMMENT table\_comment]

Crea la propiedad de tabla `comment` y la rellena con el `table_comment` que especifique.

[PARTITIONED BY (col\_name data\_type [ COMMENT col\_comment ], ... )]

Crea una tabla con particiones, con una o varias columnas de partición que tienen especificados los elementos `col_name`, `data_type` y `col_comment`. Una tabla puede tener una o varias particiones, que consisten en una combinación de nombre de columna y valor diferenciada. Se crea un directorio de datos independiente por cada combinación especificada, lo que puede mejorar el rendimiento de la consulta en algunas circunstancias. Las columnas particionadas no existen dentro de los propios datos de la tabla. Recibirá un error si utiliza un valor para `col_name` igual al de una columna de la tabla. Para obtener más información, consulte la sección sobre [partición de datos](#).

#### Note

Después de crear una tabla con particiones, ejecute una consulta posterior que consista en la cláusula [MSCK REPAIR TABLE](#) para actualizar los metadatos de partición; por ejemplo, `MSCK REPAIR TABLE c1oudfront_logs`; . Para las particiones que no son compatibles con Hive, utilice [ALTER TABLE ADD PARTITION](#) para cargar las particiones de modo que pueda consultar los datos.

[CLUSTERED BY (col\_name, col\_name, ...) INTO num\_buckets BUCKETS]

Divide los datos, con o sin particiones, de las columnas `col_name` especificadas en subconjuntos de datos llamados buckets. El parámetro `num_buckets` especifica el número de buckets que

se va a crear. La asignación de buckets puede mejorar el rendimiento de algunas consultas en conjuntos de datos grandes.

#### [ROW FORMAT row\_format]

Especifica el formato de fila de la tabla y su origen de datos subyacente, si procede. En `row_format` puede especificar uno o varios delimitadores con la cláusula `DELIMITED` o bien puede utilizar la cláusula `SERDE` tal y como se describe más abajo. Si `ROW FORMAT` se omite o bien se especifica `ROW FORMAT DELIMITED`, se utilizará un SerDe nativo.

- [DELIMITED FIELDS TERMINATED BY char [ESCAPED BY char]]
- [DELIMITED COLLECTION ITEMS TERMINATED BY char]
- [MAP KEYS TERMINATED BY char]
- [LINES TERMINATED BY char]
- [NULL DEFINED AS char]

Disponible solo con Hive 0.13 y cuando el formato de archivo `STORED AS` es `TEXTFILE`.

--O BIEN--

- `SERDE 'serde_name' [WITH SERDEPROPERTIES ("property_name" = "property_value", "property_name" = "property_value" [, ...] )]`

`serde_name` indica el SerDe que debe utilizarse. La cláusula `WITH SERDEPROPERTIES` le permite proporcionar una o varias propiedades personalizadas que el SerDe permite.

#### [STORED AS file\_format]

Especifica el formato de archivo de los datos de la tabla. Si se omite, `TEXTFILE` es la opción predeterminada. Las opciones para `file_format` son:

- SEQUENCEFILE
- TEXTFILE
- RCFILE
- ORC
- PARQUET
- AVRO
- ION

- INPUTFORMAT input\_format\_classname OUTPUTFORMAT output\_format\_classname

[LOCATION 's3://DOC-EXAMPLE-BUCKET/[folder]/']

Especifica la ubicación de los datos subyacentes en Amazon S3 a partir de los cuales se crea la tabla. La ruta de la ubicación debe ser un nombre de bucket solo o un nombre de bucket y una o varias carpetas. Si utiliza particiones, especifique la raíz de los datos particionados. Para obtener más información sobre la ubicación de la tabla, consulte [Ubicación de las tablas en Simple Storage Service \(Amazon S3\)](#). Para obtener información sobre el formato de datos y los permisos, consulte [Requisitos para las tablas en Athena y los datos en Amazon S3](#).

Utilice una barra diagonal final para su carpeta o bucket. No utilice nombres de archivo ni caracteres glob.

Uso:

s3://DOC-EXAMPLE-BUCKET/

s3://DOC-EXAMPLE-BUCKET/*folder*/

s3://DOC-EXAMPLE-BUCKET/*folder/anotherfolder*/

No utilice:

s3://DOC-EXAMPLE-BUCKET

s3://DOC-EXAMPLE-BUCKET/\*

s3://DOC-EXAMPLE-BUCKET/*mydatafile.dat*

[TBLPROPERTIES ( ['has\_encrypted\_data'='true | false',] ['classification'='classification\_value',] property\_name=property\_value [, ...] ) ]

Especifica los pares de clave-valor de metadatos personalizados para la definición de la tabla, además de las propiedades de tablas predefinidas como "comment".

**has\_encrypted\_data:** Athena tiene una propiedad integrada, `has_encrypted_data`. Establezca esta propiedad en `true` para indicar que el conjunto de datos subyacente que `LOCATION` especifica está cifrado. Si se omite y si la configuración del grupo de trabajo no invalida la configuración del lado del cliente, se supone el valor `false`. Si se omite o se establece en `false` cuando los datos subyacentes están cifrados, los resultados de la consulta generarán un error. Para obtener más información, consulte [Cifrado en reposo](#).

clasificación: las tablas creadas para Athena en la consola de CloudTrail añaden `cloudtrail` como valor a la propiedad `classification`. Para ejecutar trabajos de ETL, AWS Glue necesita que usted cree una tabla con la propiedad `classification` para indicar que el tipo de datos para AWS Glue es `csv`, `parquet`, `orc`, `avro` o `json`. Por ejemplo, `'classification'='csv'`. Los trabajos de ETL generarán un error si no especifica esta propiedad. Puede especificarla posteriormente utilizando la consola, la API o la CLI de AWS Glue. Para obtener más información, consulte [Uso de AWS Glue para realizar trabajos de ETL con Athena](#) y [Creación de trabajos en AWS Glue](#) en la Guía para desarrolladores de AWS Glue.

`compression_level`: la propiedad `compression_level` especifica el nivel de compresión que se debe utilizar. Esta propiedad solo se aplica a la compresión ZSTD. Los valores posibles están comprendidos entre 1 y 22. El valor predeterminado es 3. Para obtener más información, consulte [Uso de los niveles de compresión ZSTD en Athena](#).

Para obtener más información sobre otras propiedades de tabla, consulte [ALTER TABLE SET TBLPROPERTIES](#).

## Ejemplos

La siguiente instrucción `CREATE TABLE` de ejemplo crea una tabla basada en datos planetarios separados por tabulaciones almacenados en Amazon S3.

```
CREATE EXTERNAL TABLE planet_data (  
  planet_name string,  
  order_from_sun int,  
  au_to_sun float,  
  mass float,  
  gravity_earth float,  
  orbit_years float,  
  day_length float  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t'  
STORED AS TEXTFILE  
LOCATION 's3://DOC-EXAMPLE-BUCKET/tsv/'
```

Tenga en cuenta los siguientes puntos:

- La cláusula `ROW FORMAT DELIMITED` indica que los datos están delimitados por un carácter específico.

- La cláusula `FIELDS TERMINATED BY '\t'` especifica que los campos de los datos TSV están separados por el carácter de tabulación (“\t”).
- La cláusula `STORED AS TEXTFILE` indica que los datos se almacenan como archivos de texto sin formato en Amazon S3.

Para consultar los datos, puede utilizar una instrucción `SELECT` sencilla como la siguiente:

```
SELECT * FROM planet_data
```

Para utilizar el ejemplo de creación de una tabla TSV propia en Athena, sustituya los nombres de tabla y columna por los nombres y tipos de datos de la tabla y las columnas propias, y actualice la cláusula `LOCATION` de modo que se dirija a la ruta de Amazon S3 en la que se almacenan los archivos TSV.

Para obtener más información sobre cómo crear tablas, consulte [Creación de tablas en Athena](#).

## CREATE TABLE AS

Crea una nueva tabla rellena con los resultados de una consulta [SELECT](#). Para crear una tabla vacía, utilice [CREATE TABLE](#). `CREATE TABLE AS` combina una instrucción DDL `CREATE TABLE` con una instrucción DML `SELECT` y, por lo tanto, técnicamente contiene tanto DDL como DML. Tenga en cuenta que, aunque `CREATE TABLE AS` se agrupa aquí con otras instrucciones DDL, las consultas CTAS en Athena se tratan como DML a efectos de Service Quotas. Para obtener información acerca de Service Quotas de Athena, consulte [Service Quotas](#).

### Note

Para las instrucciones CTAS, la configuración del propietario del bucket esperado no se aplica a la ubicación de la tabla de destino en Amazon S3. La configuración del propietario esperado del bucket se aplica solo a la ubicación de salida de Amazon S3 especificada para los resultados de las consultas de Athena. Para obtener más información, consulte [Especificación de una ubicación de resultados de consulta mediante la consola de Athena](#).

Para obtener información adicional sobre `CREATE TABLE AS` más allá del alcance de este tema de referencia, consulte [Creación de una tabla a partir de los resultados de una consulta \(CTAS\)](#).

## Temas

- [Sinopsis](#)
- [Propiedades de la tabla CTAS](#)
- [Ejemplos](#)

## Sinopsis

```
CREATE TABLE table_name
[ WITH ( property_name = expression [, ...] ) ]
AS query
[ WITH [ NO ] DATA ]
```

Donde:

WITH ( property\_name = expression [, ...] )

Una lista de propiedades de tabla CTAS opcionales, algunas de las cuales son específicas del formato de almacenamiento de datos. Consulte [Propiedades de la tabla CTAS](#).

consulta

Una consulta [SELECT](#) que se utiliza para crear una tabla nueva.

### Important

Si tiene previsto crear una consulta con particiones, especifique los nombres de columnas particionadas en último lugar en la lista de columnas en la instrucción SELECT.

[ WITH [ NO ] DATA ]

Si se utiliza WITH NO DATA, se crea una nueva tabla vacía con el mismo esquema que la tabla original.

### Note

Para incluir encabezados de columna en el resultado de la consulta, puede utilizar una consulta SELECT simple en lugar de una consulta CTAS. Puede recuperar los resultados de la ubicación de los resultados de la consulta o descargar los resultados directamente



por medio de la consola de Athena. Para obtener más información, consulte [Trabajo con resultados de las consultas, consultas recientes y archivos de salida](#).

## Propiedades de la tabla CTAS

Cada tabla CTAS de Athena tiene una lista de propiedades de tabla CTAS opcionales que se especifican mediante `WITH (property_name = expression [, ...] )`. Para obtener información sobre cómo usar estos parámetros, consulte [Ejemplos de consultas CTAS](#).

**WITH (property\_name = expression [, ...], )**  
**table\_type = ['HIVE', 'ICEBERG']**

Opcional. El valor predeterminado es HIVE. Especifica el tipo de tabla de la tabla resultante.

Ejemplo:

```
WITH (table_type = 'ICEBERG')
```

**external\_location = [location]**

### Note

Como las tablas de Iceberg no son externas, esta propiedad no se aplica a ellas. Para definir la ubicación raíz de una tabla de Iceberg en una instrucción CTAS, use la propiedad `location` que se describe más adelante en esta sección.

Opcional. La ubicación en la que Athena guarda la consulta CTAS en Amazon S3.

Ejemplo:

```
WITH (external_location = 's3://DOC-EXAMPLE-BUCKET/tables/parquet_table/')
```

Athena no usa la misma ruta para los resultados de la consulta dos veces. Si especifica la ubicación de forma manual, asegúrese de que la ubicación de Amazon S3 que especifica no tenga datos. Athena nunca intenta eliminar los datos. Si desea utilizar la misma ubicación de nuevo, borre manualmente los datos; de lo contrario, su consulta CTAS producirá un error.

Si ejecuta una consulta CTAS que especifica un `external_location` en un grupo de trabajo que [aplica una ubicación de resultados de consulta](#), la consulta devuelve un mensaje de error. Para ver la ubicación de los resultados de la consulta especificada para el grupo de trabajo, [consulte los detalles del grupo de trabajo](#).

Si el grupo de trabajo anula la configuración del lado cliente para la ubicación de los resultados de la consulta, Athena crea la tabla en la siguiente ubicación:

```
s3://DOC-EXAMPLE-BUCKET/tables/query-id/
```

Si no utiliza la propiedad `external_location` para especificar una ubicación y el grupo de trabajo no anula la configuración del lado del cliente, Athena utiliza la [configuración del lado del cliente](#) para la ubicación de resultados de consulta para crear la tabla en la siguiente ubicación:

```
s3://DOC-EXAMPLE-BUCKET/Unsaved-or-query-name/year/month/date/tables/query-id/
```

### **is\_external = [boolean]**

Opcional. Indica si la tabla es externa. El valor predeterminado es true. Para las tablas de Iceberg, se debe establecer en false.

Ejemplo:

```
WITH (is_external = false)
```

### **location = [location]**

Obligatoria para las tablas de Iceberg. Especifica la ubicación raíz de la tabla de Iceberg que se va a crear a partir de los resultados de la consulta.

Ejemplo:

```
WITH (location = 's3://DOC-EXAMPLE-BUCKET/tables/iceberg_table/')
```

### **field\_delimiter = [delimiter]**

Opcional y específico de los formatos de almacenamiento de datos basados en texto. El delimitador de campo de un solo carácter para los archivos en CSV, TSV y archivos de texto. Por ejemplo, `WITH (field_delimiter = ',')`. Actualmente, no se admiten delimitadores

de campos de caracteres múltiples para las consultas CTAS. Si no especifica un delimitador de campo, se utiliza `\001` de forma predeterminada.

### **format = [storage\_format]**

El formato de almacenamiento de los resultados de las consultas CTAS, como ORC, PARQUET, AVRO, JSON, ION o TEXTFILE. Los formatos permitidos para las tablas de Iceberg son ORC, PARQUET y AVRO. Si se omite, se utiliza PARQUET de forma predeterminada. El nombre de este parámetro `format`, debe incluirse en minúsculas o la consulta CTAS producirá un error.

Ejemplo:

```
WITH (format = 'PARQUET')
```

### **bucketed\_by = ARRAY[ column\_name[,...], bucket\_count = [int] ]**

#### Note

Esta propiedad no se aplica a las tablas de Iceberg. Para las tablas de Iceberg, use el particionamiento con transformación de buckets.

Una matriz de buckets de datos. Si se omite, Athena no crea un bucket con sus datos en esta consulta.

### **bucket\_count = [int]**

#### Note

Esta propiedad no se aplica a las tablas de Iceberg. Para las tablas de Iceberg, use el particionamiento con transformación de buckets.


El número de buckets de sus datos. Si se omite, Athena no crea un bucket con sus datos.

Ejemplo:

```
CREATE TABLE bucketed_table WITH (  
  bucketed_by = ARRAY[column_name],  
  bucket_count = 30, format = 'PARQUET',  
  external_location = 's3://DOC-EXAMPLE-BUCKET/tables/parquet_table/'  
) AS  
SELECT
```

```
*
FROM
  table_name
```

**partitioned\_by = ARRAY[ col\_name[,...] ]**

 Note

Esta propiedad no se aplica a las tablas de Iceberg. Para usar las transformaciones de partición en las tablas de Iceberg, use la propiedad `partitioning` que se describe más adelante en esta sección.

Opcional. Una matriz de columnas por las que se particiona la tabla CTAS. Compruebe que los nombres de las columnas particionadas aparecen en último lugar en la lista de columnas en la instrucción SELECT.

**partitioning = ARRAY[partition\_transform, ...]**

Opcional. Especifica el particionamiento de la tabla de Iceberg que se va a crear. Iceberg admite una amplia variedad de transformaciones y evoluciones de partición. Las transformaciones de partición se resumen en la siguiente tabla.

Transformar	Descripción
<code>year(ts)</code>	Crea una partición para cada año. El valor de la partición es la diferencia de años, expresada con números enteros, entre <code>ts</code> y el 1 de enero de 1970.
<code>month(ts)</code>	Crea una partición para cada mes de cada año. El valor de la partición es la diferencia de meses, expresada con números enteros, entre <code>ts</code> y el 1 de enero de 1970.
<code>day(ts)</code>	Crea una partición para cada día de cada año. El valor de la partición es la diferencia de días, expresada con números enteros, entre <code>ts</code> y el 1 de enero de 1970.
<code>hour(ts)</code>	Crea una partición para cada hora de cada día. El valor de la partición es una marca de tiempo con los minutos y segundos establecidos en cero.

Transformar	Descripción
<code>bucket(x, nbuckets)</code>	Coloca los datos en el número especificado de buckets. El valor de la partición es un valor de hash entero de <code>x</code> , comprendido entre 0 y <code>nbuckets - 1</code> , ambos inclusive.
<code>truncate(s, nchars)</code>	Hace que el valor de la partición sea los <code>nchars</code> primeros caracteres de <code>s</code> .

Ejemplo:

```
WITH (partitioning = ARRAY['month(order_date)',
                           'bucket(account_number, 10)',
                           'country']))
```

### **optimize\_rewrite\_min\_data\_file\_size\_bytes = [long]**

Opcional. Configuración específica de optimización de datos. Los archivos más pequeños que el valor especificado se incluyen para la optimización. El valor predeterminado es 0,75 veces el valor de `write_target_data_file_size_bytes`. Esta propiedad se aplica solo a las tablas de Iceberg. Para obtener más información, consulte [Optimización de las tablas de Iceberg](#).

Ejemplo:

```
WITH (optimize_rewrite_min_data_file_size_bytes = 402653184)
```

### **optimize\_rewrite\_max\_data\_file\_size\_bytes = [long]**

Opcional. Configuración específica de optimización de datos. Los archivos más grandes que el valor especificado se incluyen para la optimización. El valor predeterminado es 1,8 veces el valor de `write_target_data_file_size_bytes`. Esta propiedad se aplica solo a las tablas de Iceberg. Para obtener más información, consulte [Optimización de las tablas de Iceberg](#).

Ejemplo:

```
WITH (optimize_rewrite_max_data_file_size_bytes = 966367641)
```

**optimize\_rewrite\_data\_file\_threshold = [int]**

Opcional. Configuración específica de optimización de datos. Si los archivos de datos que requieren optimización son menores que el límite determinado, los archivos no se reescriben. Esto permite acumular más archivos de datos para producir archivos más cercanos al tamaño objetivo y omitir cálculos innecesarios para ahorrar costos. El valor predeterminado es 5. Esta propiedad se aplica solo a las tablas de Iceberg. Para obtener más información, consulte [Optimización de las tablas de Iceberg](#).

Ejemplo:

```
WITH (optimize_rewrite_data_file_threshold = 5)
```

**optimize\_rewrite\_delete\_file\_threshold = [int]**

Opcional. Configuración específica de optimización de datos. Si los archivos de eliminación asociados a un archivo de datos son menores que el límite, el archivo de datos no se reescribe. Esto permite acumular más archivos de eliminación para cada archivo de datos para ahorrar costos. El valor predeterminado es 2. Esta propiedad se aplica solo a las tablas de Iceberg. Para obtener más información, consulte [Optimización de las tablas de Iceberg](#).

Ejemplo:

```
WITH (optimize_rewrite_delete_file_threshold = 2)
```

**vacuum\_min\_snapshots\_to\_keep = [int]**

Opcional. Configuración específica de Vacuum. El número mínimo de instantáneas más recientes que se deben retener. El valor predeterminado de es 1. Esta propiedad se aplica solo a las tablas de Iceberg. Para obtener más información, consulte [VACUUM](#).

**Note**

La propiedad `vacuum_min_snapshots_to_keep` requiere la versión 3 del motor Athena.

Ejemplo:

```
WITH (vacuum_min_snapshots_to_keep = 1)
```

**vacuum\_max\_snapshot\_age\_seconds = [long]**

Opcional. Configuración específica de Vacuum. Un periodo en segundos que representa la antigüedad de las instantáneas que se van a retener. El valor predeterminado es 432 000 (5 días). Esta propiedad se aplica solo a las tablas de Iceberg. Para obtener más información, consulte [VACUUM](#).

**Note**

La propiedad `vacuum_max_snapshot_age_seconds` requiere la versión 3 del motor Athena.

Ejemplo:

```
WITH (vacuum_max_snapshot_age_seconds = 432000)
```

**write\_compression = [compression\_format]**

Tipo de compresión que se va a utilizar para cualquier formato de almacenamiento que permita especificar la compresión. El valor `compression_format` especifica la compresión que se utilizará cuando los datos se escriben en la tabla. Puede especificar la compresión para los formatos de archivo TEXTFILE, JSON, PARQUET y ORC.

Por ejemplo, si la propiedad `format` especifica PARQUET como formato de almacenamiento, el valor de `write_compression` especifica el formato de compresión para Parquet. En este caso, especificar un valor para `write_compression` equivale a especificar un valor para `parquet_compression`.

De igual modo, si la propiedad `format` especifica ORC como formato de almacenamiento, el valor de `write_compression` especifica el formato de compresión para ORC. En este caso, especificar un valor para `write_compression` equivale a especificar un valor para `orc_compression`.

No se pueden especificar varias propiedades de tabla de formato de compresión en la misma consulta CTAS. Por ejemplo, no puede especificar `write_compression` y `parquet_compression` en la misma consulta. Lo mismo se aplica para `write_compression` y `orc_compression`. Para obtener información sobre los tipos

de compresión admitidos para cada formato de archivo, consulte [Compatibilidad con la compresión de Athena](#).

### **orc\_compression = [compression\_format]**

Tipo de compresión que se utiliza para el formato de archivo ORC cuando se escriben los datos de ORC en la tabla. Por ejemplo, WITH (`orc_compression = 'ZLIB'`). Los fragmentos del archivo ORC (excepto el ORC Postscript) se comprimen con la compresión que especifique. Si se omite, se utiliza la compresión ZLIB de forma predeterminada para ORC.

#### Note

Por motivos de coherencia, le recomendamos que utilice la propiedad `write_compression` en lugar de `orc_compression`. Utilice la propiedad `format` para especificar el formato de almacenamiento como ORC y, a continuación, utilice la propiedad `write_compression` para especificar el formato de compresión que utilizará ORC.

### **parquet\_compression = [compression\_format]**

Tipo de compresión que se utiliza para el formato de archivo Parquet cuando se escriben los datos de Parquet en la tabla. Por ejemplo, WITH (`parquet_compression = 'SNAPPY'`). Esta compresión se aplica a los fragmentos de columna de los archivos Parquet. Si se omite, se utiliza la compresión GZIP de forma predeterminada para Parquet.

#### Note

Por motivos de coherencia, le recomendamos que utilice la propiedad `write_compression` en lugar de `parquet_compression`. Utilice la propiedad `format` para especificar el formato de almacenamiento como PARQUET y, a continuación, utilice la propiedad `write_compression` para especificar el formato de compresión que utilizará PARQUET.

### **compression\_level = [compression\_level]**

Nivel de compresión que se debe utilizar. Esta propiedad solo se aplica a la compresión ZSTD. Los valores posibles están comprendidos entre 1 y 22. El valor predeterminado es 3. Para obtener más información, consulte [Uso de los niveles de compresión ZSTD en Athena](#).



## Ejemplos

Para obtener ejemplos de consultas CTAS, consulte los siguientes recursos.

- [Ejemplos de consultas CTAS](#)
- [Uso de CTAS e INSERT INTO en ETL y análisis de datos](#)
- [Use CTAS statements with Amazon Athena to reduce cost and improve performance](#) (Utilice instrucciones CTAS con Amazon Athena para reducir los costos y mejorar el rendimiento)
- [Uso de CTAS e INSERT INTO para evitar el límite de 100 particiones](#)

## CREATE VIEW

Crea una nueva vista de una consulta SELECT especificada. La vista es una tabla lógica a la que se puede hacer referencia en futuras consultas. Las vistas no contienen datos ni escriben datos. En su lugar, la consulta especificada por la vista se ejecuta cada vez que se hace referencia a la vista en otra consulta.

### Note

En este tema se proporciona un resumen de información de referencia. Para obtener información más detallada acerca del uso de vistas en Athena, consulte [Uso de vistas](#). Para obtener información acerca de limitaciones de vistas, consulte [Limitaciones de las vistas](#).

## Sinopsis

```
CREATE [ OR REPLACE ] VIEW view_name AS query
```

La cláusula opcional OR REPLACE le permite actualizar la vista existente sustituyéndola. Para obtener más información, consulte [Creación de vistas](#).

## Ejemplos

Para crear una vista test a partir de la tabla orders, utilice una consulta similar a la siguiente:

```
CREATE VIEW test AS
```

```
SELECT
orderkey,
orderstatus,
totalprice / 2 AS half
FROM orders;
```

Para crear una vista `orders_by_date` a partir de la tabla `orders`, utilice la siguiente consulta:

```
CREATE VIEW orders_by_date AS
SELECT orderdate, sum(totalprice) AS price
FROM orders
GROUP BY orderdate;
```

Para actualizar una vista existente, utilice un ejemplo similar al siguiente:

```
CREATE OR REPLACE VIEW test AS
SELECT orderkey, orderstatus, totalprice / 4 AS quarter
FROM orders;
```

Consulte también [SHOW COLUMNS](#), [SHOW CREATE VIEW](#), [DESCRIBE VIEW](#) y [DROP VIEW](#).

## DESCRIBE

Muestra una o más columnas, incluidas las columnas de partición, de la tabla especificada. Este comando resulta útil para examinar los atributos de columnas complejas.

### Sinopsis

```
DESCRIBE [EXTENDED | FORMATTED] [db_name.]table_name [PARTITION partition_spec]
[col_name ( [.field_name] | [.'$elem$'] | [.'$key$'] | [.'$value$'] )]
```

#### Important

La sintaxis de esta instrucción es `DESCRIBE table_name`, no `DESCRIBE TABLE table_name`. El uso de la última sintaxis da como resultado el mensaje de error FAILED: SemanticException [Error 10001]: Table not found table (ERROR: Excepción semántica [Error 10001]: tabla no encontrada).

## Parámetros

### [EXTENDED | FORMATTED]

Determina el formato de la salida. Si se omiten estos parámetros, se muestran los nombres de las columnas y sus tipos de datos correspondientes, incluidas las columnas de partición, en formato tabular. Si se especifica FORMATTED, se muestran no solo los nombres de las columnas y los tipos de datos en formato tabular, sino también información detallada de las tablas y el almacenamiento. EXTENDED muestra información de columnas y tipos de datos en formato tabular y metadatos detallados de la tabla en formato serializado de Thrift. Este formato es menos legible y es útil principalmente para depurar.

### [PARTITION partition\_spec]

Si se incluye, enumera los metadatos de la partición que especifica `partition_spec`, donde `partition_spec` tiene el formato (`partition_column = partition_col_value, partition_column = partition_col_value, ...`).

### [col\_name ( [.field\_name] | [ '\$elem\$' ] | [ '\$key\$' ] | [ '\$value\$' ] ) \* ]

Especifica la columna y los atributos que deben examinarse. Puede especificar `.field_name` para un elemento de una estructura, `'$elem$'` para un elemento matriz, `'$key$'` para una clave de asignación y `'$value$'` para un valor de mapa. Puede especificarlo recursivamente para explorar en profundidad la columna compleja.

## Ejemplos

```
DESCRIBE orders
```

```
DESCRIBE FORMATTED mydatabase.mytable PARTITION (part_col = 100) columnA;
```

En la consulta y el resultado siguientes, se muestra información de columnas y tipos de datos de una tabla `impressions` basada en los datos de muestra de Amazon EMR.

```
DESCRIBE impressions
```

```
requestbegintime      string      from
  deserializer
adid                   string      from
  deserializer
```

```

impressionid      string      from
  deserializer
referrer          string      from
  deserializer
useragent         string      from
  deserializer
usercookie        string      from
  deserializer
ip                string      from
  deserializer
number            string      from
  deserializer
processid         string      from
  deserializer
browsercookie     string      from
  deserializer
requestendtime    string      from
  deserializer
timers            struct<modelllookup:string,requesttime:string> from
  deserializer
threadid          string      from
  deserializer
hostname          string      from
  deserializer
sessionid         string      from
  deserializer
dt                string

# Partition Information
# col_name        data_type      comment

dt                string

```

En la consulta y el resultado de ejemplo siguientes, se muestra el resultado de la misma tabla cuando se utiliza la opción FORMATTED.

```
DESCRIBE FORMATTED impressions
```

```

requestbegintime  string      from
  deserializer
adid              string      from
  deserializer

```

```

impressionid      string      from
  deserializer
referrer          string      from
  deserializer
useragent         string      from
  deserializer
usercookie        string      from
  deserializer
ip                string      from
  deserializer
number            string      from
  deserializer
processid         string      from
  deserializer
browsercookie    string      from
  deserializer
requestendtime    string      from
  deserializer
timers            struct<modelllookup:string,requesttime:string> from
  deserializer
threadid          string      from
  deserializer
hostname          string      from
  deserializer
sessionid         string      from
  deserializer
dt                string

```

## # Partition Information

```
# col_name      data_type      comment
```

```
dt              string
```

## # Detailed Table Information

```

Database:      sampledb
Owner:         hadoop
CreateTime:    Thu Apr 23 02:55:21 UTC 2020
LastAccessTime: UNKNOWN
Protect Mode:  None
Retention:     0
Location:      s3://us-east-1.elasticmapreduce/samples/hive-ads/tables/
impressions
Table Type:    EXTERNAL_TABLE
Table Parameters:

```

```

EXTERNAL          TRUE
transient_lastDdlTime 1587610521

# Storage Information
SerDe Library:      org.openx.data.jsonserde.JsonSerDe
InputFormat:       org.apache.hadoop.mapred.TextInputFormat
OutputFormat:
  org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat
Compressed:        No
Num Buckets:       -1
Bucket Columns:    []
Sort Columns:      []
Storage Desc Params:
  paths              requestbegtintime, adid, impressionid,
  referrer, useragent, usercookie, ip
  serialization.format 1

```

En la consulta y el resultado de ejemplo siguientes, se muestra el resultado de la misma tabla cuando se utiliza la opción EXTENDED. La información detallada de la tabla se muestra en una sola línea, pero se formatea aquí para que sea legible.

```
DESCRIBE EXTENDED impressions
```

```

requestbegtintime      string      from
  deserializer
adid                   string      from
  deserializer
impressionid           string      from
  deserializer
referrer               string      from
  deserializer
useragent              string      from
  deserializer
usercookie             string      from
  deserializer
ip                     string      from
  deserializer
number                 string      from
  deserializer
processid              string      from
  deserializer

```

```

browsercookie      string      from
  deserializer
requestendtime     string      from
  deserializer
timers             struct<modelllookup:string,requesttime:string> from
  deserializer
threadid          string      from
  deserializer
hostname           string      from
  deserializer
sessionid         string      from
  deserializer
dt                string

```

#### # Partition Information

```

# col_name          data_type          comment

dt                string

```

```

Detailed Table Information      Table(tableName:impressions, dbName:sampled,
  owner:hadoop, createTime:1587610521,
  lastAccessTime:0, retention:0, sd:StorageDescriptor(cols:
  [FieldSchema(name:requeststarttime, type:string, comment:null),
  FieldSchema(name:adid, type:string, comment:null), FieldSchema(name:impressionid,
  type:string, comment:null),
  FieldSchema(name:referrer, type:string, comment:null), FieldSchema(name:useragent,
  type:string, comment:null),
  FieldSchema(name:usercookie, type:string, comment:null), FieldSchema(name:ip,
  type:string, comment:null),
  FieldSchema(name:number, type:string, comment:null), FieldSchema(name:processid,
  type:string, comment:null),
  FieldSchema(name:browsercookie, type:string, comment:null),
  FieldSchema(name:requestendtime, type:string, comment:null),
  FieldSchema(name:timers, type:struct<modelllookup:string,requesttime:string>,
  comment:null), FieldSchema(name:threadid,
  type:string, comment:null), FieldSchema(name:hostname, type:string, comment:null),
  FieldSchema(name:sessionid,
  type:string, comment:null)], location:s3://us-east-1.elasticmapreduce/samples/hive-ads/
  tables/impressions,
  inputFormat:org.apache.hadoop.mapred.TextInputFormat,
  outputFormat:org.apache.hadoop.hive ql.io.IgnoreKeyTextOutputFormat, compressed:false,
  numBuckets:-1,
  serdeInfo:SerDeInfo(name:null, serializationLib:org.openx.data.jsonserde.JsonSerDe,
  parameters:{serialization.format=1,

```

```
paths=requestbegintime, adid, impressionid, referrer, useragent, usercookie, ip}),
  bucketCols:[], sortCols:[], parameters:{},
skewedInfo:SkewedInfo(skewedColNames:[], skewedColValues:[],
  skewedColValueLocationMaps:{}),
storedAsSubDirectories:false), partitionKeys:[FieldSchema(name:dt, type:string,
  comment:null)],
parameters:{EXTERNAL=TRUE, transient_lastDdlTime=1587610521}, viewOriginalText:null,
  viewExpandedText:null,
tableType:EXTERNAL_TABLE)
```

## DESCRIBE VIEW

Muestra la lista de columnas de la vista citada. Esto le permite examinar los atributos de una vista compleja.

### Sinopsis

```
DESCRIBE [db_name.]view_name
```

### Ejemplo

```
DESCRIBE orders;
```

Consulte también [SHOW COLUMNS](#), [SHOW CREATE VIEW](#), [SHOW VIEWS](#) y [DROP VIEW](#).

## DROP DATABASE

Borra del catálogo la base de datos nombrada. Si la base de datos contiene tablas, debe anular las tablas antes de ejecutar DROP DATABASE o bien utilizar la cláusula CASCADE. El uso de DATABASE y SCHEMA es intercambiable. Significan lo mismo.

### Sinopsis

```
DROP {DATABASE | SCHEMA} [IF EXISTS] database_name [RESTRICT | CASCADE]
```

### Parámetros

#### [IF EXISTS]

Hace que se suprima el error si database\_name no existe.



## [RESTRICT|CASCADE]

Determina cómo se consideran las tablas de dentro de `database_name` durante la operación DROP. Si especifica RESTRICT, la base de datos no se anula si contiene tablas. Este es el comportamiento predeterminado. Si especifica CASCADE, se anularán la base de datos y todas sus tablas.

### Ejemplos

```
DROP DATABASE clickstreams;
```

```
DROP SCHEMA IF EXISTS clickstreams CASCADE;
```

#### Note

Cuando intenta eliminar una base de datos cuyo nombre contenga caracteres especiales (por ejemplo, `my-database`), puede recibir un mensaje de error. Para resolver este problema, pruebe a incluir el nombre de la base de datos entre tildes invertidas (```). Para más información sobre la asignación de nombres de bases de datos en Athena, consulte [Nombres de tablas, bases de datos y columnas](#).

## DROP TABLE

Elimina la definición de mesa de metadatos de la tabla llamada `table_name`. Cuando se anula una tabla externa, los datos subyacentes permanecen intactos.

### Sinopsis

```
DROP TABLE [IF EXISTS] table_name
```

### Parámetros

#### [ IF EXISTS ]

Hace que se suprima el error si `table_name` no existe.

## Ejemplos

```
DROP TABLE fulfilled_orders
```

```
DROP TABLE IF EXISTS fulfilled_orders
```

Cuando utilice el Editor de consultas de la consola de Athena para eliminar una tabla que tenga caracteres especiales distintos del guion bajo (\_), utilice acentos graves, como en el ejemplo siguiente.

```
DROP TABLE `my-athena-database-01.my-athena-table`
```

Cuando se utiliza el conector JDBC para eliminar una tabla que tiene caracteres especiales, no se requieren acentos graves.

```
DROP TABLE my-athena-database-01.my-athena-table
```

## DROP VIEW

Elimina una vista existente. La cláusula `IF EXISTS` opcional provoca la eliminación del error si la vista no existe.

Para obtener más información, consulte [Uso de vistas](#).

### Sinopsis

```
DROP VIEW [ IF EXISTS ] view_name
```

## Ejemplos

```
DROP VIEW orders_by_date
```

```
DROP VIEW IF EXISTS orders_by_date
```

Consulte también [CREATE VIEW](#), [SHOW COLUMNS](#), [SHOW CREATE VIEW](#), [SHOW VIEWS](#) y [DESCRIBE VIEW](#).

## MSCK REPAIR TABLE

Utilice el comando `MSCK REPAIR TABLE` para actualizar los metadatos del catálogo después de agregar particiones compatibles con Hive.

El comando `MSCK REPAIR TABLE` analiza un sistema de archivos como Amazon S3 en busca de particiones compatibles con Hive que se agregaron al sistema de archivos después de crear la tabla. `MSCK REPAIR TABLE` compara las particiones en los metadatos de la tabla y las particiones en S3. Si hay nuevas particiones en la ubicación de S3 que especificó al crear la tabla, agrega esas particiones a los metadatos y a la tabla Athena.

Al agregar particiones físicas, los metadatos del catálogo se vuelven incoherentes con el diseño de los datos en el sistema de archivos, y la información sobre las nuevas particiones debe agregarse al catálogo. Para actualizar los metadatos, ejecute `MSCK REPAIR TABLE` para consultar los datos en las nuevas particiones de Athena.

### Note

`MSCK REPAIR TABLE` solo agrega particiones a los metadatos; no las elimina. Para eliminar particiones de los metadatos después de haber eliminado las particiones manualmente en Amazon S3, ejecute el comando `ALTER TABLE table-name DROP PARTITION`. Para obtener más información, consulte [ALTER TABLE DROP PARTITION](#).

### Consideraciones y limitaciones

Al utilizar `MSCK REPAIR TABLE`, tenga en cuenta los siguientes puntos:

- Es posible que tarde un tiempo en añadir todas las particiones. Si esta operación agota el tiempo de espera, se encontrará en un estado incompleto en el que solo se habrán añadido unas cuantas particiones al catálogo. Debe ejecutar `MSCK REPAIR TABLE` en la misma tabla hasta que se hayan agregado todas las particiones. Para obtener más información, consulte [Particiones de datos en Athena](#).
- En el caso de las particiones que no son compatibles con Hive, utilice [ALTER TABLE ADD PARTITION](#) para cargar las particiones de modo que pueda consultar los datos.
- Las ubicaciones de partición que se van a utilizar con Athena deben utilizar el protocolo `s3` (por ejemplo, `s3://DOC-EXAMPLE-BUCKET/folder/`). En Athena, las ubicaciones que utilizan otros protocolos (por ejemplo, `s3a://bucket/folder/`) producirán errores de consulta cuando se ejecuten consultas `MSCK REPAIR TABLE` en las tablas que contienen.

- Dado que `MSCK REPAIR TABLE` analiza una carpeta y sus subcarpetas para encontrar un esquema de partición coincidente, asegúrese de mantener los datos de tablas separadas en jerarquías de carpetas separadas. Por ejemplo, supongamos que tiene los datos de la tabla 1 en `s3://DOC-EXAMPLE-BUCKET1` y los datos de la tabla 2 en `s3://DOC-EXAMPLE-BUCKET1/table-2-data`. Si ambas tablas están particionadas por cadena, `MSCK REPAIR TABLE` agregará las particiones de la tabla 2 a la tabla 1. Para evitarlo, utilice estructuras de carpetas independientes, como `s3://DOC-EXAMPLE-BUCKET1` y `s3://DOC-EXAMPLE-BUCKET2` en su lugar. Tenga en cuenta que este comportamiento es coherente con Amazon EMR y Apache Hive.
- Debido a un problema conocido, `MSCK REPAIR TABLE` falla silenciosamente cuando los valores de partición contienen el carácter de dos puntos (`:`) (por ejemplo, cuando el valor de la partición es una marca de tiempo). Como alternativa, utilice [ALTER TABLE ADD PARTITION](#).
- `MSCK REPAIR TABLE` no agrega nombres de columnas de particiones que comiencen por un carácter de guion bajo (`_`). Para evitar este límite, utilice [ALTER TABLE ADD PARTITION](#).

## Sinopsis

```
MSCK REPAIR TABLE table_name
```

## Ejemplos

```
MSCK REPAIR TABLE orders;
```

## Resolución de problemas

Después de ejecutar `MSCK REPAIR TABLE`, si Athena no agrega las particiones a la tabla en AWS Glue Data Catalog, verifique lo siguiente:

- Acceso a AWS Glue: asegúrese de que el rol de AWS Identity and Access Management (IAM) tenga una política que permita la acción `glue:BatchCreatePartition`. Para obtener más información, consulte [Permitir glue:BatchCreatePartition en la política de IAM](#) más adelante en este documento.
- Acceso a Amazon S3: asegúrese de que el rol tenga una política con permisos suficientes para acceder a Amazon S3, incluida la acción [s3:DescribeJob](#). Para obtener un ejemplo de las acciones de Amazon S3 que debe permitir, consulte el ejemplo de política de bucket en [Acceso entre cuentas en Athena a los buckets de Simple Storage Service \(Amazon S3\)](#).
- Mayúsculas en las claves de objeto en Amazon S3: asegúrese de que la ruta de Amazon S3 esté escrita en minúsculas en lugar de una combinación de mayúsculas y minúsculas (por ejemplo,

`userid` en lugar de `userId`) o use `ALTER TABLE ADD PARTITION` para especificar los nombres de las claves de objeto. Para obtener más información, consulte [Cambiar o redefinir la ruta de Amazon S3](#) más adelante en este documento.

- Tiempos de espera de consulta: `MSCK REPAIR TABLE` se utiliza mejor al crear una tabla por primera vez o cuando hay incertidumbre acerca de la paridad entre los datos y los metadatos de partición. Si utiliza `MSCK REPAIR TABLE` para agregar particiones nuevas con frecuencia (por ejemplo, a diario) y experimentan tiempos de espera de consulta, considere usar [ALTER TABLE ADD PARTITION](#).
- Faltan particiones en el sistema de archivos: si elimina una partición de forma manual en Amazon S3 y, a continuación, ejecuta `MSCK REPAIR TABLE`, puede recibir el mensaje de error Faltan particiones en el sistema de archivos. Esto sucede porque `MSCK REPAIR TABLE` no elimina particiones obsoletas de los metadatos de la tabla. Para quitar las particiones eliminadas de los metadatos de la tabla, ejecute [ALTER TABLE DROP PARTITION](#) en su lugar. Tenga en cuenta que [SHOW PARTITIONS](#) (Mostrar particiones) muestra de manera similar solo las particiones en los metadatos, no las particiones en el sistema de archivos.
- Error “NullPointerException name is null” (El nombre NullPointerException es nulo)

Si utiliza la operación de la API [CreateTable](#) de AWS Glue o la plantilla [AWS::Glue::Table](#) de AWS CloudFormation para crear una tabla para su uso en Athena sin especificar la propiedad `TableType`, y luego ejecuta una consulta DDL como `SHOW CREATE TABLE` o `MSCK REPAIR TABLE`, puede recibir el mensaje de error `FAILED: NullPointerException Name is null (ERROR: el nombre NullPointerException es nulo)`.

Para resolver el error, especifique un valor para el atributo `TableType` [TableInput](#) como parte de la llamada a la API `CreateTable` de AWS Glue o la [plantilla AWS CloudFormation](#). Entre los valores posibles para `TableType`, se incluyen `EXTERNAL_TABLE` o `VIRTUAL_VIEW`.

Este requisito se aplica únicamente cuando se crea una tabla mediante la operación de la API `CreateTable` de AWS Glue o la plantilla `AWS::Glue::Table`. Si crea una tabla para Athena mediante una instrucción DDL o un rastreador de AWS Glue, la propiedad `TableType` se define automáticamente.

Las secciones siguientes proporcionan algunos detalles adicionales.

## Permitir glue:BatchCreatePartition en la política de IAM

Revise las políticas de IAM asociadas al rol que utiliza para ejecutar `MSCK REPAIR TABLE`. Cuando [utiliza AWS Glue Data Catalog con Athena](#), la política de IAM debe permitir la acción `glue:BatchCreatePartition`. Para obtener un ejemplo de una política de IAM que permita la acción `glue:BatchCreatePartition`, consulte [Política administrada de AWS: AmazonAthenaFullAccess](#).

## Cambiar o redefinir la ruta de Amazon S3

Si una o más claves de objeto de la ruta de Amazon S3 están en mayúsculas y minúsculas y no solo en minúsculas, `MSCK REPAIR TABLE` podría no agregar las particiones a AWS Glue Data Catalog. Por ejemplo, si la ruta de Amazon S3 incluye el nombre de la clave del objeto `userId`, es posible que las siguientes particiones no se agreguen al AWS Glue Data Catalog:

```
s3://DOC-EXAMPLE-BUCKET/path/userId=1/
s3://DOC-EXAMPLE-BUCKET/path/userId=2/
s3://DOC-EXAMPLE-BUCKET/path/userId=3/
```

Para resolver este problema, siga uno de estos pasos:

- Utilice minúsculas en lugar de una combinación de mayúsculas y minúsculas cuando cree las claves de objeto de Amazon S3:

```
s3://DOC-EXAMPLE-BUCKET/path/userid=1/
s3://DOC-EXAMPLE-BUCKET/path/userid=2/
s3://DOC-EXAMPLE-BUCKET/path/userid=3/
```

- Use [ALTER TABLE ADD PARTITION](#) para redefinir la ubicación, como en el siguiente ejemplo:

```
ALTER TABLE table_name ADD [IF NOT EXISTS]
PARTITION (userId=1)
LOCATION 's3://DOC-EXAMPLE-BUCKET/path/userId=1/'
PARTITION (userId=2)
LOCATION 's3://DOC-EXAMPLE-BUCKET/path/userId=2/'
PARTITION (userId=3)
LOCATION 's3://DOC-EXAMPLE-BUCKET/path/userId=3/'
```

Tenga en cuenta que, aunque los nombres de las claves de objeto de Amazon S3 pueden estar en mayúsculas, los nombres de los buckets de Amazon S3 deben estar siempre en minúsculas. Para obtener más información, consulte las [pautas de nomenclatura de claves de objeto](#) y las [reglas de nomenclatura de buckets](#) en la Guía del usuario de Amazon S3.

## SHOW COLUMNS

Solo muestra los nombres de las columnas de una única tabla o vista especificada. Para obtener información más detallada, consulte el AWS Glue Data Catalog en su lugar. Para obtener información y ejemplos, consulte las siguientes secciones del tema [Consulta de AWS Glue Data Catalog](#):

- Para ver metadatos de columna, como el tipo de datos, consulte [Enumeración o búsqueda de las columnas de una tabla o una vista especificadas](#).
- Para ver todas las columnas de todas las tablas en una base de datos específica en `AwsDataCatalog`, consulte [Enumeración o búsqueda de las columnas de una tabla o una vista especificadas](#).
- Para ver todas las columnas de todas las tablas en todas las bases de datos en `AwsDataCatalog`, consulte [Enumeración de todas las columnas de todas las tablas](#).
- Para ver las columnas que tienen en común tablas específicas de una base de datos, consulte [Enumeración de las columnas que tienen en común tablas específicas](#).

### Sinopsis

```
SHOW COLUMNS {FROM|IN} database_name.table_name
```

```
SHOW COLUMNS {FROM|IN} table_name [{FROM|IN} database_name]
```

Las palabras clave FROM y IN se pueden utilizar indistintamente. Si *table\_name* o *database\_name* tiene caracteres especiales como guiones, indique el nombre con acentos invertidos (por ejemplo, ``my-database``. ``my-table``). No indique *table\_name* o *database\_name* con comillas simples o dobles. Actualmente, el uso de LIKE y las expresiones de asociación de patrones no son compatibles.

## Ejemplos

Los siguientes ejemplos equivalentes muestran las columnas de la tabla `orders` en la base de datos `customers`. En los dos primeros ejemplos se presupone que `customers` es la base de datos actual.

```
SHOW COLUMNS FROM orders
```

```
SHOW COLUMNS IN orders
```

```
SHOW COLUMNS FROM customers.orders
```

```
SHOW COLUMNS IN customers.orders
```

```
SHOW COLUMNS FROM orders FROM customers
```

```
SHOW COLUMNS IN orders IN customers
```

## SHOW CREATE TABLE

Analiza una tabla existente con el nombre `table_name` para generar la consulta que la creó.

### Sinopsis

```
SHOW CREATE TABLE [db_name.]table_name
```

### Parámetros

TABLE [db\_name.]table\_name

El parámetro `db_name` es opcional. Si se omite, se usa como contexto predeterminado la base de datos actual.

#### Note

El nombre de la tabla es obligatorio.



## Ejemplos

```
SHOW CREATE TABLE orderclickstoday;
```

```
SHOW CREATE TABLE `salesdata.orderclickstoday`;
```

## Solución de problemas

Si utiliza la operación de la API [CreateTable](#) de AWS Glue o la plantilla [AWS::Glue::Table](#) de AWS CloudFormation para crear una tabla para su uso en Athena sin especificar la propiedad `TableType`, y luego ejecuta una consulta DDL como `SHOW CREATE TABLE` o `MSCK REPAIR TABLE`, puede recibir el mensaje de error `FAILED: NullPointerException Name is null (ERROR: el nombre NullPointerException es nulo)`.

Para resolver el error, especifique un valor para el atributo `TableType` [TableInput](#) como parte de la llamada a la API `CreateTable` de AWS Glue o la [plantilla AWS CloudFormation](#). Entre los valores posibles para `TableType`, se incluyen `EXTERNAL_TABLE` o `VIRTUAL_VIEW`.

Este requisito se aplica únicamente cuando se crea una tabla mediante la operación de la API `CreateTable` de AWS Glue o la plantilla `AWS::Glue::Table`. Si crea una tabla para Athena mediante una instrucción DDL o un rastreador de AWS Glue, la propiedad `TableType` se define automáticamente.

## SHOW CREATE VIEW

Muestra la instrucción SQL que crea la vista especificada.

### Sinopsis

```
SHOW CREATE VIEW view_name
```

## Ejemplos

```
SHOW CREATE VIEW orders_by_date
```

Consulte también [CREATE VIEW](#) y [DROP VIEW](#).

## SHOW DATABASES

Enumera todas las bases de datos definidas en el metaalmacén. Puede utilizar DATABASES o SCHEMAS. Significan lo mismo.

El equivalente programático de SHOW DATABASES es la acción de la API de Athena [ListDatabases](#). El método equivalente en AWS SDK for Python (Boto3) es [list\\_databases](#).

### Sinopsis

```
SHOW {DATABASES | SCHEMAS} [LIKE 'regular_expression']
```

### Parámetros

[LIKE '*regular\_expression*']

Filtra la lista de bases de datos para obtener las bases de datos que coinciden con la *regular\_expression* que especifica. Para la coincidencia de caracteres comodín, puede usar la combinación `.*`, que coincide con cualquier carácter cero en veces ilimitadas.

### Ejemplos

```
SHOW SCHEMAS;
```

```
SHOW DATABASES LIKE '.*analytics';
```

## SHOW PARTITIONS

Muestra todas las particiones de una tabla de Athena en orden sin clasificar.

### Sinopsis

```
SHOW PARTITIONS table_name
```

- Para mostrar las particiones en una tabla y enumerarlas en un orden específico, consulte la sección [Enumeración de las particiones de una tabla específica](#) de la página [Consulta de AWS Glue Data Catalog](#).

- Para ver el contenido de una partición, consulte la sección [Consulta de los datos](#) de la página [Particiones de datos en Athena](#).
- SHOW PARTITIONS no enumera las particiones que están proyectadas por Athena pero no están registradas en el catálogo de AWS Glue. Para obtener información sobre la proyección de particiones, consulte [Proyección de particiones con Amazon Athena](#).
- SHOW PARTITIONS enumera las particiones en metadatos, no las particiones en el sistema de archivos real. Para actualizar los metadatos después de eliminar particiones manualmente en Amazon S3, ejecute [ALTER TABLE DROP PARTITION](#).

## Ejemplos

En la siguiente consulta de ejemplo se muestran las particiones de la tabla `flight_delays_csv`, que muestra los datos de la tabla de vuelos del Departamento de Transporte de los Estados Unidos. Para obtener más información acerca de la tabla `flight_delays_csv`, consulte [LazySimpleSerDe para CSV, TSV y archivos con delimitación personalizada](#). La tabla está particionada por año.

```
SHOW PARTITIONS flight_delays_csv
```

## Resultados

```
year=2007
year=2015
year=1999
year=1993
year=1991
year=2003
year=1996
year=2014
year=2004
year=2011
...
```

En la siguiente consulta de ejemplo se muestran las particiones de la tabla `impressions`, que contiene datos de muestra de navegación web. Para obtener más información acerca de la tabla `impressions`, consulte [Particiones de datos en Athena](#). La tabla está particionada por la columna `dt` (datetime).

```
SHOW PARTITIONS impressions
```

## Resultados

```
dt=2009-04-12-16-00
dt=2009-04-13-18-15
dt=2009-04-14-00-20
dt=2009-04-12-13-00
dt=2009-04-13-02-15
dt=2009-04-14-12-05
dt=2009-04-14-06-15
dt=2009-04-12-21-15
dt=2009-04-13-22-15
...
```

### Listado de particiones en orden clasificado

Para ordenar las particiones en la lista de resultados, utilice la siguiente sintaxis SELECT en lugar de SHOW PARTITIONS.

```
SELECT * FROM database_name."table_name$partitions" ORDER BY column_name
```

La consulta a continuación muestra la lista de particiones para el ejemplo `flight_delays_csv`, pero en orden clasificado.

```
SELECT * FROM "flight_delays_csv$partitions" ORDER BY year
```

## Resultados

```
year
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
```

...

Para obtener más información, consulte la sección [Enumeración de las particiones de una tabla específica](#) de la página [Consulta de AWS Glue Data Catalog](#).

## SHOW TABLES

Enumera todas las tablas de base y vistas de una base de datos.

### Sinopsis

```
SHOW TABLES [IN database_name] ['regular_expression']
```

### Parámetros

[IN database\_name]

Especifica la `database_name` cuyas tablas se enumerarán. Si se omite, se presupondrá que es la base de datos del contexto actual.

#### Note

`SHOW TABLES` puede fallar si `database_name` usa un [carácter no admitido](#), como un guion. Como solución alternativa, intente incluir el nombre de la base de datos entre comillas hacia atrás.

['regular\_expression']

Filtra la lista de tablas para obtener las tablas que coinciden con la `regular_expression` que especifica. Para indicar cualquier carácter en las tablas `AWSDataCatalog`, puede utilizar la expresión comodín `*` o `.*`. Para las bases de datos de Apache Hive, utilice la expresión comodín `.*`. Para indicar una opción entre caracteres, utilice el carácter `|`.

### Ejemplos

Example – Mostrar todas las tablas de la base de datos **sampledb**

```
SHOW TABLES IN sampledb
```

## Results

```
alb_logs
cloudfront_logs
elb_logs
flights_2016
flights_parquet
view_2016_flights_dfw
```

Example – Mostrar los nombres de todas las tablas de **samp1edb** que incluyen la palabra “flights” (vuelos)

```
SHOW TABLES IN samp1edb '*flights*'
```

## Results

```
flights_2016
flights_parquet
view_2016_flights_dfw
```

Example – Mostrar los nombres de todas las tablas de **samp1edb** que terminan con la palabra “logs” (registros)

```
SHOW TABLES IN samp1edb '*logs'
```

## Results

```
alb_logs
cloudfront_logs
elb_logs
```

## SHOW TBLPROPERTIES

Muestra las propiedades de la tabla indicada.

### Sinopsis

```
SHOW TBLPROPERTIES table_name [('property_name')]
```

## Parámetros

[('property\_name')]

Si se especifica, solo se muestra el valor de la propiedad `property_name`.

## Ejemplos

```
SHOW TBLPROPERTIES orders;
```

```
SHOW TBLPROPERTIES orders('comment');
```

## SHOW VIEWS

Muestra todas las vistas de la base de datos especificada o de la base de datos actual si se omite el nombre de la base de datos. Utilice la cláusula opcional `LIKE` con una expresión regular para limitar la lista de nombres de vista.

Athena devuelve una lista de valores de tipo `STRING` donde cada valor es un nombre de vista.

## Sinopsis

```
SHOW VIEWS [IN database_name] [LIKE 'regular_expression']
```

## Parámetros

[IN database\_name]

Especifica el `database_name` a partir de la cual se enumerarán las vistas. Si se omite, se presupondrá que es la base de datos del contexto actual.

[LIKE 'regular\_expression']

Filtra la lista de vistas para obtener las vistas que coinciden con la `regular_expression` que especifique. Solo se puede utilizar el carácter comodín `*`, que equivale a cualquier carácter, o el comodín `|`, que indica una elección entre caracteres.

## Ejemplos

```
SHOW VIEWS;
```

```
SHOW VIEWS IN marketing_analytics LIKE 'orders*'
```

Consulte también [SHOW COLUMNS](#), [SHOW CREATE VIEW](#), [DESCRIBE VIEW](#) y [DROP VIEW](#).

## Consideraciones y limitaciones de las consultas SQL en Amazon Athena

Al ejecutar consultas en Athena, tenga en cuenta las siguientes consideraciones y limitaciones:

- Procedimientos almacenados: no se admiten los procedimientos almacenados.
- Número máximo de particiones: el número máximo de particiones que se pueden crear con las instrucciones `CREATE TABLE AS SELECT` (CTAS) es 100. Para obtener más información, consulte [CREATE TABLE AS](#). Para obtener una solución alternativa, consulte [Uso de CTAS e INSERT INTO para evitar el límite de 100 particiones](#).
- Instrucción no compatible: no se admiten las siguientes instrucciones:
  - `CREATE TABLE LIKE` no se admite.
  - `DESCRIBE INPUT` y `DESCRIBE OUTPUT` no son compatibles.
  - La instrucción `MERGE` solo se admite para formatos de tablas transaccionales. Para obtener más información, consulte [MERGE INTO](#).
  - No se admiten las instrucciones `UPDATE`.
- Conectores de Trino y Presto: no se admiten los conectores de [Trino](#) ni de [Presto](#). Utilice la consulta federada de Amazon Athena para conectar orígenes de datos. Para obtener más información, consulte [Uso de consulta federada de Amazon Athena](#).
- Tiempos de espera en tablas con muchas particiones: Athena puede agotar el tiempo de espera al consultar una tabla que tiene muchos miles de particiones. Esto puede ocurrir cuando la tabla tiene muchas particiones que no son de tipo `string`. Cuando utiliza el tipo `string`, Athena elimina las particiones en el nivel de metaalmacén. Sin embargo, cuando utiliza otros tipos de datos, Athena elimina las particiones en el lado del servidor. Cuantas más particiones tenga, más tardará este proceso y más probable será que se agote el tiempo de espera de las consultas. Para resolver este problema, establezca el tipo de partición en `string` de modo que Athena elimine las particiones en el nivel de metaalmacén. Esto reduce la sobrecarga y evita que se agote el tiempo de espera de las consultas.
- Compatibilidad con S3 Glacier: para obtener información sobre cómo consultar objetos restaurados de Amazon S3 Glacier, consulte [Consulta de objetos de Amazon S3 Glacier restaurados](#).
- Archivos tratados como ocultos: Athena trata los archivos de origen que comienzan con un guion bajo (`_`) o un punto (`.`) como ocultos. Para evitar esta limitación, cambie el nombre de los archivos.



- Limitación de tamaño de fila o columna: el tamaño de una sola fila o sus columnas no puede superar los 32 megabytes. Este límite se puede superar cuando, por ejemplo, una fila en un archivo CSV o JSON contiene una sola columna de 300 megabytes. Exceder este límite también puede generar el mensaje de error Línea demasiado larga en el archivo de texto. Para evitar esta limitación, asegúrese de que la suma de los datos de las columnas de cualquier fila sea inferior a 32 MB.
- Máximo de la cláusula LIMIT: el número máximo de filas que se puede especificar para la cláusula LIMIT es

9223372036854775807. Cuando se utiliza ORDER BY, el número máximo de filas admitidas para la cláusula LIMIT es 2147483647. Si se excede este límite, se produce un mensaje de error NOT\_SUPPORTED: ORDER BY LIMIT > 2 147 483 647 no es compatible.

- esquema\_information: hacer una consulta a information\_schema es más eficaz si tiene una cantidad de pequeña a moderada de metadatos de AWS Glue. Si tiene una gran cantidad de metadatos, pueden producirse errores. Para obtener información sobre la consulta de la base de datos de information\_schema para metadatos de AWS Glue, consulte el [Consulta de AWS Glue Data Catalog](#).
- Inicializaciones de matriz: debido a una limitación en Java, no es posible inicializar una matriz en Athena que tenga más de 254 argumentos.
- Columnas de metadatos ocultas: las columnas de metadatos ocultas de Hive o Iceberg \$bucket, \$file\_modified\_time, \$file\_size y \$partition no son compatibles con las vistas. Para obtener información sobre el uso de la columna de metadatos \$path en Athena, consulte [Obtención de las ubicaciones de archivos para los datos de origen en Simple Storage Service \(Amazon S3\)](#).

Para obtener información sobre la longitud máxima de la cadena de consulta, las cuotas para los tiempos de espera de las consultas y las cuotas para el número activo de consultas DML, consulte [Service Quotas](#).

## Solución de problemas en Athena

El equipo de Athena ha recopilado la siguiente información de solución de problemas de clientes. Si bien no es exhaustiva, incluye recomendaciones sobre algunos problemas comunes de rendimiento, tiempo de espera y falta de memoria.

### Temas

- [CREATE TABLE AS SELECT \(CTAS\)](#)
- [Problemas de archivos de datos](#)
- [Tablas de Linux Foundation Delta Lake](#)
- [Consultas federadas](#)
- [Errores relacionados con JSON](#)
- [MSCK REPAIR TABLE](#)
- [Problemas de salida](#)
- [Problemas de Parquet](#)
- [Problemas de particiones](#)
- [Permisos](#)
- [Problemas con la sintaxis de las consultas](#)
- [Problemas con el tiempo de espera](#)
- [Problemas de limitación](#)
- [Vistas](#)
- [Grupos de trabajo](#)
- [Recursos adicionales de](#)
- [Catálogo de errores de Athena](#)

## CREATE TABLE AS SELECT (CTAS)

Los datos duplicados se producen con instrucciones CTAS simultáneas

Athena no mantiene la validación simultánea para CTAS. Asegúrese de que no haya una instrucción CTAS duplicada para la misma ubicación al mismo tiempo. Incluso si una instrucción CTAS o INSERT INTO falla, los datos huérfanos se pueden dejar en la ubicación de datos especificada en la instrucción.

## HIVE\_TOO\_MANY\_OPEN\_PARTITIONS

Cuando utiliza una instrucción CTAS para crear una tabla con más de 100 particiones, es posible que reciba el error HIVE\_TOO\_MANY\_OPEN\_PARTITIONS: Exceeded limit of 100 open writers for partitions/buckets (HIVE\_TOO\_MANY\_OPEN\_PARTITIONS: Se ha superado el límite de 100 autores abiertos para particiones/buckets). Para evitar esta limitación, puede utilizar una instrucción CTAS y una serie de instrucciones INSERT INTO que crean o insertan hasta 100 particiones cada

una. Para obtener más información, consulte [Uso de CTAS e INSERT INTO para evitar el límite de 100 particiones](#).

## Problemas de archivos de datos

### Athena no puede leer archivos ocultos

Athena trata los archivos de origen que comienzan con un guion bajo (\_) o un punto (.) como ocultos. Para evitar esta limitación, cambie el nombre de los archivos.

### Athena lee los archivos que excluí del rastreador de AWS Glue

Athena no reconoce los [patrones de exclusión](#) que especifica en un rastreador de AWS Glue. Por ejemplo, si tiene un bucket de Amazon S3 que contiene tanto .csv como .json y se excluyen los archivos .json desde el rastreador, Athena consulta ambos grupos de archivos. Para evitar esto, coloque los archivos que desea excluir en una ubicación diferente.

### HIVE\_BAD\_DATA: Error parsing field value

Este problema se puede producir en el siguiente escenario:

- El tipo de datos definido en la tabla no coincide con los datos de origen, o un solo campo contiene diferentes tipos de datos. Para obtener resoluciones sugeridas, consulte [My Amazon Athena query fails with the error "HIVE\\_BAD\\_DATA: Error parsing field value for field x: For input string: "12312845691" \(Mi consulta de Amazon Athena devuelve el error "HIVE\\_BAD\\_DATA: error al momento de analizar el valor del campo x: para la cadena de entrada: "12312845691"\) en el Centro de conocimientos de AWS.](#)
- Los valores nulos están presentes en un campo entero. Una solución alternativa es crear la columna con los valores nulos como `string` y, a continuación, utilizar `CAST` para convertir el campo de una consulta y proporcionar un valor predeterminado de `0` para valores nulos. Para obtener más información, consulte [Cuando consulto datos CSV en Athena, obtengo el error "HIVE\\_BAD\\_DATA: error al momento de analizar el valor " del campo x: para la cadena de entrada" en el Centro de conocimientos de AWS.](#)

### HIVE\_CANNOT\_OPEN\_SPLIT: se produjo un error al abrir la partición de Hive s3://DOC-EXAMPLE-BUCKET

Este error se puede producir cuando se consulta un prefijo de bucket de Amazon S3 que tiene un gran número de objetos. Para obtener más información, consulte [¿Cómo se resuelve el](#)

[error “HIVE\\_CANNOT\\_OPEN\\_SPLIT: Se produjo un error al abrir la partición de Hive s3://DOC-EXAMPLE-BUCKET/: Ralentización” en Athena?](#) en el Centro de conocimiento de AWS.

**HIVE\_CURSOR\_ERROR:** com.amazonaws.services.s3.model.AmazonS3Exception:  
The specified key does not exist

Este error se suele producir cuando se elimina un archivo cuando se está ejecutando una consulta. Vuelva a ejecutar la consulta o verifique el flujo de trabajo para ver si otro trabajo o proceso está modificando los archivos cuando se está ejecutando la consulta.

**HIVE\_CURSOR\_ERROR:** Unexpected end of input stream

Este mensaje indica que el archivo está dañado o vacío. Verifique la integridad del archivo y vuelva a ejecutar la consulta.

**HIVE\_FILESYSTEM\_ERROR:** Incorrect fileSize **1234567** for file

Este mensaje puede aparecer cuando un archivo ha cambiado entre la planificación y la ejecución de consultas. Normalmente se produce cuando un archivo de Amazon S3 se reemplaza in situ (por ejemplo, un PUT se realiza en una clave en la que ya existe un objeto). Athena no admite eliminar o reemplazar el contenido de un archivo en el momento de ejecución de una consulta. Para evitar este error, programe trabajos que sobrescriban o eliminen archivos en momentos en que las consultas no se ejecutan, o solo escriben datos en nuevos archivos o particiones.

**HIVE\_UNKNOWN\_ERROR:** Unable to create input format

Este error puede ser el resultado de problemas como los que se enumeran a continuación:

- El rastreador de AWS Glue no pudo clasificar el formato de datos
- Ciertas propiedades de definición de tablas de AWS Glue están vacías
- Athena no admite el formato de datos de los archivos de Amazon S3

Para obtener más información, consulte [¿Cómo resuelvo el error “unable to create input format” en Athena?](#) en el Centro de Conocimientos de AWS o vea el [video](#) del Centro de conocimientos.

La ubicación de S3 proporcionada para guardar los resultados de la consulta no es válida.

Asegúrese de que ha especificado una ubicación de S3 válida para los resultados de la consulta. Para obtener más información, consulte [Especificación de una ubicación de resultados de consulta](#) en el tema [Trabajo con resultados de las consultas, consultas recientes y archivos de salida](#).

## Tablas de Linux Foundation Delta Lake

El esquema de la tabla de Delta Lake no está sincronizado

Cuando se consulta una tabla de Delta Lake que tiene un esquema en AWS Glue desactualizado, se puede recibir el siguiente mensaje de error:

```
INVALID_GLUE_SCHEMA: Delta Lake table schema in Glue does not match the most recent schema of the Delta Lake transaction log. Please ensure that you have the correct schema defined in Glue.
```

El esquema puede quedar desactualizado si se modifica AWS Glue después de añadirlo a Athena. Para actualizar el esquema, siga uno de los pasos a continuación:

- En AWS Glue, ejecute el [rastreador AWS Glue](#).
- En Athena, [suelte la tabla](#) y [créela](#) de nuevo.
- Agregue las columnas que faltan manualmente, ya sea mediante la instrucción [ALTER TABLE ADD COLUMNS](#) de Athena o [editando el esquema de la tabla en AWS Glue](#).

## Consultas federadas

Se agotó el tiempo de espera al llamar a ListTableMetadata

El tiempo de espera de una llamada a la API [ListTableMetadata](#) puede agotarse si hay muchas tablas en el origen de datos, si el origen de datos es lento, o si la red es lenta. Para solucionar este problema, siga estos pasos.

- Comprobar la cantidad de tablas: si tiene más de 1000 tablas, intente reducir el número. Para obtener la respuesta ListTableMetadata más rápida, recomendamos tener menos de 1000 tablas por catálogo.

- Comprobar la configuración de Lambda: la supervisión del comportamiento de la función de Lambda es fundamental. Cuando utilice catálogos federados, asegúrese de examinar los registros de ejecución de la función de Lambda. En función de los resultados, ajuste los valores de memoria y tiempo de espera en consecuencia. Para identificar cualquier posible problema relacionado con los tiempos de espera, revise su configuración de Lambda. Para obtener más información, consulte [Configuración del tiempo de espera de la función \(consola\)](#) en la Guía para desarrolladores de AWS Lambda.
- Comprobar los registros de los orígenes de datos federados: examine los registros y los mensajes de error del origen de datos federado para ver si hay algún problema o error. Los registros pueden proporcionar información valiosa sobre la causa del tiempo de espera.
- Utilice **StartQueryExecution** para recuperar metadatos: si tiene más de 1000 tablas, recuperar los metadatos con el conector federado puede tardar más de lo esperado. Dado que la naturaleza asíncrona de [StartQueryExecution](#) garantiza que Athena ejecute la consulta de la forma más óptima, considere utilizar `StartQueryExecution` como alternativa a `ListTableMetadata`. Los siguientes ejemplos de AWS CLI muestran cómo se puede utilizar `StartQueryExecution` en lugar de `ListTableMetadata` para obtener todos los metadatos de las tablas del catálogo de datos.

En primer lugar, ejecute una consulta que obtenga todas las tablas, como en el ejemplo siguiente.

```
aws athena start-query-execution --region us-east-1 \  
--query-string "SELECT table_name FROM information_schema.tables LIMIT 50" \  
--work-group "your-work-group-name"
```

A continuación, recupere los metadatos de una tabla individual, como en el siguiente ejemplo.

```
aws athena start-query-execution --region us-east-1 \  
--query-string "SELECT * FROM information_schema.columns \  
WHERE table_name = 'your-table-name' AND \  
table_catalog = 'your-catalog-name'" \  
--work-group "your-work-group-name"
```

El tiempo necesario para obtener los resultados depende del número de tablas del catálogo.

Para obtener más información sobre cómo solucionar problemas relacionados con las consultas federadas, consulte [Common Problems](#) en la sección `awslabs/aws-athena-query-federation` de GitHub o la documentación sobre los [conectores de orígenes de datos de Athena](#) individuales.

## Errores relacionados con JSON

### Errores de datos NULL o incorrectos al intentar leer datos JSON

Los errores de datos NULL o incorrectos al intentar leer datos JSON pueden deberse a una serie de causas. Para identificar las líneas que están generando errores al utilizar OpenX SerDe, establezca `ignore.malformed.json` en `true`. Los registros con formato incorrecto se devolverán como NULL. Para obtener más información, consulte [Recibo errores cuando intento leer datos JSON en Amazon Athena](#) en el Centro de conocimientos de AWS o vea el [video](#) del Centro de conocimientos.

**HIVE\_BAD\_DATA: Error parsing field value for field 0: java.lang.String cannot be cast to org.openx.data.jsonserde.json.JSONObject**

[El SerDe JSON de OpenX](#) arroja este error cuando no puede analizar una columna en una consulta de Athena. Esto puede suceder si define una columna como `map` o `struct`, pero los datos subyacentes son en realidad `string`, `int` u otro tipo primitivo.

**HIVE\_CURSOR\_ERROR: Row is not a valid JSON object - JSONException: Duplicate key (HIVE\_CURSOR\_ERROR: la fila no es un objeto JSON válido - JSONException: clave duplicada)**

Este error se produce cuando utiliza Athena para consultar recursos AWS Config que tienen varias etiquetas con el mismo nombre en diferentes casos. La solución consiste en ejecutar `CREATE TABLE` con `WITH SERDEPROPERTIES 'case.insensitive'='false'` y asignar los nombres. Para obtener más información acerca de `case.insensitive` y el mapeo, consulte [Bibliotecas de SerDe JSON](#). Para obtener más información, consulte [¿Cómo resuelvo el error “HIVE\\_CURSOR\\_ERROR: Row is not a valid JSON object - JSONException: Duplicate key” \(HIVE\\_CURSOR\\_ERROR: la fila no es un objeto JSON válido - JSONException: clave duplicada\) al leer archivos de AWS Config en Athena?](#) en el centro de conocimientos de AWS.

### Mensajes HIVE\_CURSOR\_ERROR con formato de impresión JSON

Las bibliotecas [El SerDe JSON de Hive](#) y [El SerDe JSON de OpenX](#) esperan que cada documento JSON esté en una sola línea de texto sin caracteres de terminación de línea que separen los campos del registro. Si el texto JSON está en formato de impresión, puede recibir un mensaje de error como `HIVE_CURSOR_ERROR: Row is not a valid JSON Object (HIVE_CURSOR_ERROR: la fila no es un objeto JSON válido)` o `HIVE_CURSOR_ERROR: JsonParseException: Unexpected end-of-input: expected close marker for OBJECT (HIVE_CURSOR_ERROR: JsonParseException: fin de entrada inesperado: marcador de cierre esperado para OBJECT)` cuando intenta consultar la tabla después

de crearla. Para obtener más información, consulte [Archivos de datos JSON](#) en la documentación de OpenX SerDe en GitHub.

## Varios registros JSON devuelven un SELECT COUNT de 1

Si utiliza [El SerDe JSON de OpenX](#), asegúrese de que los registros estén separados por un carácter de nueva línea. Para obtener más información, consulte [La consulta SELECT COUNT en Amazon Athena devuelve solo un registro aunque el archivo JSON de entrada tenga varios registros](#) en el Centro de conocimientos de AWS.

## No se puede consultar una tabla creada por un rastreador de AWS Glue que utiliza un clasificador JSON personalizado

El motor Athena no es compatible con [clasificadores JSON personalizados](#). Para evitar este problema, cree una tabla nueva sin el clasificador personalizado. Para transformar el JSON, puede utilizar CTAS o crear una vista. Por ejemplo, si está trabajando con matrices, puede utilizar la opción UNNEST para aplanar el JSON. Otra opción consiste en utilizar un trabajo ETL de AWS Glue que admita el clasificador personalizado, convertir los datos a Parquet en Amazon S3 y, a continuación, consultarlo en Athena.

## MSCK REPAIR TABLE

Para obtener información acerca de los problemas relacionados con MSCK REPAIR TABLE, consulte las secciones [Consideraciones y limitaciones](#) y [Resolución de problemas](#) de la página [MSCK REPAIR TABLE](#).

## Problemas de salida

### No se puede verificar/crear el bucket de salida

Este error se puede producir si la ubicación del resultado de la consulta especificada no existe o si los permisos adecuados no están presentes. Para obtener más información, consulte [¿Cómo resuelvo el error “No se puede verificar/crear el bucket de salida” en Amazon Athena?](#) en el Centro de conocimientos de AWS.

### El resultado TIMESTAMP está vacío

Athena requiere el formato TIMESTAMP de Java. Para obtener más información, consulte [Cuando consulto una tabla en Amazon Athena, el resultado TIMESTAMP está vacío](#) en el Centro de conocimientos de AWS.



## Almacenar la salida de la consulta de Athena en un formato distinto de CSV

De forma predeterminada, Athena genera archivos únicamente en formato CSV. Para generar los resultados de una consulta SELECT en un formato diferente, puede utilizar la instrucción UNLOAD. Para obtener más información, consulte [UNLOAD](#). También puede utilizar una consulta de CTAS que use la [propiedad de la tabla](#) `format` para configurar el formato de salida. A diferencia de UNLOAD, la técnica de CTAS requiere la creación de una tabla. Para obtener más información, consulte [¿Cómo puedo almacenar una salida de consulta Athena en un formato distinto a CSV, como un formato comprimido?](#) en el Centro de conocimientos de AWS.

La ubicación de S3 proporcionada para guardar los resultados de la consulta no es válida

Es posible que reciba este mensaje de error si la ubicación del bucket de salida no está en la misma región que la región en la que ejecuta la consulta. Para evitar esto, especifique una ubicación de resultados de consulta en la región en la que ejecuta la consulta. Para ver los pasos, consulte [Especificación de una ubicación de resultados de consulta](#).

## Problemas de Parquet

```
org.apache.parquet.io.GroupColumnIO cannot be cast to
org.apache.parquet.io.PrimitiveColumnIO
```

Este error se debe a una falta de coincidencia de esquema de Parquet. Una columna que tiene un tipo no primitivo (por ejemplo, `array`) se ha declarado como un tipo primitivo (por ejemplo, `string`) en AWS Glue. Para solucionar este problema, verifique el esquema de datos en los archivos y compárelo con el esquema declarado en AWS Glue.

## Problemas con las estadísticas de Parquet

Al leer los datos de Parquet, es posible que reciba mensajes de error como los siguientes:

```
HIVE_CANNOT_OPEN_SPLIT: Index x out of bounds for length y
HIVE_CURSOR_ERROR: Failed to read x bytes
HIVE_CURSOR_ERROR: FailureException at Malformed input: offset=x
HIVE_CURSOR_ERROR: FailureException at java.io.IOException:
can not read class org.apache.parquet.format.PageHeader: Socket is closed by peer.
```

Para solucionar este problema, utilice la instrucción [ALTER TABLE SET TBLPROPERTIES](#) o [CREATE TABLE](#) para establecer la propiedad `parquet.ignore.statistics` de Parquet SerDe en `true`, como en los siguientes ejemplos.

### Ejemplo de CREATE TABLE

```
...  
ROW FORMAT SERDE  
'org.apache.hadoop.hive.q1.io.parquet.serde.ParquetHiveSerDe'  
WITH SERDEPROPERTIES ('parquet.ignore.statistics'='true')  
STORED AS PARQUET  
...
```

### Ejemplo de ALTER TABLE

```
ALTER TABLE ... SET TBLPROPERTIES ('parquet.ignore.statistics'='true')
```

Para obtener más información sobre Parquet Hive SerDe, consulte [El SerDe de Parquet](#).

## Problemas de particiones

### MSCK REPAIR TABLE no elimina las particiones obsoletas

Si elimina una partición de forma manual en Amazon S3 y, a continuación, ejecuta `MSCK REPAIR TABLE`, puede recibir el mensaje de error `Partitions missing from filesystem` (Faltan particiones en el sistema de archivos). Esto sucede porque `MSCK REPAIR TABLE` no elimina particiones obsoletas de los metadatos de la tabla. Utilice [ALTER TABLE DROP PARTITION](#) para eliminar las particiones obsoletas de forma manual. Para obtener más información, consulte la sección “Solución de problemas” del tema [MSCK REPAIR TABLE](#).

### Error MSCK REPAIR TABLE

Cuando una gran cantidad de particiones (por ejemplo, más de 100 000) están asociadas a una tabla determinada, `MSCK REPAIR TABLE` puede fallar debido a limitaciones de la memoria. Para evitar este límite, utilice [ALTER TABLE ADD PARTITION](#) en su lugar.

### MSCK REPAIR TABLE detecta particiones pero no las agrega a AWS Glue

Este problema se puede producir si una ruta de Amazon S3 tiene una combinación de mayúsculas y minúsculas en lugar de minúsculas o si una política de IAM no permite la acción

`glue:BatchCreatePartition`. Para obtener más información, consulte [MSCK REPAIR TABLE detecta particiones en Athena pero no las agrega a AWS Glue Data Catalog](#) en el Centro de conocimientos de AWS.

Los intervalos de proyección de particiones con el formato de fecha de `dd-MM-yyyy-HH-mm-ss` o `yyyy-MM-dd` no funcionan

Para que funcione correctamente, se debe establecer el formato de fecha en `yyyy-MM-dd HH:00:00`. Para obtener más información, consulte la publicación de Stack Overflow [Athena partition projection not working as expected](#) (La proyección de particiones de Athena no funciona de la manera esperada).

`PARTITION BY` no admite el tipo `BIGINT`

Convierta el tipo de datos a `string` y vuelva a intentarlo.

No hay particiones significativas disponibles

Por lo general, este mensaje de error significa que la configuración de la partición está dañada. Para resolver este problema, elimine la tabla y cree una tabla con nuevas particiones.

Proyección de particiones no funciona junto con particiones de intervalo

Verifique que la unidad de intervalo de tiempo `projection.<columnName>.interval.unit` coincide con el delimitador de las particiones. Por ejemplo, si las particiones están delimitadas por días, una unidad de rango de horas no funcionará.

Error de proyección de particiones cuando el intervalo se especifica con un guion

Si se especifica la propiedad de la tabla `range` con un guion en lugar de una coma, se produce un error similar al de `INVALID_TABLE_PROPERTY`: para la cadena de entrada: `"number-number"`. Asegúrese de que los valores del intervalo se encuentren separados por una coma, no por un guion. Para obtener más información, consulte [Tipo entero](#).

`HIVE_UNKNOWN_ERROR: Unable to create input format`

Una o más de las particiones de Glue se declaran en un formato diferente, ya que cada partición de Glue tiene su propio formato de entrada específico de forma independiente. Verifique cómo se definen las particiones en AWS Glue.

## HIVE\_PARTITION\_SCHEMA\_MISMATCH

Si el esquema de una partición es diferente al esquema de la tabla, es posible que una consulta falle con el mensaje de error HIVE\_PARTITION\_SCHEMA\_MISMATCH. Para obtener más información, consulte [Sincronización de un esquema de partición para evitar "HIVE\\_PARTITION\\_SCHEMA\\_MISMATCH"](#).

La tabla SemanticException no está particionada, pero la especificación de partición existe

Este error se puede producir cuando no se definieron particiones en la instrucción CREATE TABLE. Para obtener más información, consulte [¿Cómo puedo solucionar el error "FAILED: la tabla SemanticException no está particionada, pero la especificación de partición existe" en Athena?](#) en el Centro de conocimientos de AWS.

### Archivos `_$folder$` sin bytes

Si ejecuta una declaración ALTER TABLE ADD PARTITION y especifica erróneamente una partición que ya existe y una ubicación de Amazon S3 incorrecta, se crean archivos de marcador de posición de cero bytes del formato `partition_value_$folder$` en Amazon S3. Debe eliminar esos archivos en forma manual.

Para evitar que esto ocurra, utilice la sintaxis ADD IF NOT EXISTS de su declaración ALTER TABLE ADD PARTITION, así:

```
ALTER TABLE table_name ADD IF NOT EXISTS PARTITION [...]
```

### Cero registros devueltos de datos particionados

Este problema se puede producir por diversas razones. Para conocer las posibles causas y resoluciones, consulte [Creé una tabla en Amazon Athena con particiones definidas, pero cuando consulto la tabla, se devuelven cero registros](#) en el Centro de conocimientos de AWS.

Véase también [HIVE\\_TOO\\_MANY\\_OPEN\\_PARTITIONS](#).

## Permisos

### Error de acceso denegado al momento de consultar Simple Storage Service (Amazon S3)

Esto puede suceder cuando no tiene permiso para leer los datos del bucket, permiso para escribir en el bucket de resultados, o si la ruta de Amazon S3 contiene un punto de conexión de región como `us-east-1.amazonaws.com`. Para obtener más información, consulte [Cuando ejecuto una consulta de Athena, aparece un error de “acceso denegado”](#) en el Centro de conocimientos de AWS.

### Acceso denegado con código de estado: error 403 al momento de ejecutar consultas DDL en datos cifrados en Amazon S3

Es posible que reciba el mensaje de error Acceso denegado (Servicio: Amazon S3; código de estado: 403; código de error: AccessDenied; ID de solicitud: `<request_id>`) si se cumplen las siguientes condiciones:

1. Ejecuta una consulta DDL como `ALTER TABLE ADD PARTITION` o `MSCK REPAIR TABLE`.
2. Tiene un bucket con [cifrado predeterminado](#) configurado para usar SSE-S3.
3. El bucket también tiene una política de bucket como la siguiente, que obliga a las solicitudes `PutObject` a especificar los encabezados `PUT "s3:x-amz-server-side-encryption": "true"` y `"s3:x-amz-server-side-encryption": "AES256"`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::<resource-name>/*",
      "Condition": {
        "Null": {
          "s3:x-amz-server-side-encryption": "true"
        }
      }
    },
    {
      "Effect": "Deny",
      "Principal": "*",
```

```
"Action": "s3:PutObject",
"Resource": "arn:aws:s3:::<resource-name>/*",
"Condition": {
  "StringNotEquals": {
    "s3:x-amz-server-side-encryption": "AES256"
  }
}
]
```

En un caso como este, la solución recomendada es eliminar la política de bucket como la anterior, dado que el cifrado predeterminado del bucket ya está presente.

## Acceso denegado con código de estado: 403 al momento de consultar un bucket de Amazon S3 en otra cuenta

Este error se puede producir al intentar consultar los registros escritos por otro Servicio de AWS y la segunda cuenta es el propietario del bucket, pero no posee los objetos del bucket. Para obtener más información, consulte [Recibo la excepción de Amazon S3 “acceso denegado con código de estado: 403” en Amazon Athena cuando consulto un bucket de otra cuenta](#) en el Centro de conocimientos de AWS o vea el [video](#) del Centro de conocimientos.

## Utilice credenciales de rol de IAM para conectarse al controlador JDBC de Athena

Puede recuperar las credenciales temporales de un rol para autenticar la [conexión JDBC a Athena](#). Las credenciales temporales tienen una vida útil máxima de 12 horas. Para obtener más información, consulte [¿Cómo puedo usar mis credenciales de rol de IAM o cambiar a otro rol de IAM cuando me conecto a Athena usando el controlador JDBC?](#) en el Centro de conocimientos de AWS.

## Problemas con la sintaxis de las consultas

FAILED: NullPointerException name is null (ERROR: el nombre NullPointerException es nulo)

Si utiliza la operación de la API [CreateTable](#) de AWS Glue o la plantilla [AWS::Glue::Table](#) de AWS CloudFormation para crear una tabla para su uso en Athena sin especificar la propiedad `TableType`, y luego ejecuta una consulta DDL como `SHOW CREATE TABLE` o `MSCK REPAIR`

TABLE, puede recibir el mensaje de error FAILED: NullPointerException Name is null (ERROR: el nombre NullPointerException es nulo).

Para resolver el error, especifique un valor para el atributo `TableType` [TableInput](#) como parte de la llamada a la API `CreateTable` de AWS Glue o la [plantilla AWS CloudFormation](#). Entre los valores posibles para `TableType`, se incluyen `EXTERNAL_TABLE` o `VIRTUAL_VIEW`.

Este requisito se aplica únicamente cuando se crea una tabla mediante la operación de la API `CreateTable` de AWS Glue o la plantilla `AWS::Glue::Table`. Si crea una tabla para Athena mediante una instrucción DDL o un rastreador de AWS Glue, la propiedad `TableType` se define automáticamente.

## Función no registrada

Este error se produce al intentar utilizar una función que no es compatible con Athena. Para obtener una lista de las funciones compatibles con Athena, consulte [Funciones en Amazon Athena](#) o ejecute la instrucción `SHOW FUNCTIONS` en el Editor de consultas. También puede escribir su propia [función definida por el usuario \(UDF, user defined function\)](#). Para obtener más información, consulte [¿Cómo resuelvo el error de sintaxis "función no registrada" en Athena?](#) en el Centro de conocimientos de AWS.

## Excepciones GENERIC\_INTERNAL\_ERROR

Las excepciones `GENERIC_INTERNAL_ERROR` pueden tener diversas causas, entre otras, las siguientes:

- `GENERIC_INTERNAL_ERROR: Null`: puede ver esta excepción en cualquiera de las siguientes condiciones:
  - Hay una discrepancia de esquema entre el tipo de datos de una columna de la definición de tabla y el tipo de datos real del conjunto de datos.
  - Ejecuta una consulta (CTAS) `CREATE TABLE AS SELECT` con sintaxis inexacta.
- `GENERIC_INTERNAL_ERROR: Parent builder is null (GENERIC_INTERNAL_ERROR: el generador principal es nulo)`: es posible que aparezca esta excepción cuando consulte una tabla con columnas de tipo de datos `array` y esté utilizando la biblioteca `OpenCSVSerDe`. El formato `OpenCSVSerDe` no es compatible con el tipo de datos `array`.
- `GENERIC_INTERNAL_ERROR: el valor supera MAX_INT`: puede aparecer esta excepción cuando la columna de datos de origen se define con el tipo de datos `INT` y tiene un valor numérico superior a 2.147.483.647.

- **GENERIC\_INTERNAL\_ERROR**: el valor supera **MAX\_BYTE**: puede aparecer esta excepción cuando la columna de datos de origen tiene un valor numérico superior al tamaño permitido para el tipo de datos **BYTE**. El tipo de datos **BYTE** es equivalente a **TINYINT**. **TINYINT** es un número entero firmado de 8 bits en formato de complemento de dos, con un valor mínimo de -128 y un valor máximo de 127.
- **GENERIC\_INTERNAL\_ERROR**: el número de valores de partición no coincide con el número de filtros: puede aparecer esta excepción si tiene particiones incoherentes en los datos de Amazon Simple Storage Service (Amazon S3). Es posible que haya particiones incoherentes en cualquiera de las siguientes condiciones:
  - Las particiones de Amazon S3 han cambiado (ejemplo: se han agregado nuevas particiones).
  - El número de columnas de partición de la tabla no coincide con el de los metadatos de la partición.

Para obtener información más detallada acerca de cada uno de estos errores, consulte [¿Cómo resuelvo el error “GENERIC\\_INTERNAL\\_ERROR” cuando consulto una tabla en Amazon Athena?](#) en el Centro de conocimientos de AWS.

## El número de grupos coincidentes no concuerda con el número de columnas

Este error se produce al utilizar [SerDe de Regex](#) en una instrucción **CREATE TABLE** y cuando el número de grupos coincidentes de expresiones regulares no concuerda con el número de columnas especificado para la tabla. Para obtener más información, consulte [¿Cómo resuelvo el error de RegexSerDe “el número de grupos coincidentes no concuerda con el número de columnas” en Amazon Athena?](#) en el Centro de conocimientos de AWS.

## QueryString no cumplió con la restricción: la longitud debe ser menor o igual a 262 144

La longitud máxima de la cadena de consultas en Athena (262 144 bytes) no es una cuota ajustable. AWS Support no puede aumentar la cuota, pero puede evitar este problema al dividir las consultas largas en otras más pequeñas. Para obtener más información, consulte [¿Cómo puedo aumentar la longitud máxima de la cadena de consultas en Athena?](#) en el Centro de conocimientos de AWS.

## SYNTAX\_ERROR: No se puede resolver la columna

Este error se puede producir al consultar una tabla creada por un rastreador de AWS Glue desde un archivo CSV codificado UTF-8 que tiene una marca de orden de bytes (BOM, marca de orden de bytes). AWS Glue no reconoce las BOM y las cambia por signos de interrogación, que Amazon



Athena no reconoce. La solución consiste en eliminar el signo de interrogación en Athena o en AWS Glue.

## Hay demasiados argumentos para llamar a la función

En la versión 3 del motor de Athena, las funciones no pueden aceptar más de 127 argumentos. Esta limitación se estable por diseño. Si utiliza una función con más de 127 parámetros, aparece un mensaje de error como el siguiente:

TOO\_MANY\_ARGUMENTS: línea *nnn:nn*: hay demasiados argumentos para llamar a la función *function\_name()*.

Para resolver este problema, utilice menos parámetros por llamada a la función.

## Problemas con el tiempo de espera

Si se producen errores de tiempo de espera en las consultas de Athena, compruebe los registros de CloudTrail. Las consultas pueden agotarse debido a la limitación de AWS Glue o a las API de Lake Formation. Cuando se producen estos errores, los mensajes de error correspondientes pueden indicar un problema de tiempo de espera de la consulta en lugar de un problema de limitación. Para solucionar el problema, puede comprobar los registros de CloudTrail antes de ponerse en contacto con AWS Support. Para obtener más información, consulte [Consulta de registros de AWS CloudTrail](#) y [Registro de las llamadas a la API de Amazon Athena con AWS CloudTrail](#).

Para obtener información sobre los problemas de tiempo de espera de las consultas federadas al llamar a la API `ListTableMetadata`, consulte [Se agotó el tiempo de espera al llamar a ListTableMetadata](#).

## Problemas de limitación

Si las consultas superan los límites de los servicios dependientes, como Amazon S3, AWS KMS, AWS Glue o AWS Lambda, se pueden esperar los siguientes mensajes. Para resolver estos problemas, reduzca el número de llamadas simultáneas que se originan desde la misma cuenta.

Servicio	Mensaje de error
AWS Glue	AWSGlueException: Rate exceeded. (Se superó el índice).
AWS KMS	Ha superado la tasa a la que puede llamar a KMS. Reduzca la frecuencia de las llamadas.

Servicio	Mensaje de error
AWS Lambda	Rate Exceeded (Se superó el límite) TooManyRequestsException (Excepción de demasiadas solicitudes)
Amazon S3	AmazonS3Exception: Please reduce your request rate (AmazonS3Exception: Reduzca el índice de solicitudes).

Para obtener información sobre las formas de evitar la limitación de Amazon S3 cuando utiliza Athena, consulte [Evitar la limitación de Amazon S3](#).

## Vistas

### Las vistas creadas en el shell de Apache Hive no funcionan en Athena

Debido a las implementaciones completamente diferentes, las vistas creadas en el shell de Apache Hive no son compatibles con Athena. Para resolver este problema, vuelva a crear las vistas en Athena.

### La vista es obsoleta; debe volver a crearse

Puede recibir este error si se ha modificado o eliminado la tabla subyacente de una vista. La resolución consiste en volver a crear la vista. Para obtener más información, consulte [¿Cómo puedo resolver el error “La vista es obsoleta; debe volver a crearse” en Athena?](#) en el Centro de conocimientos de AWS.

## Grupos de trabajo

Para obtener información sobre cómo solucionar problemas relacionados con grupos de trabajo, consulte [Solución de problemas de grupos de trabajo](#).

## Recursos adicionales de

En las páginas siguientes se proporciona información adicional para la solución de problemas con Amazon Athena.

- [Catálogo de errores de Athena](#)
- [Service Quotas](#)

- [Consideraciones y limitaciones de las consultas SQL en Amazon Athena](#)
- [DDL no admitido](#)
- [Nombres de tablas, bases de datos y columnas](#)
- [Tipos de datos en Amazon Athena](#)
- [Formatos de datos y SerDes compatibles](#)
- [Compatibilidad con la compresión de Athena](#)
- [Palabras clave reservadas](#)
- [Solución de problemas de grupos de trabajo](#)

También pueden ser de ayuda los siguientes recursos de AWS:

- [Temas sobre Athena en el Centro de conocimientos de AWS](#)
- [Preguntas de Amazon Athena sobre AWS re:Post](#)
- [Publicaciones sobre Athena en el Blog de macrodatos de AWS](#)

A menudo, la solución de problemas requiere consultas y descubrimiento iterativos por parte de un experto o de una comunidad de ayudantes. Si sigue teniendo problemas después de probar las sugerencias de esta página, contacte con AWS Support (en AWS Management Console, haga clic en Support (Asistencia), Support Center [Centro de asistencia]) o haga una pregunta en [AWS re:Post](#) con la etiqueta Amazon Athena.

## Catálogo de errores de Athena

Athena proporciona información de errores estandarizada para ayudar a comprender las consultas fallidas y tomar medidas después de que se produzca un error en la consulta. La característica `AthenaError` incluye un campo `ErrorCategory` y un campo `ErrorType`. `ErrorCategory` especifica si la causa de la consulta fallida se debe a un error del sistema, error de usuario u otro error. `ErrorType` proporciona información más detallada sobre el origen del fallo. Al combinar los dos campos, puede comprender mejor las circunstancias que rodean y las causas del error específico que se produjo.

### Categoría de errores

En la siguiente tabla se enumeran los valores de la categoría de errores de Athena y sus significados.

Categoría de errores	Origen
1	SYSTEM
2	USER
3	OTHER

## Referencia de tipo de error

En la siguiente tabla se enumeran los valores de tipo de error de Athena y sus significados.

Tipo de error	Descripción
0	La consulta agotó los recursos en este factor de escala
1	La consulta agotó los recursos en este factor de escala
2	La consulta agotó los recursos en este factor de escala
3	La consulta agotó los recursos en este factor de escala
4	La consulta agotó los recursos en este factor de escala
5	La consulta agotó los recursos en este factor de escala
6	La consulta agotó los recursos en este factor de escala
7	La consulta agotó los recursos en este factor de escala
8	La consulta agotó los recursos en este factor de escala
100	Error de servicio interno
200	Error interno del motor de consultas
201	Error interno del motor de consultas
202	Error interno del motor de consultas

Tipo de error	Descripción
203	Error del controlador
204	Error del metaalmacén
205	Error interno del motor de consultas
206	Tiempo de espera de la consulta agotado
207	Error interno del motor de consultas
208	Error interno del motor de consultas
209	Error al cancelar la consulta
210	Tiempo de espera de la consulta agotado
211	Error interno del motor de consultas
212	Error interno del motor de consultas
213	Error interno del motor de consultas
214	Error interno del motor de consultas
215	Error interno del motor de consultas
216	Error interno del motor de consultas
217	Error interno del motor de consultas
218	Error interno del motor de consultas
219	Error interno del motor de consultas
220	Error interno del motor de consultas
221	Error interno del motor de consultas
222	Error interno del motor de consultas

Tipo de error	Descripción
223	Error interno del motor de consultas
224	Error interno del motor de consultas
225	Error interno del motor de consultas
226	Error interno del motor de consultas
227	Error interno del motor de consultas
228	Error interno del motor de consultas
229	Error interno del motor de consultas
230	Error interno del motor de consultas
231	Error interno del motor de consultas
232	Error interno del motor de consultas
233	Error de Iceberg
234	Error de Lake Formation
235	Error interno del motor de consultas
236	Error interno del motor de consultas
237	Error de serialización
238	Error al cargar metadatos en Simple Storage Service (Amazon S3)
239	Error de persistencia general
240	Error al enviar la consulta
300	Error de servicio interno
301	Error de servicio interno

Tipo de error	Descripción
302	Error de servicio interno
303	Error de servicio interno
400	Error de servicio interno
401	Error al escribir los resultados de las consultas en Simple Storage Service (Amazon S3)
402	Error al escribir los resultados de las consultas en Simple Storage Service (Amazon S3)
1 000	Error de usuario
1001	Error de datos
1002	Error de datos
1003	Error de tarea DDL
1004	Error de esquema
1005	Error de serialización
1006	Error de sintaxis
1007	Error de datos
1008	Consulta rechazada
1009	Error de consulta
1010	Error de servicio interno
1011	Consulta cancelada por el usuario
1012	Error interno del motor de consultas
1013	Error interno del motor de consultas

Tipo de error	Descripción
1014	Consulta cancelada por el usuario
1100	Argumento no válido proporcionado
1101	Propiedad no válida proporcionada
1102	Error interno del motor de consultas
1103	Propiedad de tabla no válida proporcionada
1104	Error interno del motor de consultas
1105	Error interno del motor de consultas
1106	Argumento de función no válido proporcionado
1107	Vista no válida
1108	Error al registrar la función
1109	No se encuentra la ruta de Simple Storage Service (Amazon S3) proporcionada
1110	La tabla o vista proporcionada no existe
1200	Consulta no admitida
1201	No se admite el decodificador proporcionado
1202	Tipo de consulta no admitido
1300	Error no encontrado general
1301	Entidad general no encontrada
1302	Archivo no encontrado
1303	Función proporcionada o implementación de función no encontrada
1304	Error interno del motor de consultas



Tipo de error	Descripción
1305	Error interno del motor de consultas
1306	Bucket de Simple Storage Service (Amazon S3) no encontrado
1307	Motor seleccionado no encontrado
1308	Error interno del motor de consultas
1400	Error de limitación
1401	Error de consulta debido a limitación de AWS Glue
1402	Error de consulta debido a demasiadas versiones de tabla en AWS Glue
1403	Error de consulta debido a limitación de Simple Storage Service (Amazon S3)
1404	Error de consulta debido a limitación de Amazon Athena
1405	Error de consulta debido a limitación de Amazon Athena
1406	Error de consulta debido a limitación de Amazon Athena
1500	Error de permiso
1501	Error de permiso de Simple Storage Service (Amazon S3)
1602	Se ha superado el límite de la capacidad reservada. Capacidad insuficiente para ejecutar esta consulta.
1700	Se produjo un error en la consulta debido a una excepción interna de Lake Formation.
1701	Se produjo un error en la consulta debido a una excepción interna AWS Glue
9999	Error de servicio interno

# Ejemplos de código

Los ejemplos de este tema utilizan SDK para Java 2.x como punto de partida para escribir aplicaciones de Athena.

## Note

Para obtener información acerca de la programación de Athena usando otros SDK de AWS de lenguaje específico, consulte los siguientes recursos:

- AWS Command Line Interface ([athena](#))
- AWS SDK for .NET ([Amazon.Athena.Model](#))
- AWS SDK for C++ ([Aws::Athena::AthenaClient](#))
- AWS SDK for Go ([athena](#))
- AWS SDK for JavaScript v3 ([AthenaClient](#))
- AWS SDK for PHP 3.x ([Aws\Athena](#))
- AWS SDK for Python (Boto3) ([Athena.Client](#))
- AWS SDK for Ruby v3 ([Aws::Athena::Client](#))

Para obtener más información sobre cómo ejecutar los ejemplos de código Java en esta sección, consulte el [archivo Readme de Java de Amazon Athena](#) en el [Repositorio de ejemplos de código de AWS](#) en GitHub. Para obtener la referencia de programación en Java para Athena, consulte [AthenaClient](#) en AWS SDK for Java 2.x.

- Ejemplos de código Java
  - [Constantes](#)
  - [Crear un cliente para obtener acceso a Athena](#)
  - Uso de ejecuciones de consultas
    - [Iniciar la ejecución de una consulta](#)
    - [Detener la ejecución de una consulta](#)
    - [Generar una lista de ejecuciones de consultas](#)
  - Uso de consultas con nombre
    - [Crear una consulta con nombre](#)

- [Eliminar una consulta con nombre](#)
- [Generar una lista de ejecuciones de consultas](#)

### Note

Estas muestras utilizan constantes (como `ATHENA_SAMPLE_QUERY`) para cadenas, que se definen en una declaración de clase `ExampleConstants.java`. Reemplace estas constantes con sus propias cadenas o constantes definidas.

## Constantes

La clase `ExampleConstants.java` demuestra cómo consultar una tabla creada por el tutorial [Introducción](#) en Athena.

```
package aws.example.athena;

public class ExampleConstants {

    public static final int CLIENT_EXECUTION_TIMEOUT = 100000;
    public static final String ATHENA_OUTPUT_BUCKET = "s3://bucketscott2"; // change
the Amazon S3 bucket name to match                                     // your
environment
    // Demonstrates how to query a table with a comma-separated value (CSV) table.
    // For information, see
    // https://docs.aws.amazon.com/athena/latest/ug/work-with-data.html
    public static final String ATHENA_SAMPLE_QUERY = "SELECT * FROM scott2"; // change
the Query statement to match                                       // your
environment
    public static final long SLEEP_AMOUNT_IN_MS = 1000;
    public static final String ATHENA_DEFAULT_DATABASE = "mydatabase"; // change the
database to match your database

}
```

## Crear un cliente para obtener acceso a Athena

La clase `AthenaClientFactory.java` muestra cómo crear y configurar un cliente de Amazon Athena.

```
package aws.example.athena;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.athena.AthenaClient;
import software.amazon.awssdk.services.athena.AthenaClientBuilder;

public class AthenaClientFactory {
    private final AthenaClientBuilder builder = AthenaClient.builder()
        .region(Region.US_WEST_2)
        .credentialsProvider(ProfileCredentialsProvider.create());

    public AthenaClient createClient() {
        return builder.build();
    }
}
```

## Iniciar la ejecución de una consulta

`StartQueryExample` muestra cómo enviar una consulta a Athena, esperar hasta que los resultados estén disponibles y, a continuación, procesar los resultados.

```
package aws.example.athena;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.athena.AthenaClient;
import software.amazon.awssdk.services.athena.model.QueryExecutionContext;
import software.amazon.awssdk.services.athena.model.ResultConfiguration;
import software.amazon.awssdk.services.athena.model.StartQueryExecutionRequest;
import software.amazon.awssdk.services.athena.model.StartQueryExecutionResponse;
import software.amazon.awssdk.services.athena.model.AthenaException;
import software.amazon.awssdk.services.athena.model.GetQueryExecutionRequest;
import software.amazon.awssdk.services.athena.model.GetQueryExecutionResponse;
import software.amazon.awssdk.services.athena.model.QueryExecutionState;
import software.amazon.awssdk.services.athena.model.GetQueryResultsRequest;
import software.amazon.awssdk.services.athena.model.GetQueryResultsResponse;
import software.amazon.awssdk.services.athena.model.ColumnInfo;
```

```
import software.amazon.awssdk.services.athena.model.Row;
import software.amazon.awssdk.services.athena.model.Datum;
import software.amazon.awssdk.services.athena.paginators.GetQueryResultsIterable;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class StartQueryExample {

    public static void main(String[] args) throws InterruptedException {
        AthenaClient athenaClient = AthenaClient.builder()
            .region(Region.US_WEST_2)
            .build();

        String queryExecutionId = submitAthenaQuery(athenaClient);
        waitForQueryToComplete(athenaClient, queryExecutionId);
        processResultRows(athenaClient, queryExecutionId);
        athenaClient.close();
    }

    // Submits a sample query to Amazon Athena and returns the execution ID of the
    // query.
    public static String submitAthenaQuery(AthenaClient athenaClient) {
        try {
            // The QueryExecutionContext allows us to set the database.
            QueryExecutionContext queryExecutionContext =
                QueryExecutionContext.builder()
                    .database(ExampleConstants.ATHENA_DEFAULT_DATABASE)
                    .build();

            // The result configuration specifies where the results of the query should
            // go.
            ResultConfiguration resultConfiguration = ResultConfiguration.builder()
                .outputLocation(ExampleConstants.ATHENA_OUTPUT_BUCKET)
                .build();

            StartQueryExecutionRequest startQueryExecutionRequest =
                StartQueryExecutionRequest.builder()
```

```

        .queryString(ExampleConstants.ATHENA_SAMPLE_QUERY)
        .queryExecutionContext(queryExecutionContext)
        .resultConfiguration(resultConfiguration)
        .build();

    StartQueryExecutionResponse startQueryExecutionResponse = athenaClient
        .startQueryExecution(startQueryExecutionRequest);
    return startQueryExecutionResponse.queryExecutionId();

} catch (AthenaException e) {
    e.printStackTrace();
    System.exit(1);
}
return "";
}

// Wait for an Amazon Athena query to complete, fail or to be cancelled.
public static void waitForQueryToComplete(AthenaClient athenaClient, String
queryExecutionId)
    throws InterruptedException {
    GetQueryExecutionRequest getQueryExecutionRequest =
GetQueryExecutionRequest.builder()
        .queryExecutionId(queryExecutionId)
        .build();

    GetQueryExecutionResponse getQueryExecutionResponse;
    boolean isQueryStillRunning = true;
    while (isQueryStillRunning) {
        getQueryExecutionResponse =
athenaClient.getQueryExecution(getQueryExecutionRequest);
        String queryState =
getQueryExecutionResponse.queryExecution().status().state().toString();
        if (queryState.equals(QueryExecutionState.FAILED.toString())) {
            throw new RuntimeException(
                "The Amazon Athena query failed to run with error message: " +
getQueryExecutionResponse
                    .queryExecution().status().stateChangeReason());
        } else if (queryState.equals(QueryExecutionState.CANCELLED.toString())) {
            throw new RuntimeException("The Amazon Athena query was cancelled.");
        } else if (queryState.equals(QueryExecutionState.SUCCEEDED.toString())) {
            isQueryStillRunning = false;
        } else {
            // Sleep an amount of time before retrying again.
            Thread.sleep(ExampleConstants.SLEEP_AMOUNT_IN_MS);

```

```
        }
        System.out.println("The current status is: " + queryState);
    }
}

// This code retrieves the results of a query
public static void processResultRows(AthenaClient athenaClient, String
queryExecutionId) {
    try {
        // Max Results can be set but if its not set,
        // it will choose the maximum page size.
        GetQueryResultsRequest getQueryResultsRequest =
GetQueryResultsRequest.builder()
            .queryExecutionId(queryExecutionId)
            .build();

        GetQueryResultsIterable getQueryResultsResults = athenaClient
            .getQueryResultsPaginator(getQueryResultsRequest);
        for (GetQueryResultsResponse result : getQueryResultsResults) {
            List<ColumnInfo> columnInfoList =
result.resultSet().resultSetMetadata().columnInfo();
            List<Row> results = result.resultSet().rows();
            processRow(results, columnInfoList);
        }

    } catch (AthenaException e) {
        e.printStackTrace();
        System.exit(1);
    }
}

private static void processRow(List<Row> row, List<ColumnInfo> columnInfoList) {
    for (Row myRow : row) {
        List<Datum> allData = myRow.data();
        for (Datum data : allData) {
            System.out.println("The value of the column is " +
data.varCharValue());
        }
    }
}
}
```

## Detener la ejecución de una consulta

`StopQueryExecutionExample` ejecuta una consulta de ejemplo, detiene la consulta inmediatamente y comprueba el estado de esta para garantizar que se ha cancelado.

```
package aws.example.athena;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.athena.AthenaClient;
import software.amazon.awssdk.services.athena.model.StopQueryExecutionRequest;
import software.amazon.awssdk.services.athena.model.GetQueryExecutionRequest;
import software.amazon.awssdk.services.athena.model.GetQueryExecutionResponse;
import software.amazon.awssdk.services.athena.model.QueryExecutionState;
import software.amazon.awssdk.services.athena.model.AthenaException;
import software.amazon.awssdk.services.athena.model.QueryExecutionContext;
import software.amazon.awssdk.services.athena.model.ResultConfiguration;
import software.amazon.awssdk.services.athena.model.StartQueryExecutionRequest;
import software.amazon.awssdk.services.athena.model.StartQueryExecutionResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class StopQueryExecutionExample {
    public static void main(String[] args) {
        AthenaClient athenaClient = AthenaClient.builder()
            .region(Region.US_WEST_2)
            .build();

        String sampleQueryExecutionId = submitAthenaQuery(athenaClient);
        stopAthenaQuery(athenaClient, sampleQueryExecutionId);
        athenaClient.close();
    }

    public static void stopAthenaQuery(AthenaClient athenaClient, String
sampleQueryExecutionId) {
        try {
            StopQueryExecutionRequest stopQueryExecutionRequest =
StopQueryExecutionRequest.builder()
```



```
        .queryExecutionId(sampleQueryExecutionId)
        .build();

    athenaClient.stopQueryExecution(stopQueryExecutionRequest);
    GetQueryExecutionRequest getQueryExecutionRequest =
    GetQueryExecutionRequest.builder()
        .queryExecutionId(sampleQueryExecutionId)
        .build();

    GetQueryExecutionResponse getQueryExecutionResponse = athenaClient
        .getQueryExecution(getQueryExecutionRequest);
    if (getQueryExecutionResponse.queryExecution()
        .status()
        .state()
        .equals(QueryExecutionState.CANCELLED)) {

        System.out.println("The Amazon Athena query has been cancelled!");
    }

} catch (AthenaException e) {
    e.printStackTrace();
    System.exit(1);
}
}

// Submits an example query and returns a query execution Id value
public static String submitAthenaQuery(AthenaClient athenaClient) {
    try {
        QueryExecutionContext queryExecutionContext =
    QueryExecutionContext.builder()
        .database(ExampleConstants.ATHENA_DEFAULT_DATABASE)
        .build();

        ResultConfiguration resultConfiguration = ResultConfiguration.builder()
            .outputLocation(ExampleConstants.ATHENA_OUTPUT_BUCKET)
            .build();

        StartQueryExecutionRequest startQueryExecutionRequest =
    StartQueryExecutionRequest.builder()
        .queryExecutionContext(queryExecutionContext)
        .queryString(ExampleConstants.ATHENA_SAMPLE_QUERY)
        .resultConfiguration(resultConfiguration).build();

        StartQueryExecutionResponse startQueryExecutionResponse = athenaClient
```

```
        .startQueryExecution(startQueryExecutionRequest);
        return startQueryExecutionResponse.queryExecutionId();

    } catch (AthenaException e) {
        e.printStackTrace();
        System.exit(1);
    }
    return null;
}
}
```

## Generar una lista de ejecuciones de consultas

`ListQueryExecutionsExample` muestra cómo obtener una lista de ID de ejecución de consultas.

```
package aws.example.athena;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.athena.AthenaClient;
import software.amazon.awssdk.services.athena.model.AthenaException;
import software.amazon.awssdk.services.athena.model.ListQueryExecutionsRequest;
import software.amazon.awssdk.services.athena.model.ListQueryExecutionsResponse;
import software.amazon.awssdk.services.athena.paginators.ListQueryExecutionsIterable;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class ListQueryExecutionsExample {
    public static void main(String[] args) {
        AthenaClient athenaClient = AthenaClient.builder()
            .region(Region.US_WEST_2)
            .build();

        listQueryIds(athenaClient);
        athenaClient.close();
    }
}
```

```
public static void listQueryIds(AthenaClient athenaClient) {
    try {
        ListQueryExecutionsRequest listQueryExecutionsRequest =
ListQueryExecutionsRequest.builder().build();
        ListQueryExecutionsIterable listQueryExecutionResponses = athenaClient
            .listQueryExecutionsPaginator(listQueryExecutionsRequest);
        for (ListQueryExecutionsResponse listQueryExecutionResponse :
listQueryExecutionResponses) {
            List<String> queryExecutionIds =
listQueryExecutionResponse.queryExecutionIds();
            System.out.println("\n" + queryExecutionIds);
        }

    } catch (AthenaException e) {
        e.printStackTrace();
        System.exit(1);
    }
}
}
```

## Crear una consulta con nombre

CreateNamedQueryExample cómo crear y asignar un nombre a una consulta.

```
package aws.example.athena;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.athena.AthenaClient;
import software.amazon.awssdk.services.athena.model.AthenaException;
import software.amazon.awssdk.services.athena.model.CreateNamedQueryRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */

public class CreateNamedQueryExample {
    public static void main(String[] args) {
```

```
final String USAGE = ""

    Usage:
        <name>

    Where:
        name - the name of the Amazon Athena query.\s
    """;

if (args.length != 1) {
    System.out.println(USAGE);
    System.exit(1);
}

String name = args[0];
AthenaClient athenaClient = AthenaClient.builder()
    .region(Region.US_WEST_2)
    .build();

createNamedQuery(athenaClient, name);
athenaClient.close();
}

public static void createNamedQuery(AthenaClient athenaClient, String name) {
    try {
        // Create the named query request.
        CreateNamedQueryRequest createNamedQueryRequest =
CreateNamedQueryRequest.builder()
        .database(ExampleConstants.ATHENA_DEFAULT_DATABASE)
        .queryString(ExampleConstants.ATHENA_SAMPLE_QUERY)
        .description("Sample Description")
        .name(name)
        .build();

        athenaClient.createNamedQuery(createNamedQueryRequest);
        System.out.println("Done");

    } catch (AthenaException e) {
        e.printStackTrace();
        System.exit(1);
    }
}
}
```

## Eliminar una consulta con nombre

DeleteNamedQueryExample muestra cómo eliminar una consulta con nombre mediante el ID de consulta con nombre.

```
package aws.example.athena;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.athena.AthenaClient;
import software.amazon.awssdk.services.athena.model.DeleteNamedQueryRequest;
import software.amazon.awssdk.services.athena.model.AthenaException;
import software.amazon.awssdk.services.athena.model.CreateNamedQueryRequest;
import software.amazon.awssdk.services.athena.model.CreateNamedQueryResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class DeleteNamedQueryExample {
    public static void main(String[] args) {
        final String USAGE = ""

            Usage:
                <name>

            Where:
                name - the name of the Amazon Athena query.\s
            """;

        if (args.length != 1) {
            System.out.println(USAGE);
            System.exit(1);
        }

        String name = args[0];
        AthenaClient athenaClient = AthenaClient.builder()
            .region(Region.US_WEST_2)
            .build();
    }
}
```

```
String sampleNamedQueryId = getNamedQueryId(athenaClient, name);
deleteQueryName(athenaClient, sampleNamedQueryId);
athenaClient.close();
}

public static void deleteQueryName(AthenaClient athenaClient, String
sampleNamedQueryId) {
    try {
        DeleteNamedQueryRequest deleteNamedQueryRequest =
DeleteNamedQueryRequest.builder()
            .namedQueryId(sampleNamedQueryId)
            .build();

        athenaClient.deleteNamedQuery(deleteNamedQueryRequest);

    } catch (AthenaException e) {
        e.printStackTrace();
        System.exit(1);
    }
}

public static String getNamedQueryId(AthenaClient athenaClient, String name) {
    try {
        CreateNamedQueryRequest createNamedQueryRequest =
CreateNamedQueryRequest.builder()
            .database(ExampleConstants.ATHENA_DEFAULT_DATABASE)
            .queryString(ExampleConstants.ATHENA_SAMPLE_QUERY)
            .name(name)
            .description("Sample description")
            .build();

        CreateNamedQueryResponse createNamedQueryResponse =
athenaClient.createNamedQuery(createNamedQueryRequest);
        return createNamedQueryResponse.namedQueryId();

    } catch (AthenaException e) {
        e.printStackTrace();
        System.exit(1);
    }
    return null;
}
}
```

## Generar una lista de consultas con nombre

ListNamedQueryExample muestra cómo obtener una lista de ID de consultas con nombre.

```
package aws.example.athena;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.athena.AthenaClient;
import software.amazon.awssdk.services.athena.model.AthenaException;
import software.amazon.awssdk.services.athena.model.ListNamedQueriesRequest;
import software.amazon.awssdk.services.athena.model.ListNamedQueriesResponse;
import software.amazon.awssdk.services.athena.paginators.ListNamedQueriesIterable;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class ListNamedQueryExample {
    public static void main(String[] args) {
        AthenaClient athenaClient = AthenaClient.builder()
            .region(Region.US_WEST_2)
            .build();

        listNamedQueries(athenaClient);
        athenaClient.close();
    }

    public static void listNamedQueries(AthenaClient athenaClient) {
        try {
            ListNamedQueriesRequest listNamedQueriesRequest =
                ListNamedQueriesRequest.builder()
                    .build();

            ListNamedQueriesIterable listNamedQueriesResponses = athenaClient
                .listNamedQueriesPaginator(listNamedQueriesRequest);
            for (ListNamedQueriesResponse listNamedQueriesResponse :
                listNamedQueriesResponses) {
                List<String> namedQueryIds = listNamedQueriesResponse.namedQueryIds();
            }
        }
    }
}
```

```
        System.out.println(namedQueryIds);
    }

    } catch (AthenaException e) {
        e.printStackTrace();
        System.exit(1);
    }
}
}
```



# Uso de Apache Spark en Amazon Athena

Amazon Athena facilita la ejecución interactiva del análisis y la exploración de datos mediante Apache Spark sin necesidad de planificar, configurar ni administrar los recursos. Ejecutar aplicaciones de Apache Spark en Athena significa enviar el código de Spark para su procesamiento y recibir los resultados directamente sin necesidad de configuración adicional. Puede utilizar la experiencia simplificada de cuadernos de la consola de Amazon Athena para desarrollar aplicaciones de Apache Spark mediante las API de cuadernos de Athena o Python. Apache Spark en Amazon Athena no requiere servidor y proporciona un escalado automático y bajo demanda que ofrece computación instantánea para cumplir con los cambios en los volúmenes de datos y los requisitos de procesamiento.

Amazon Athena ofrece las siguientes características:

- **Uso de la consola:** envíe aplicaciones de Spark desde la consola de Amazon Athena.
- **Creación de scripts:** cree y depure aplicaciones de Apache Spark en Python de forma rápida e interactiva.
- **Escalado dinámico:** Amazon Athena determina automáticamente los recursos de procesamiento y memoria necesarios para ejecutar un trabajo y los escala continuamente en consecuencia hasta los máximos que usted especifique. Este escalado dinámico reduce los costos sin afectar a la velocidad.
- **Experiencia con cuadernos:** utilice el editor de cuadernos de Athena para crear, editar y ejecutar cálculos mediante una interfaz familiar. Los cuadernos de Athena son compatibles con los cuadernos de Jupyter y contienen una lista de celdas que se ejecutan en orden como cálculos. El contenido de las celdas puede incluir código, texto, Markdown, cálculos, gráficos y contenido multimedia enriquecido.

Para obtener más información, consulte [Explore su lago de datos con Amazon Athena para Apache Spark](#) en el blog sobre macrodatos de AWS.

## Consideraciones y limitaciones

- Actualmente, Amazon Athena para Apache Spark está disponible en las siguientes Regiones de AWS:
  - Asia-Pacífico (Mumbai)

- Asia-Pacífico (Singapur)
- Asia-Pacífico (Sídney)
- Asia-Pacífico (Tokio)
- Europa (Fráncfort)
- Europa (Irlanda)
- Este de EE. UU. (Norte de Virginia)
- Este de EE. UU. (Ohio)
- Oeste de EE. UU. (Oregón)
- AWS Lake Formation no se admite.
- No se admiten las tablas que utilizan proyección de particiones.
- Los grupos de trabajo habilitados para Apache Spark pueden usar el editor de cuadernos de Athena, pero no el editor de consultas de Athena. Solo los grupos de trabajo de Athena SQL pueden usar el editor de consultas de Athena.
- No se admiten las consultas de vista entre motores. Athena para Spark no puede consultar las vistas creadas por Athena SQL. Dado que las vistas de los dos motores se implementan de manera diferente, no son compatibles para el uso entre motores.
- No se admiten MLlib (biblioteca de machine learning de Apache Spark) y el paquete `pyspark.ml`. Para obtener una lista de las bibliotecas Python compatibles, consulte [Lista de bibliotecas de Python preinstaladas](#).
- Actualmente, `pip install` no es compatible con Athena para las sesiones de Spark.
- Solo se permite una sesión activa por cuaderno.
- Cuando varios usuarios utilizan la consola para abrir una sesión existente en un grupo de trabajo, acceden al mismo cuaderno. Para evitar confusiones, abra solo las sesiones que cree usted mismo.
- Los dominios de host de las aplicaciones de Apache Spark que puede utilizar con Amazon Athena (por ejemplo, `analytics-gateway.us-east-1.amazonaws.com`) están registrados en la [Lista de sufijos públicos \(PSL\)](#) de Internet. Si alguna vez necesita configurar cookies confidenciales en sus dominios, le recomendamos que utilice cookies con un prefijo `__Host-` para proteger su dominio de los intentos de falsificación de solicitudes entre sitios (CSRF). Para obtener más información, consulte la página [Set-Cookie](#) en la red de desarrolladores de Mozilla.org.
- Para obtener información sobre la solución de problemas de cuadernos, sesiones y grupos de trabajo de Spark en Athena, consulte [Solución de problemas de Athena para Spark](#).

# Introducción a Apache Spark en Amazon Athena

Para empezar a usar Apache Spark en Amazon Athena, primero cree un grupo de trabajo habilitado para Spark. Después de cambiar al grupo de trabajo, puede crear un cuaderno o abrir uno existente. Al abrir un cuaderno en Athena, se inicia automáticamente una nueva sesión y puede trabajar con él directamente en el editor de cuadernos de Athena.

## Note

Asegúrese de crear un grupo de trabajo habilitado para Spark antes de crear un cuaderno.

## Creación de un grupo de trabajo habilitado para Spark en Athena

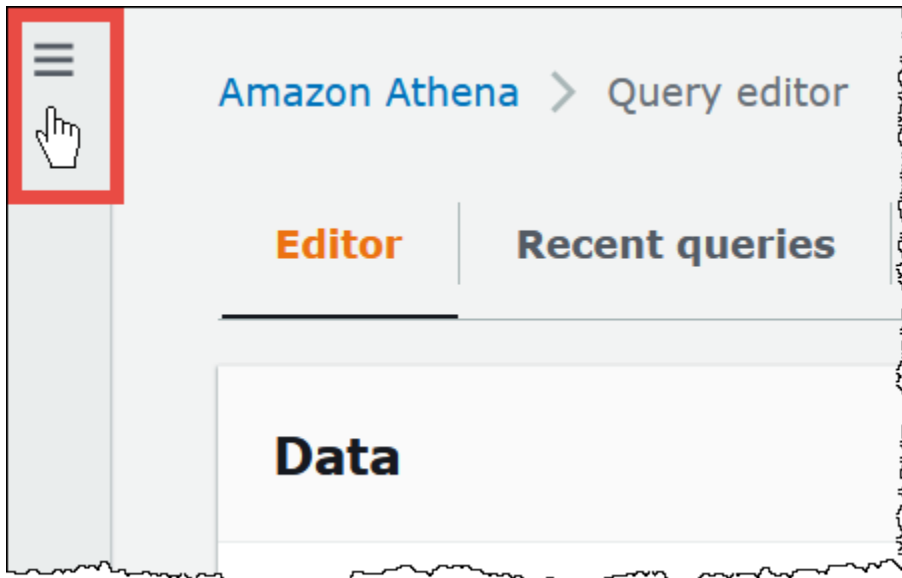
Puede usar [grupos de trabajo](#) en Athena para agrupar usuarios, equipos, aplicaciones o cargas de trabajo, así como para realizar un seguimiento de los costos. Para usar Apache Spark en Amazon Athena, debe crear un grupo de trabajo de Amazon Athena que utilice un motor de Spark.

## Note

Los grupos de trabajo habilitados para Apache Spark pueden usar el editor de cuadernos de Athena, pero no el editor de consultas de Athena. Solo los grupos de trabajo de Athena SQL pueden usar el editor de consultas de Athena.

Para crear un grupo de trabajo habilitado para Spark en Athena

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.



3. En el panel de navegación, elija Grupos de trabajo.
4. En el panel Grupos de trabajo, elija Crear grupo de trabajo.
5. En Workgroup name (Nombre del grupo de trabajo), ingrese un nombre para el grupo de trabajo de Apache Spark.
6. (Opcional) En Description (Descripción), ingrese una descripción para el grupo de trabajo.
7. En Analytics engine (Motor de análisis), elija Apache Spark.

**Note**

Tras crear un grupo de trabajo, no se puede cambiar su tipo de motor de análisis. Por ejemplo, un grupo de trabajo de la versión 3 del motor de Athena no se puede cambiar a un grupo de trabajo de la versión 3 del motor de PySpark.

8. Para los fines de este tutorial, seleccione Turn on example notebook (Activar un cuaderno de ejemplo). Esta característica opcional agrega un cuaderno de ejemplo con el nombre `example-notebook-random_string` al grupo de trabajo y agrega los permisos asociados a AWS Glue que el cuaderno utiliza para crear, mostrar y eliminar bases de datos y tablas específicas de su cuenta, así como los permisos de lectura en Amazon S3 para el conjunto de datos de ejemplo. Para ver los permisos agregados, elija View additional permissions details (Ver detalles de permisos adicionales).

 Note

La ejecución del cuaderno de ejemplo puede suponer un costo adicional.

9. En **Additional configurations** (Configuraciones adicionales), haga lo siguiente:
  - Utilice la configuración **Use defaults** (Usar valores predeterminados). Esta opción es la predeterminada y lo ayuda a empezar con su grupo de trabajo habilitado para Spark. Con esta opción, Athena crea para usted un rol de IAM y una ubicación de resultados de cálculo en Amazon S3. El nombre del rol de IAM y la ubicación del bucket de S3 que se van a crear aparecen en el cuadro situado debajo del encabezado **Additional configurations** (Configuraciones adicionales).
  - Desactive la configuración **Use defaults** (Usar valores predeterminados) y, a continuación, continúe con los pasos de la sección [Especificación de configuraciones propias de grupo de trabajo](#) para configurar el grupo de trabajo manualmente.
10. (Opcional) **Tags** (Etiquetas): utilice esta opción para agregar etiquetas al grupo de trabajo. Para obtener más información, consulte [Etiquetado de recursos de Athena](#).
11. Elija **Crear grupo de trabajo**. Un mensaje le informa de que el grupo de trabajo se ha creado correctamente y aparecerá en la lista de grupos de trabajo.

## Especificación de configuraciones propias de grupo de trabajo

Si desea especificar su propio rol de IAM y ubicación para los resultados de cálculo del cuaderno, siga los pasos de esta sección. Si ha elegido **Use defaults** (Usar valores predeterminados) en la opción **Additional configurations** (Configuraciones adicionales), omita esta sección y vaya directamente a [Apertura del explorador de cuadernos y cambio de grupo de trabajo](#).

El siguiente procedimiento supone que ha completado los pasos del 1 al 9 del procedimiento **Para crear un grupo de trabajo habilitado para Spark en Athena** de la sección anterior.

Para especificar configuraciones propias de grupo de trabajo

1. Si desea crear o utilizar su propio rol de IAM o configurar el cifrado del cuaderno, expanda **IAM role configuration** (Configuración del rol de IAM).
  - En **Service Role** (Rol de servicio), realice una de las operaciones siguientes:

- **Create a service role (Crear un rol de servicio):** elija esta opción para que Athena cree un rol de servicio para usted. Para ver los permisos que concede el rol, elija [View permission details \(Ver detalles de los permisos\)](#).
- **Choose an existing service role (Elegir un rol de servicio existente):** en el menú desplegable, elija un rol existente. El rol que elija debe incluir los permisos de la primera opción. Para obtener más información sobre los permisos de los grupos de trabajo habilitados para cuadernos, consulte [Solución de problemas de grupos de trabajo habilitados para Spark](#).
- **En Notebook and calculation code encryption key management (Administrar las claves de cifrado del cuaderno y el código de cálculo),** elija una de las siguientes opciones:
  - **Owned by Amazon Athena (Propiedad de Amazon Athena):** la clave de AWS KMS pertenece a Amazon Athena, que también se ocupa de su administración. No se cobrará ningún cargo adicional por el uso de esta clave.
  - **A symmetric key stored in your account, owned and managed by you (Una clave simétrica almacenada en su cuenta, de su propiedad y administrada por usted):** en esta opción, realice una de las siguientes acciones:
    - Para usar una clave existente, utilice el cuadro de búsqueda para elegir una clave de AWS KMS o ingresar un ARN de clave.
    - Para crear una clave en la consola de AWS KMS, elija [Crear una clave de AWS KMS](#). Su rol de ejecución debe tener permiso para utilizar la clave que cree.


#### Important

Al cambiar la [AWS KMS key](#) de un grupo de trabajo, los cuadernos administrados antes de la actualización siguen haciendo referencia a la antigua clave de KMS. Los cuadernos administrados después de la actualización utilizan la nueva clave de KMS. Para actualizar los cuadernos antiguos para que hagan referencia a la nueva clave de KMS, exporte y después importe cada uno de los cuadernos antiguos. Si elimina la clave de KMS antigua antes de actualizar las referencias de los cuadernos antiguos a la nueva clave de KMS, los cuadernos antiguos ya no se pueden descifrar ni recuperar.

Este comportamiento también se aplica a las actualizaciones de los [alias](#), que son nombres descriptivos de las claves de KMS. Al actualizar un alias de clave de KMS para que apunte a una nueva clave de KMS, los cuadernos administrados antes de la actualización del alias siguen haciendo referencia a la clave de KMS antigua, y los

cuadernos administrados después de la actualización del alias utilizan la nueva clave de KMS. Tenga en cuenta estos puntos antes de actualizar las claves o alias de KMS.

2. Si desea especificar una configuración propia de los resultados de cálculo, expanda Calculation result settings (Configuraciones de los resultados de cálculo) y, a continuación, elija una de las siguientes opciones.
  - Create a new S3 bucket (Crear un nuevo bucket de S3): esta opción crea un bucket de Amazon S3 en su cuenta para los resultados de sus cálculos. El nombre del bucket tiene el formato `account_id-region-athena-results-bucket-alphanumeric_id` y utiliza las configuraciones de ACL inhabilitadas, acceso público bloqueado, control de versiones inhabilitado y propietario de bucket aplicado.
  - Choose an existing S3 location (Elegir una ubicación de S3 existente): en esta opción, haga lo siguiente:
    - Ingrese la ruta de S3 a una ubicación existente en el cuadro de búsqueda o elija Browse S3 (Examinar S3) para elegir un bucket de una lista.

 Note

Al seleccionar una ubicación existente en Amazon S3, no agregue una barra diagonal (/) a la ubicación. Si lo hace, el enlace a la ubicación de los resultados del cálculo en la [página de detalles del cálculo](#) apuntará al directorio equivocado. Si esto ocurre, edite la ubicación de los resultados del grupo de trabajo para eliminar la barra diagonal situada al final.

- (Opcional) Elija View (Ver) para abrir la página Buckets de la consola de Amazon S3, donde podrá ver más información sobre el bucket existente que haya elegido.
- (Opcional) En Expected bucket owner (Propietario esperado del bucket), ingrese el ID de cuenta de AWS que espera que sea el propietario de su bucket de ubicación de salida de los resultados de las consultas. Le recomendamos que elija esta opción como medida de seguridad adicional siempre que sea posible. Si el ID de cuenta del propietario del bucket no coincide con el ID que especifique, los intentos de generar el bucket generarán errores. Para obtener información detallada, consulte [Verificación de la propiedad del bucket con la condición de propietario del bucket](#) en la Guía del usuario de Amazon S3.
- (Opcional) Seleccione Assign bucket owner full control over query results (Asignar al propietario del bucket control total sobre los resultados de las consultas) si la ubicación de

los resultados de cálculo pertenece a otra cuenta y desea concederle el control total sobre los resultados de sus consultas.

3. (Opcional) Seleccione **Encrypt calculation results** (Cifrar resultados de cálculo) y, a continuación, elija una de las siguientes opciones:
  - **SSE\_S3**: una clave de cifrado del servidor administrada por S3.
  - **SSE\_KMS**: una clave que usted proporciona. En **Elegir una clave de AWS KMS**, puede elegir una de las siguientes opciones:
    - Usar clave que pertenece a AWS: utilice una clave que AWS administra y tiene en propiedad.
    - Elegir una clave de AWS KMS diferente (avanzado): elija o cree una clave.
      - Para usar una clave existente, utilice el cuadro de búsqueda para elegir una clave de AWS KMS o ingresar un ARN de clave.
      - Para crear una clave en la consola de KMS, elija **Crear una clave de AWS KMS**. Después de crear la clave en la consola de KMS, vuelva a la página **Crear grupo de trabajo** de la consola de Athena y, a continuación, utilice el cuadro de búsqueda **Elegir una clave de AWS KMS** o ingresar un ARN para elegir la clave que acaba de crear.
4. (Opcional) **Other settings** (Otras configuraciones): expanda esta opción para activar o desactivar la opción **Publish CloudWatch metrics** (Publicar métricas de CloudWatch) del grupo de trabajo. Este campo se selecciona de forma predeterminada. Para obtener más información, consulte [Supervisión de los cálculos de Apache Spark con métricas de CloudWatch](#).
5. (Opcional) **Tags** (Etiquetas): utilice esta opción para agregar etiquetas al grupo de trabajo. Para obtener más información, consulte [Etiquetado de recursos de Athena](#).
6. Elija **Crear grupo de trabajo**. Un mensaje le informa de que el grupo de trabajo se ha creado correctamente y aparecerá en la lista de grupos de trabajo.

## Apertura del explorador de cuadernos y cambio de grupo de trabajo

Antes de poder usar el grupo de trabajo habilitado para Spark que acaba de crear, debe cambiar de grupo de trabajo. Para cambiar los grupos de trabajo habilitados para Spark, puede usar la opción **Workgroup** (Grupo de trabajo) del explorador o del editor de cuadernos.



**Note**

Antes de empezar, compruebe que el navegador no bloquea las cookies de terceros. Cualquier navegador que bloquee las cookies de terceros, ya sea de manera predeterminada o porque el usuario lo haya configurado así, impedirá que se inicien los cuadernos. Para obtener más información sobre cómo administrar las cookies, consulte:

- [Chrome](#)
- [Firefox](#)
- [Safari](#)

Para abrir el explorador de cuadernos y cambiar de grupo de trabajo

1. En el panel de navegación, elija Notebook explorer (Explorador de cuadernos).
2. Use la opción Workgroup (Grupo de trabajo) de la parte superior derecha de la consola para elegir el grupo de trabajo habilitado para Spark que ha creado. El cuaderno de ejemplo se muestra en la lista de cuadernos.

Puede usar el explorador de cuadernos de las siguientes formas:

- Elija el nombre vinculado de un cuaderno para abrirlo en una nueva sesión.
- Para cambiar el nombre del cuaderno, eliminarlo o exportarlo, utilice el menú Actions (Acciones).
- Para importar un archivo de cuaderno, elija Import file (Importar archivo).
- Para crear un cuaderno, elija Create notebook (Crear cuaderno).

## Ejecución del cuaderno de ejemplo

El cuaderno de ejemplo consulta datos de un conjunto de datos de viajes en taxi en la ciudad de Nueva York de acceso público. El cuaderno tiene ejemplos que muestran cómo trabajar con Spark DataFrames, Spark SQL y AWS Glue Data Catalog.

Para ejecutar el cuaderno de ejemplo

1. En el explorador de cuadernos, elija el nombre vinculado del cuaderno de ejemplo.

De esta forma, se inicia una sesión de cuaderno con los parámetros predeterminados y se abre el cuaderno en el editor de cuadernos. Un mensaje le informa de que se ha iniciado una nueva sesión de Apache Spark con los parámetros predeterminados (20 DPU como máximo).

- Para ejecutar las celdas en orden y ver los resultados, pulse el botón Run (Ejecutar) una vez para cada celda del cuaderno.
  - Desplácese hacia abajo para ver los resultados así como nuevas celdas.
  - En las celdas que tienen un cálculo, una barra de progreso muestra el porcentaje completado, el tiempo transcurrido y el tiempo restante.
  - El cuaderno de ejemplo crea una base de datos y una tabla de ejemplo en su cuenta. La última celda las elimina como paso de limpieza.

#### Note

Si cambia los nombres de carpetas, tablas o bases de datos en el cuaderno de ejemplo, asegúrese de que esos cambios se reflejen en los roles de IAM que utilice. De lo contrario, es posible que el cuaderno no se ejecute debido a la falta de permisos.

## Edición de detalles de la sesión

Tras iniciar una sesión de cuaderno, puede editar los detalles de la sesión, como el formato de la tabla, el cifrado, el tiempo de inactividad de la sesión y el número máximo de unidades de procesamiento de datos (DPU) simultáneas que desea utilizar. Una DPU es una medida relativa de la potencia de procesamiento que consta de 4 vCPU de capacidad de cómputo y 16 GB de memoria.

Para editar los detalles de la sesión

- En el editor de cuadernos, en el menú Session (Sesión) de la esquina superior derecha, elija Edit session (Editar sesión).
- En el cuadro de diálogo Editar detalles de la sesión, en la sección Propiedades de Spark, elija o ingrese valores para las siguientes opciones:
  - Formato de tabla adicional: elija Linux Foundation Delta Lake, Apache Hudi, Apache Iceberg o Personalizada.

- Para las opciones de tabla Delta, Hudi o Iceberg, las propiedades de tabla necesarias para el formato de tabla correspondiente se proporcionan automáticamente en las opciones Editar en tabla y Editar en JSON. Para obtener más información sobre el uso de estos formatos de tabla, consulte [Uso de formatos de tabla que no son de Hive en Amazon Athena para Apache Spark](#).
  - Para agregar o eliminar propiedades de tabla para las tablas personalizadas o de otro tipo, utilice las opciones Editar en tabla y Editar en JSON.
  - Para la opción Editar en la tabla, seleccione Agregar propiedad si desea agregar una propiedad o Eliminar si desea eliminar una propiedad. Utilice los cuadros Clave y Valor para introducir los nombres de las propiedades y sus valores.
  - Para la opción Editar en JSON, utilice el editor de texto JSON a fin de editar la configuración directamente.
    - Seleccione Copiar para copiar el texto JSON en el portapapeles.
    - Seleccione Borrar para eliminar todo el texto del editor JSON.
    - Elija el icono de ajustes (engranaje) a fin de configurar el ajuste de líneas o seleccionar un tema de color para el editor JSON.
  - Activar el cifrado de Spark: seleccione esta opción para cifrar los datos que se escriben en el disco y se envían a través de los nodos de la red de Spark. Para obtener más información, consulte [Habilitación del cifrado de Apache Spark](#).
3. En la sección Parámetros de la sesión, elija o ingrese valores para las siguientes opciones:
- Session idle timeout (Tiempo de inactividad de la sesión): elija o ingrese un valor comprendido entre 1 y 480 minutos. El valor predeterminado es 20.
  - Coordinator size (Tamaño del coordinador): un coordinador es un ejecutor especial que orquesta el trabajo de procesamiento y administra a otros ejecutores en una sesión de cuaderno. Actualmente, 1 DPU es el valor predeterminado y el único posible.
  - Executor size (Tamaño del ejecutor): un ejecutor es la unidad de cálculo más pequeña que una sesión de cuaderno puede solicitar a Athena. Actualmente, 1 DPU es el valor predeterminado y el único posible.
  - Max concurrent value (Valor máximo de simultaneidad): la cantidad máxima de DPU que se pueden ejecutar simultáneamente. El valor predeterminado es 20, el mínimo es 3 y el máximo es 60. Al aumentar este valor no se asignan automáticamente recursos adicionales, pero Athena intentará asignar hasta el máximo especificado cuando la carga de computación lo requiera y haya recursos disponibles.

4. Elija Guardar.
5. Cuando aparezca el mensaje Confirm edit (Confirmar edición), elija Confirm (Confirmar).

Athena guarda el cuaderno e inicia una nueva sesión con los parámetros especificados. Un banner en el editor de cuadernos le informa de que se ha iniciado una nueva sesión con los parámetros modificados.

#### Note

Athena recuerda la configuración de la sesión del cuaderno. Si edita los parámetros de una sesión y, a continuación, la termina, Athena utilizará los parámetros de sesión que configuró la próxima vez que inicie una sesión para el cuaderno.

## Visualización de detalles de sesión y cálculo

Tras ejecutar el cuaderno, podrá ver los detalles de la sesión y del cálculo.

Para ver los detalles de sesión y cálculo

1. En el menú Session (Sesión) de la esquina superior derecha, elija View details (Ver detalles).
  - La pestaña Current session (Sesión actual) muestra información sobre la sesión actual, que incluye el ID de sesión, la hora de creación, el estado y el grupo de trabajo.
  - La pestaña History (Historial) muestra los ID de sesiones anteriores. Para ver los detalles de una sesión anterior, elija la pestaña History (Historial) y, a continuación, elija un ID de sesión de la lista.
  - La sección Calculations (Cálculos) muestra una lista de los cálculos que se ejecutaron en la sesión.
2. Para ver los detalles de un cálculo, elija el ID del cálculo.
3. En la página Calculation details (Detalles del cálculo), puede hacer lo siguiente:
  - Para ver el código del cálculo, consulte la sección Code (Código).
  - Para ver los resultados del cálculo, seleccione la pestaña Results (Resultados).
  - Para descargar los resultados que ve en formato de texto, elija Download results (Descargar resultados).

- Para ver información sobre los resultados del cálculo en Amazon S3, elija View in S3 (Ver en S3).

## Terminación de una sesión

Para finalizar una sesión de cuaderno

1. En el editor de cuadernos, en el menú Session (Sesión) de la esquina superior derecha, elija Terminate (Terminar).
2. Cuando aparezca el mensaje Confirm session termination (Confirmar terminación de sesión), elija Confirm (Confirmar). Se guardará el cuaderno y volverá al editor de cuadernos.

### Note

El cierre de una pestaña de un cuaderno en el editor de cuadernos no termina por sí solo la sesión de un cuaderno activo. Si quiere asegurarse de que la sesión ha terminado, utilice la opción Session (Sesión), Terminate (Terminar).

## Creación de un cuaderno propio

Tras crear un grupo de trabajo de Athena habilitado para Spark, podrá crear un cuaderno propio.

Para crear un cuaderno

1. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.
2. En el panel de navegación de la consola de Athena, elija Notebook explorer (Explorador de cuadernos) o Notebook editor (Editor de cuadernos).
3. Realice una de las acciones siguientes:
  - En Notebook explorer (Explorador de cuadernos), elija Create notebook (Crear cuaderno).
  - En Notebook editor (Editor de cuadernos), elija Create notebook (Crear cuaderno) o elija el icono con el signo más (+) para agregar un cuaderno.
4. En el cuadro de diálogo Create notebook (Crear cuaderno), ingrese un nombre en Notebook name (Nombre del cuaderno).

5. (Opcional) Amplíe Propiedades de Spark y, a continuación, elija o ingrese valores para las siguientes opciones:
  - Formato de tabla adicional: elija Linux Foundation Delta Lake, Apache Hudi, Apache Iceberg o Personalizada.
    - Para las opciones de tabla Delta, Hudi o Iceberg, las propiedades de tabla necesarias para el formato de tabla correspondiente se proporcionan automáticamente en las opciones Editar en tabla y Editar en JSON. Para obtener más información sobre el uso de estos formatos de tabla, consulte [Uso de formatos de tabla que no son de Hive en Amazon Athena para Apache Spark](#).
    - Para agregar o eliminar propiedades de tabla para las tablas personalizadas o de otro tipo, utilice las opciones Editar en tabla y Editar en JSON.
    - Para la opción Editar en la tabla, seleccione Agregar propiedad si desea agregar una propiedad o Eliminar si desea eliminar una propiedad. Utilice los cuadros Clave y Valor para introducir los nombres de las propiedades y sus valores.
    - Para la opción Editar en JSON, utilice el editor de texto JSON a fin de editar la configuración directamente.
      - Seleccione Copiar para copiar el texto JSON en el portapapeles.
      - Seleccione Borrar para eliminar todo el texto del editor JSON.
      - Elija el icono de ajustes (engranaje) a fin de configurar el ajuste de líneas o seleccionar un tema de color para el editor JSON.
  - Activar el cifrado de Spark: seleccione esta opción para cifrar los datos que se escriben en el disco y se envían a través de los nodos de la red de Spark. Para obtener más información, consulte [Habilitación del cifrado de Apache Spark](#).
6. (Opcional) Expanda Session parameters (Parámetros de sesión) y, a continuación, elija o ingrese valores para las siguientes opciones:
  - Session idle timeout (Tiempo de inactividad de la sesión): elija o ingrese un valor comprendido entre 1 y 480 minutos. El valor predeterminado es 20.
  - Coordinator size (Tamaño del coordinador): un coordinador es un ejecutor especial que orquesta el trabajo de procesamiento y administra a otros ejecutores en una sesión de cuaderno. Actualmente, 1 DPU es el valor predeterminado y el único posible. Una DPU (unidad de procesamiento de datos) es una medida relativa de la potencia de procesamiento que consta de 4 vCPU de capacidad de computación y 16 GB de memoria.

- **Executor size (Tamaño del ejecutor):** un ejecutor es la unidad de cálculo más pequeña que una sesión de cuaderno puede solicitar a Athena. Actualmente, 1 DPU es el valor predeterminado y el único posible.
  - **Max concurrent value (Valor máximo de simultaneidad):** la cantidad máxima de DPU que se pueden ejecutar simultáneamente. El valor predeterminado es 20 y el máximo es 60. Al aumentar este valor no se asignan automáticamente recursos adicionales, pero Athena intentará asignar hasta el máximo especificado cuando la carga de computación lo requiera y haya recursos disponibles.
7. Seleccione **Crear**. El cuaderno se abre en una nueva sesión en el editor de cuadernos.

## Apertura de un cuaderno creado anteriormente

Para abrir un cuaderno creado anteriormente

1. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.
2. En el panel de navegación de la consola de Athena, elija **Notebook editor (Editor de cuadernos)** o **Notebook explorer (Explorador de cuadernos)**.
3. Realice una de las acciones siguientes:
  - En **Notebook editor (Editor de cuadernos)**, elija un cuaderno de la lista **Recent notebooks (Cuadernos recientes)** o **Saved notebooks (Cuadernos guardados)**. El cuaderno se abre en una nueva sesión.
  - En **Notebook explorer (Explorador de cuadernos)**, elija el nombre de un cuaderno de la lista. El cuaderno se abre en una nueva sesión.

Para obtener más información sobre la administración de los archivos de cuadernos, consulte [Administración de los archivos de cuaderno](#).

## Uso de los cuadernos

Administre sus cuadernos en el explorador de cuadernos de Athena y edítelos y ejecútelos en sesiones con el editor de cuadernos de Athena. Puede configurar el uso de la DPU para las sesiones de su cuaderno según sus necesidades.

Al detener un cuaderno, se termina la sesión asociada. Se guardan todos los archivos, pero se pierden los cambios en curso en las variables, funciones y clases declaradas. Al reiniciar el cuaderno, Athena vuelve a cargar los archivos del cuaderno y puede volver a ejecutar el código.

## Sesiones y cálculos

Cada cuaderno está asociado a un único núcleo de Python y ejecuta código Python. Un cuaderno puede tener una o más celdas que contengan comandos. Para ejecutar las celdas de un cuaderno, primero debe crear una sesión para el cuaderno. Las sesiones realizan un seguimiento de las variables y el estado de los cuadernos.

Ejecutar una celda en un cuaderno significa ejecutar un cálculo en la sesión actual. Los cálculos hacen avanzar el estado del cuaderno y pueden realizar tareas como leer desde Amazon S3 o escribir en otros almacenes de datos. Mientras se esté ejecutando una sesión, los cálculos utilizan y modifican el estado que se mantiene en el cuaderno.

Cuando ya no necesite el estado, puede finalizar una sesión. Al finalizar una sesión, el cuaderno permanece, pero las variables y el resto de información de estado se destruyen. Si necesita trabajar en varios proyectos al mismo tiempo, puede crear una sesión para cada proyecto y las sesiones serán independientes entre sí.

Las sesiones tienen una capacidad de cálculo dedicada, medida en DPU. Cuando crea una sesión, puede asignar a la sesión una cantidad de DPU. Las diferentes sesiones pueden tener diferentes capacidades según los requisitos de la tarea.

## Uso del editor de cuadernos de Athena

El editor de cuadernos de Athena es un entorno interactivo para escribir y ejecutar código. En las siguientes secciones, se describen las características del entorno.

### Modo de comandos frente a modo de edición

El editor de cuadernos tiene una interfaz de usuario modal: un modo de edición para ingresar texto en una celda y un modo de comandos para enviar comandos al propio editor, como copiar, pegar o ejecutar.

Para utilizar el modo de edición y el modo de comandos, puede realizar las siguientes tareas:

- Para ingresar al modo de edición, pulse **ENTER** o elija una celda. Cuando una celda está en modo de edición, su margen izquierdo está en verde.



- Para entrar en el modo de comandos, pulse **ESC** o haga clic fuera de una celda. Tenga en cuenta que, normalmente, los comandos solo se aplican a la celda seleccionada actualmente, no a todas las celdas. Cuando el editor está en modo de comandos, el margen izquierdo de la celda está en azul.
- En el modo de comandos, puede utilizar los atajos de teclado y el menú situado encima del editor, pero no puede ingresar texto en celdas individuales.
- Para seleccionar una celda, elíjala.
- Para seleccionar todas las celdas, pulse **Ctrl+A** (Windows) o **Cmd+A** (Mac).

## Menú del editor de cuadernos

Los iconos del menú situados en la parte superior del editor de cuadernos ofrecen las siguientes opciones:

- Save (Guardar): guarda el estado actual del cuaderno.
- Insert cell below (Insertar celda debajo): agrega una celda nueva (vacía) debajo de la seleccionada actualmente.
- Cut selected cells (Suprimir celdas seleccionadas): elimina la celda seleccionada de su ubicación actual y la copia en la memoria.
- Copy selected cells (Copiar celdas seleccionadas): copia la celda seleccionada en la memoria.
- Paste cells below (Pegar celdas debajo): pega la celda copiada debajo de la celda actual.
- Move selected cells up (Mover celdas seleccionadas hacia arriba): sitúa la celda actual encima de la celda superior.
- Move selected cells down (Mover celdas seleccionadas hacia abajo): sitúa la celda actual debajo de la celda inferior.
- Run (Ejecutar): ejecuta la celda actual (seleccionada). El resultado se muestra justo debajo de la celda actual.
- Run all (Ejecutar todo): ejecuta todas las celdas del cuaderno. El resultado de cada celda se muestra justo debajo de la celda.
- Stop (Interrupt the kernel) (Detener [interrumpir el kernel]): interrumpe el kernel para detener el cuaderno actual.
- Format option (Opción de formato): selecciona el formato de celda, que puede ser uno de los siguientes:

- **Code:** se usa para código Python (predeterminado).
- **Markdown:** se usa para ingresar texto en formato [Markdown al estilo de GitHub](#). Para renderizar el Markdown, ejecute la celda.
- **Raw NBConvert:** se usa para ingresar texto en forma no modificada. Las celdas marcadas como Raw NBConvert se pueden convertir a un formato diferente, como HTML, mediante la herramienta de línea de comandos [nbconvert](#) de Jupyter.
- **Heading (Encabezado):** se usa para cambiar el nivel del encabezado de la celda.
- **Command palette (Paleta de comandos):** contiene los comandos del cuaderno de Jupyter y sus atajos de teclado. Para obtener más información acerca de los atajos de teclado, consulte las secciones que aparecen más adelante en este documento.
- **Session (Sesión):** utilice las opciones de este menú para [ver](#) los detalles de una sesión, [editar los parámetros de la sesión](#) o [terminar](#) la sesión.

## Atajos de teclado del modo de comandos

A continuación, se presentan algunos atajos de teclado habituales de un editor de cuadernos en el modo de comandos. Estos atajos están disponibles después de pulsar **ESC** para ingresar al modo de comandos. Para ver una lista completa de los comandos disponibles en el editor, pulse **ESC + H**.

Clave	Acción
<b>1 - 6</b>	Cambiar el tipo de celda a Markdown y establecer el nivel del encabezado en el número escrito
<b>a</b>	Crear una celda encima de la celda actual
<b>b</b>	Crear una celda debajo de la celda actual
<b>c</b>	Copiar la celda actual en memoria
<b>d d</b>	Eliminar la celda actual
<b>h</b>	Mostrar la pantalla de ayuda de los atajos de teclado
<b>j</b>	Ir una celda más abajo
<b>k</b>	Ir una celda más arriba

Clave	Acción
<b>m</b>	Cambiar el formato de celda actual a Markdown
<b>r</b>	Cambiar el formato de celda actual a Raw
<b>s</b>	Guardar el cuaderno
<b>v</b>	Pegar el contenido de la memoria en la celda actual
<b>x</b>	Cortar la celda o las celdas seleccionadas
<b>y</b>	Cambiar el formato de celda a Code
<b>z</b>	Deshacer
<b>Ctrl+Enter</b>	Ejecutar la celda actual e ingresar al modo de comandos
<b>Shift+Enter</b> o <b>Alt+Enter</b>	Ejecutar la celda actual, crear una nueva debajo de la salida y hacer que ingrese al modo de edición
<b>Space</b>	Página siguiente
<b>Shift+Space</b>	Página anterior
<b>Shift + L</b>	Para alternar la visibilidad de los números de línea en las celdas

## Edición de atajos del modo de comandos

El editor de cuadernos tiene una opción para personalizar los atajos de teclado del modo de comandos.

Para editar los atajos del modo de comandos

1. En el menú del editor de cuadernos, elija Command palette (Paleta de comandos).
2. En la paleta de comandos, elija el comando Edit command mode keyboard shortcuts (Editar atajos de teclado del modo de comandos).

3. Utilice la interfaz Edit command mode shortcuts (Editar atajos del modo de comandos) para asignar o reasignar los comandos que desee al teclado.

Para ver las instrucciones sobre cómo editar los atajos del modo de comandos, desplácese hasta la parte inferior de la pantalla Edit command mode shortcuts (Editar atajos del modo de comandos).

Para obtener información sobre el uso de comandos mágicos en Athena para Apache Spark, consulte [Uso de comandos mágicos](#).

## Uso de comandos mágicos

Los comandos mágicos, o magics, son comandos especiales que puede ejecutar en una celda de un cuaderno. Por ejemplo, `%env` muestra las variables de entorno de una sesión de cuaderno. Athena admite las funciones mágicas de iPython 6.0.3.

En esta sección se muestran algunos comandos mágicos de Athena para Apache Spark.

- Para ver una lista de estos comandos mágicos en Athena, ejecute el comando `%lsmagic` en una celda de un cuaderno.
- Para obtener información sobre el uso de los comandos mágicos para crear gráficos en los cuadernos de Athena, consulte [Comandos mágicos para crear gráficos de datos](#).
- Para obtener información sobre comandos mágicos adicionales, consulte [Comandos mágicos integrados](#) en la documentación de IPython.

### Note

Actualmente, el comando `%pip` genera un error cuando se ejecuta. Se trata de un problema conocido.

## Comandos mágicos de celda

Los magics que se escriben en varias líneas van precedidos de un signo de doble porcentaje (`%%`) y se denominan funciones mágicas de celda o magics de celda.

## %%sql

Este comando mágico de celda permite ejecutar instrucciones SQL directamente sin tener que decorarla con la instrucción SQL de Spark. El comando también muestra el resultado al invocar `.show()` implícitamente en el marco de datos devuelto.

```
In [1]: %%sql
        SELECT 1

Calculation started (calculation_id=dac32df7-e76b-251d-491a-603d75577bde) in (session=a6c32df6-dc5f-3390-be39-38bd204513be). Checking calculation status...

Progress: ██████████ elapsed time = 00:06s, DPU counts
100%                active/requested = 0/0

Calculation completed.
+----+
|  1 |
+----+
|  1 |
+----+
```

El comando `%%sql` trunca automáticamente las salidas de las columnas a un ancho de 20 caracteres. Esto no se puede configurar actualmente. Para evitar esta limitación, utilice la siguiente sintaxis completa y modifique los parámetros del método `show` según corresponda.

```
spark.sql("""YOUR_SQL""").show(n=number, truncate=number, vertical=bool)
```

- `nint`, opcional. El número de filas que se mostrarán.
- `truncate` – `bool` o `int`, opcional: si es `true`, trunca las cadenas de más de 20 caracteres. Si se establece en un número mayor que 1, trunca las cadenas largas hasta la longitud especificada y alinea las celdas a la derecha.
- `vertical` – `bool`, opcional. Si es `true`, imprime las filas de salida de forma vertical (una línea por valor de columna).

## Comandos mágicos de línea

Los magics que se encuentran en una sola línea van precedidos de un signo de porcentaje (%) y se denominan funciones mágicas de línea o magics de línea.

## %help

Muestra descripciones de los comandos mágicos disponibles.

```
In [6]: %help

Available Magic Commands:
Magic | Input | Description
%session_id | None | Return the session ID for the running session.
%status | None | Describes the current session and display SessionID, State,
WorkGroup, EngineVersion and StartTime
%help | None | Displays list of supported magics
%set_log_level | String | Sets the current log level to the provided log leve
ls (ERROR|INFO|WARNING etc)
%list_sessions | None | Lists the most recent sessions associated with the cu
rrent workgroup
%%sql | String | Run an SQL command against SparkSQL.
```

## %list\_sessions

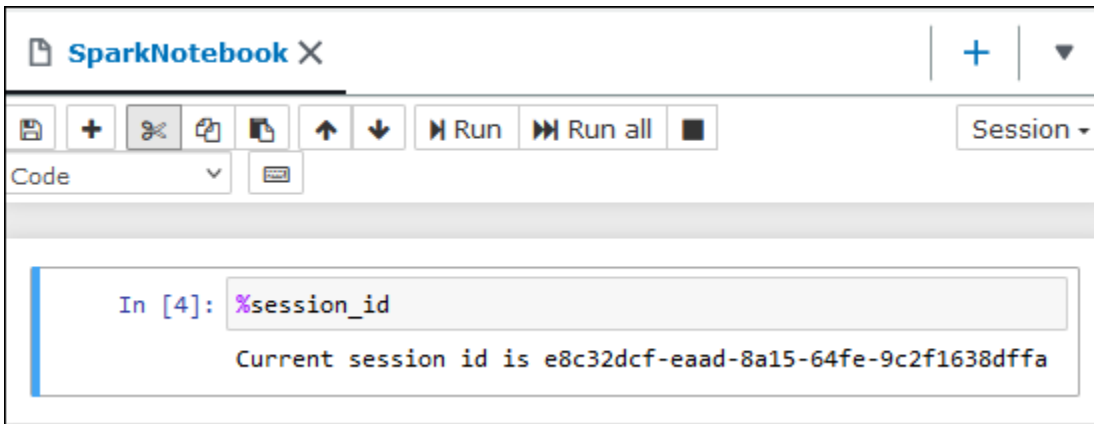
Muestra las sesiones asociadas al cuaderno. La información de cada sesión incluye el ID de la sesión, el estado de la sesión y la fecha y la hora de inicio y finalización de la sesión.

```
In [12]: %list_sessions
```

SessionId	Status	StartDateTime	EndDateTime
66c32de7-78b9-f2ee-6eb9-d8d9716c6ac8	IDLE	02/16/2023, 19:58:54	
ccc32dda-6dea-6277-d434-5c5da5e1a882	TERMINATED	02/16/2023, 19:30:24	02/16/2023, 19:51:53
e8c32dcf-eaad-8a15-64fe-9c2f1638dffa	TERMINATED	02/16/2023, 19:07:26	02/16/2023, 19:28:53

## %session\_id

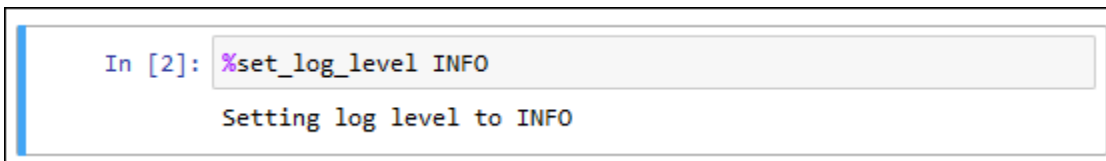
Recupera el ID de la sesión actual.



```
In [4]: %session_id
Current session id is e8c32dcf-eaad-8a15-64fe-9c2f1638dffa
```

### %set\_log\_level

Establece o restablece el registrador para utilizar el nivel de registro especificado. Los valores posibles son DEBUG, ERROR, FATAL, INFO y WARN o WARNING. Los valores deben estar en mayúsculas y no entre comillas simples o dobles.



```
In [2]: %set_log_level INFO
Setting log level to INFO
```

### %status

Describe la sesión actual. El resultado incluye el ID de la sesión, el estado de la sesión, el nombre del grupo de trabajo, la versión del motor de PySpark y la hora de inicio de la sesión. Este comando mágico requiere una sesión activa para recuperar los detalles de la sesión.

Estos son los posibles valores de estado:

CREANDO: se está iniciando la sesión, incluida la adquisición de recursos.

CREADA: se ha iniciado la sesión.

INACTIVA: la sesión puede aceptar un cálculo.

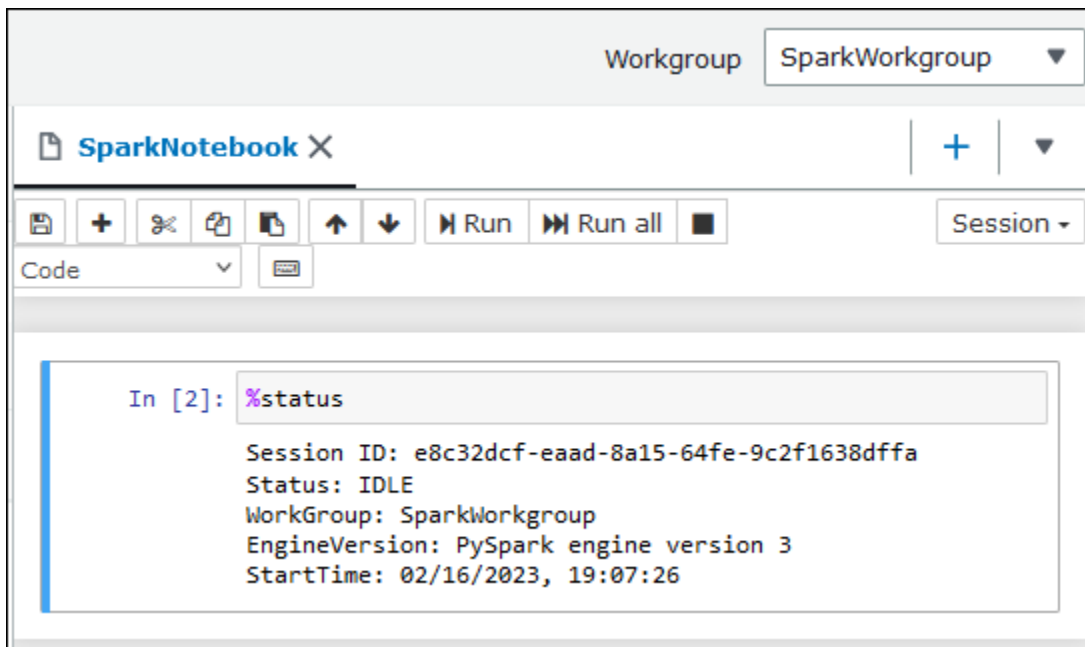
OCUPADA: la sesión está procesando otra tarea y no puede aceptar un cálculo.

FINALIZANDO: se está cerrando la sesión.

FINALIZADA: la sesión y sus recursos ya no se están ejecutando.

DEGRADADA: la sesión no tiene coordinadores en buen estado.

ERROR: debido a un error, la sesión y sus recursos ya no se están ejecutando.



## Comandos mágicos para crear gráficos de datos

Los comandos mágicos de línea de esta sección se especializan en representar datos para determinados tipos de datos o en combinación con bibliotecas de gráficos.

### %table

Puede usar el comando mágico `%table` para mostrar los datos del marco de datos en formato de tabla.

En el siguiente ejemplo, se crea un marco de datos con dos columnas y tres filas de datos y, a continuación, se muestran los datos en formato de tabla.



```
In [16]: columns = ["language","users_count"]
data = [("Java", "20000"), ("Python", "100000"), ("Scala", "3000")]
df = spark.createDataFrame(data, columns)
arr = df.collect()
%table arr
```

Calculation started (calculation\_id=12c32e0e-a76e-76e1-a108-707c09599e60) in (session=a6c32df6-dc5f-3390-be39-38bd204513be). Checking calculation status...

Progress:  elapsed time = 00:04s, DPU counts  
100% active/requested = 0/0

Calculation completed.

language	users_count
Java	20000
Python	100000
Scala	3000

`%matplotlib`

[Matplotlib](#) es una biblioteca integral para crear visualizaciones estáticas, animadas e interactivas en Python. Puede usar el comando mágico `%matplotlib` para crear un gráfico después de importar la biblioteca `matplotlib` a una celda de un cuaderno.

En el siguiente ejemplo, se importa la biblioteca `matplotlib`, se crea un conjunto de coordenadas “x” e “y”, y luego se utiliza el comando mágico `%matplotlib` para crear un gráfico de los puntos.

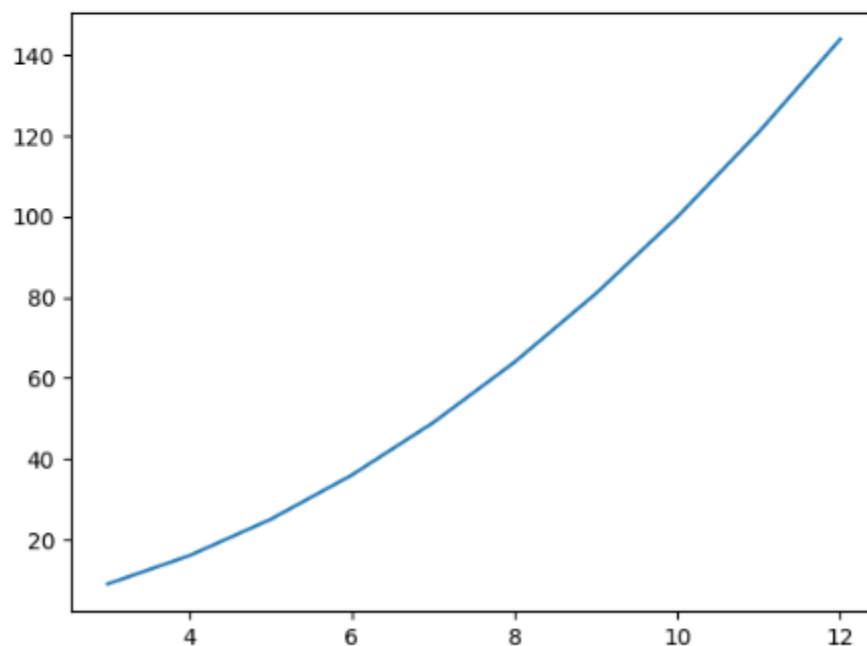
```
import matplotlib.pyplot as plt
x=[3,4,5,6,7,8,9,10,11,12]
y= [9,16,25,36,49,64,81,100,121,144]
plt.plot(x,y)
%matplotlib plt
```

```
In [12]: import matplotlib.pyplot as plt
x=[3,4,5,6,7,8,9,10,11,12]
y= [9,16,25,36,49,64,81,100,121,144]
plt.plot(x,y)
%matplotlib plt
```

Calculation started (calculation\_id=5ac32e04-81b6-9ee7-ce55-539ee2ce383e) in (session=a6c32df6-dc5f-3390-be39-38bd204513be). Checking calculation status...

Progress:  elapsed time =  
100% 00:02s

Calculation completed.



[<matplotlib.lines.Line2D object at 0x7f6e29e580>]

## Uso conjunto de las bibliotecas matplotlib y seaborn

[Seaborn](#) es una biblioteca para hacer gráficos estadísticos en Python. Se basa en matplotlib y se integra perfectamente con las estructuras de datos de [pandas](#) (análisis de datos de Python). También se puede usar el comando mágico `%matplotlib` para representar los datos de seaborn.

En el siguiente ejemplo, se utilizan las bibliotecas matplotlib y seaborn para crear un gráfico de barras sencillo.

```
import matplotlib.pyplot as plt
import seaborn as sns

x = ['A', 'B', 'C']
y = [1, 5, 3]

sns.barplot(x, y)
%matplotlib plt
```

```
In [1]: import matplotlib.pyplot as plt
import seaborn as sns

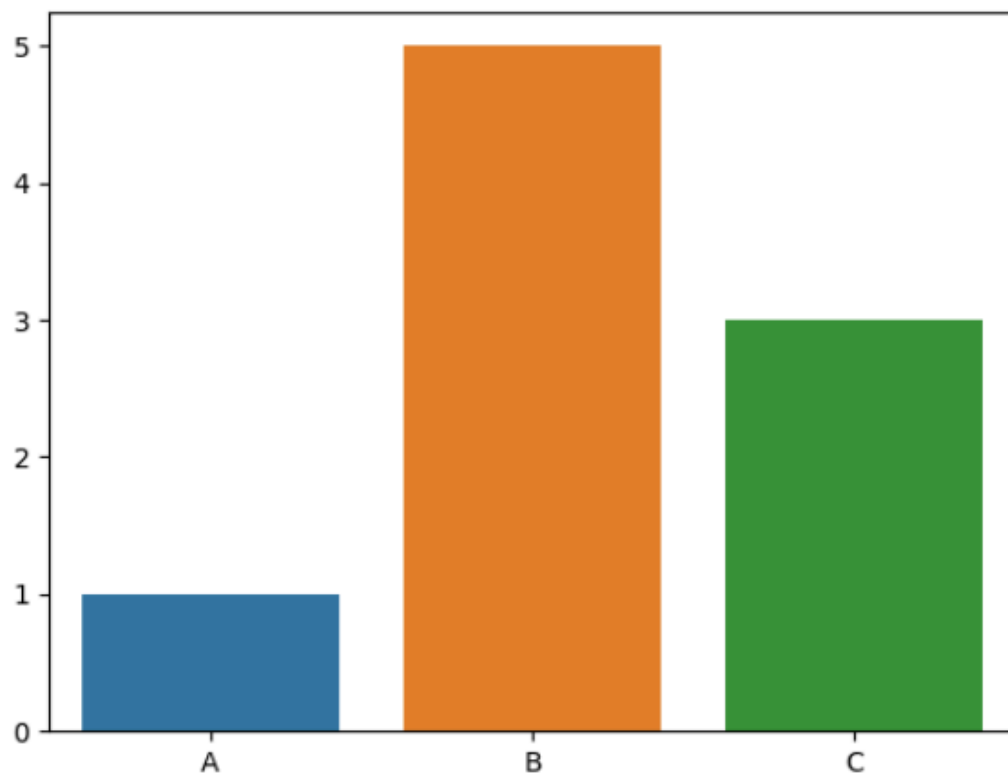
x = ['A', 'B', 'C']
y = [1, 5, 3]

sns.barplot(x, y)
%matplotlib plt
```

Calculation started (calculation\_id=08c32e1b-233b-4a72-6571-1ae7a28a7b78) in (session=64c32e1a-f45e-1d52-b54b-85202e2a9233). Checking calculation status...

Progress: 100%  elapsed time = 00:04s

Calculation completed.



## %plotly

[Plotly](#) es una biblioteca de gráficos de código abierto para Python que se puede usar para crear gráficos interactivos. El comando mágico %ploty se utiliza para representar datos de plotly.

En el siguiente ejemplo, se utilizan las bibliotecas [StringIO](#), plotly y pandas con los datos de precios de las acciones para crear un gráfico de la actividad bursátil de febrero y marzo de 2015.

```
from io import StringIO
csvString = """
Date,AAPL.Open,AAPL.High,AAPL.Low,AAPL.Close,AAPL.Volume,AAPL.Adjusted,dn,mavg,up,direction
2015-02-17,127.489998,128.880005,126.919998,127.830002,63152400,122.905254,106.7410523,117.9276
2015-02-18,127.629997,128.779999,127.449997,128.720001,44891700,123.760965,107.842423,118.94033
2015-02-19,128.479996,129.029999,128.330002,128.449997,37362400,123.501363,108.8942449,119.8891
2015-02-20,128.619995,129.5,128.050003,129.5,48948400,124.510914,109.7854494,120.7635001,131.74
2015-02-23,130.020004,133,129.660004,133,70974100,127.876074,110.3725162,121.7201668,133.067817
2015-02-24,132.940002,133.600006,131.169998,132.169998,69228100,127.078049,111.0948689,122.6648
2015-02-25,131.559998,131.600006,128.149994,128.789993,74711700,123.828261,113.2119183,123.6296
2015-02-26,128.789993,130.869995,126.610001,130.419998,91287500,125.395469,114.1652991,124.2823
2015-02-27,130,130.570007,128.240005,128.460007,62014800,123.510987,114.9668484,124.8426669,134
2015-03-02,129.25,130.279999,128.300003,129.089996,48096700,124.116706,115.8770904,125.4036668,
2015-03-03,128.960007,129.520004,128.089996,129.360001,37816300,124.376308,116.9535132,125.9551
2015-03-04,129.100006,129.559998,128.320007,128.539993,31666300,123.587892,118.0874253,126.4730
2015-03-05,128.580002,128.75,125.760002,126.410004,56517100,121.539962,119.1048311,126.848667,1
2015-03-06,128.399994,129.369995,126.260002,126.599998,72842100,121.722637,120.190797,127.22883
2015-03-09,127.959999,129.570007,125.059998,127.139999,88528500,122.241834,121.6289771,127.6311
2015-03-10,126.410004,127.220001,123.800003,124.510002,68856600,119.71316,123.1164763,127.92350
"""
csvStringIO = StringIO(csvString)

from io import StringIO
import plotly.graph_objects as go
import pandas as pd
from datetime import datetime
df = pd.read_csv(csvStringIO)
fig = go.Figure(data=[go.Candlestick(x=df['Date'],
open=df['AAPL.Open'],
high=df['AAPL.High'],
low=df['AAPL.Low'],
close=df['AAPL.Close'])])
%plotly fig
```



## Administración de los archivos de cuaderno

Además de utilizar el explorador de cuadernos para [crear](#) y [abrir](#) cuadernos, también puede utilizarlo para renombrarlos, eliminarlos, exportarlos o importarlos, o para ver el historial de sesiones de un cuaderno.

Para cambiar el nombre de un cuaderno

1. [Finalice](#) las sesiones activas del cuaderno cuyo nombre desee cambiar. Las sesiones activas del cuaderno deben finalizarse para poder cambiarle el nombre.
2. Abra el explorador de cuadernos.

3. En la lista Notebooks (Cuadernos), seleccione el botón de opción del cuaderno al que desee cambiar el nombre.
4. En el menú Actions (Acciones), elija Rename (Cambiar el nombre).
5. En la línea de comandos Rename notebook (Cambiar el nombre del cuaderno), introduzca el nuevo nombre y, a continuación, seleccione Save (Guardar). El nuevo nombre del cuaderno aparece en la lista de cuadernos.

#### Para eliminar un cuaderno

1. [Finalice](#) todas las sesiones activas del cuaderno que desee eliminar. Las sesiones activas del cuaderno deben finalizarse antes de poder eliminarlo.
2. Abra el explorador de cuadernos.
3. En la lista Notebooks (Cuadernos), seleccione el botón de opción del cuaderno que desee eliminar.
4. En el menú Actions (Acciones), elija Delete (Eliminar).
5. En la pregunta Delete notebook? (¿Eliminar cuaderno?), introduzca el nombre del cuaderno y, a continuación, elija Delete (Eliminar) para confirmar la eliminación. El nombre del cuaderno se elimina de la lista de cuadernos.

#### Para exportar un cuaderno

1. Abra el explorador de cuadernos.
2. En la lista Notebooks (Cuadernos), seleccione el botón de opción del cuaderno que desee exportar.
3. En el menú Actions (Acciones), elija Export file (Exportar archivo).

#### Para importar un cuaderno

1. Abra el explorador de cuadernos.
2. Seleccione Import File (Importar archivo).
3. Busque la ubicación del archivo que desee importar en su equipo local y, a continuación, seleccione Open (Abrir). El cuaderno importado aparece en la lista de cuadernos.

Para ver el historial de sesiones de un cuaderno

1. Abra el explorador de cuadernos.
2. En la lista Notebooks (Cuadernos), seleccione el botón de opción del cuaderno cuyo historial de sesiones desee ver.
3. En el menú Actions (Acciones), seleccione Session history (Historial de sesiones).
4. En la pestaña History (Historial), elija un ID de sesión para ver información sobre la sesión y sus cálculos.

## Uso de formatos de tabla que no son de Hive en Amazon Athena para Apache Spark

Cuando trabaja con sesiones y cuadernos en Athena para Spark, puede usar las tablas de Linux Foundation Delta Lake, Apache Hudi y Apache Iceberg, además de las tablas de Apache Hive.

### Consideraciones y limitaciones

Cuando utilice formatos de tabla que no sean de Apache Hive con Athena para Spark, tenga en cuenta los siguientes puntos:

- Además de Apache Hive, solo se admite un formato de tabla por cuaderno. Si desea usar varios formatos de tabla en Athena para Spark, cree un cuaderno independiente para cada formato de tabla. Para obtener información sobre la creación de cuadernos en Athena para Spark, consulte [Creación de un cuaderno propio](#).
- Los formatos de tabla de Delta Lake, Hudi e Iceberg se probaron en Athena para Spark utilizando AWS Glue como almacén de metadatos. Es posible que pueda usar otros almacenes de metadatos, pero actualmente no se admite ese uso.
- Para usar los formatos de tabla adicionales, anule la propiedad `spark_catalogpredeterminada`, tal y como se indica en la consola de Athena y en esta documentación. Estos catálogos que no son de Hive pueden leer tablas de Hive, además de sus propios formatos de tabla.

### Versiones de tablas

En la siguiente tabla, se muestran las versiones de tablas que no son de Hive y que se admiten en Amazon Athena para Apache Spark.



Formato de tabla	Versión compatible
Apache Iceberg	1.2.1
Apache Hudi	0,13
Linux Foundation Delta Lake	2.0.2

En Athena para Spark, estos archivos .jar con formato de tabla y sus dependencias se cargan en la ruta de clases de los controladores y ejecutores de Spark.

## Temas

- [Apache Iceberg](#)
- [Apache Hudi](#)
- [Linux Foundation Delta Lake](#)

## Apache Iceberg

[Apache Iceberg](#) es un formato de tabla de código abierto para conjuntos de datos de gran tamaño en Amazon Simple Storage Service (Amazon S3). Ofrece un rendimiento rápido de consultas en tablas grandes, confirmaciones atómicas, escrituras simultáneas y una evolución de tablas compatible con SQL.

Para usar las tablas de Apache Iceberg en Athena para Spark, configure las siguientes propiedades de Spark. Estas propiedades se configuran de forma predeterminada en la consola Athena para Spark cuando elige Apache Iceberg como formato de tabla. Para ver los pasos, consulte [Edición de detalles de la sesión](#) o [Creación de un cuaderno propio](#).

```
"spark.sql.catalog.spark_catalog": "org.apache.iceberg.spark.SparkSessionCatalog",
"spark.sql.catalog.spark_catalog.catalog-impl":
  "org.apache.iceberg.aws.glue.GlueCatalog",
"spark.sql.catalog.spark_catalog.io-impl": "org.apache.iceberg.aws.s3.S3FileIO",
"spark.sql.extensions":
  "org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions"
```

En el siguiente procedimiento, se muestra cómo utilizar una tabla de Apache Iceberg en un cuaderno Athena para Spark. Ejecute cada paso en una nueva celda del cuaderno.

## Para usar una tabla de Apache Iceberg en Athena para Spark

1. Defina las constantes que se utilizarán en el cuaderno.

```
DB_NAME = "NEW_DB_NAME"  
TABLE_NAME = "NEW_TABLE_NAME"  
TABLE_S3_LOCATION = "s3://DOC-EXAMPLE-BUCKET"
```

2. Cree un [DataFrame](#) de Apache Spark.

```
columns = ["language", "users_count"]  
data = [("Golang", 3000)]  
df = spark.createDataFrame(data, columns)
```

3. Cree una base de datos.

```
spark.sql("CREATE DATABASE {} LOCATION '{}'.format(DB_NAME, TABLE_S3_LOCATION))
```

4. Cree una tabla de Apache Iceberg vacía.

```
spark.sql("""  
CREATE TABLE {}.{} (  
language string,  
users_count int  
) USING ICEBERG  
""").format(DB_NAME, TABLE_NAME)
```

5. Inserte una fila de datos en la tabla.

```
spark.sql("""INSERT INTO {}.{} VALUES ('Golang',  
3000)""").format(DB_NAME, TABLE_NAME)
```

6. Confirme que puede consultar la nueva tabla.

```
spark.sql("SELECT * FROM {}.{}".format(DB_NAME, TABLE_NAME)).show()
```

Para obtener más información y ejemplos sobre cómo trabajar con DataFrames de Spark y tablas de Iceberg, vea [Consultas de Spark](#) en la documentación de Apache Iceberg.

## Apache Hudi

[Apache Hudi](#) es un marco de administración de datos de código abierto que simplifica el procesamiento incremental de datos. Las acciones de inserción, actualización, modificación y eliminación a nivel de registro se procesan con mayor precisión, lo que reduce la sobrecarga.

Para usar las tablas de Apache Hudi en Athena para Spark, configure las siguientes propiedades de Spark. Estas propiedades se configuran de forma predeterminada en la consola Athena para Spark cuando elige Apache Hudi como formato de tabla. Para ver los pasos, consulte [Edición de detalles de la sesión](#) o [Creación de un cuaderno propio](#).

```
"spark.sql.catalog.spark_catalog": "org.apache.spark.sql.hudi.catalog.HoodieCatalog",  
"spark.serializer": "org.apache.spark.serializer.KryoSerializer",  
"spark.sql.extensions": "org.apache.spark.sql.hudi.HoodieSparkSessionExtension"
```

En el siguiente procedimiento, se muestra cómo utilizar una tabla de Apache Hudi en un cuaderno de Athena para Spark. Ejecute cada paso en una nueva celda del cuaderno.

Para usar una tabla de Apache Hudi en Athena para Spark

1. Defina las constantes que se utilizarán en el cuaderno.

```
DB_NAME = "NEW_DB_NAME"  
TABLE_NAME = "NEW_TABLE_NAME"  
TABLE_S3_LOCATION = "s3://DOC-EXAMPLE-BUCKET"
```

2. Cree un [DataFrame](#) de Apache Spark.

```
columns = ["language", "users_count"]  
data = [("Golang", 3000)]  
df = spark.createDataFrame(data, columns)
```

3. Cree una base de datos.

```
spark.sql("CREATE DATABASE {} LOCATION '{}'.format(DB_NAME, TABLE_S3_LOCATION))
```

4. Cree una tabla de Apache Hudi vacía.

```
spark.sql("""  
CREATE TABLE {}.{} (  
language string,
```

```
users_count int
) USING HUDI
TBLPROPERTIES (
  primaryKey = 'language',
  type = 'mor'
);
""".format(DB_NAME, TABLE_NAME))
```

5. Inserte una fila de datos en la tabla.

```
spark.sql("""INSERT INTO {}.{} VALUES ('Golang',
3000)""".format(DB_NAME, TABLE_NAME))
```

6. Confirme que puede consultar la nueva tabla.

```
spark.sql("SELECT * FROM {}.{}".format(DB_NAME, TABLE_NAME)).show()
```

## Linux Foundation Delta Lake

[Linux Foundation Delta Lake](#) es un formato de tabla que se puede utilizar para el análisis de macrodatos. Puede usar Athena para Spark para leer las tablas de Delta Lake almacenadas en Amazon S3.

Para usar las tablas de Delta Lake en Athena para Spark, configure las siguientes propiedades de Spark. Estas propiedades se configuran de forma predeterminada en la consola Athena para Spark cuando elige Delta Lake como formato de tabla. Para ver los pasos, consulte [Edición de detalles de la sesión](#) o [Creación de un cuaderno propio](#).

```
"spark.sql.catalog.spark_catalog" : "org.apache.spark.sql.delta.catalog.DeltaCatalog",
"spark.sql.extensions" : "io.delta.sql.DeltaSparkSessionExtension"
```

En el siguiente procedimiento, se muestra cómo utilizar una tabla de Delta Lake en un cuaderno de Athena para Spark. Ejecute cada paso en una nueva celda del cuaderno.

Para usar una tabla de Delta Lake en Athena para Spark

1. Defina las constantes que se utilizarán en el cuaderno.

```
DB_NAME = "NEW_DB_NAME"
TABLE_NAME = "NEW_TABLE_NAME"
```

```
TABLE_S3_LOCATION = "s3://DOC-EXAMPLE-BUCKET"
```

2. Cree un [DataFrame](#) de Apache Spark.

```
columns = ["language", "users_count"]  
data = [("Golang", 3000)]  
df = spark.createDataFrame(data, columns)
```

3. Cree una base de datos.

```
spark.sql("CREATE DATABASE {} LOCATION '{}'.format(DB_NAME, TABLE_S3_LOCATION))
```

4. Cree una tabla de Delta Lake vacía.

```
spark.sql("""  
CREATE TABLE {}.{} (  
  language string,  
  users_count int  
) USING DELTA  
""".format(DB_NAME, TABLE_NAME))
```

5. Inserte una fila de datos en la tabla.

```
spark.sql("""INSERT INTO {}.{} VALUES ('Golang',  
3000)""").format(DB_NAME, TABLE_NAME))
```

6. Confirme que puede consultar la nueva tabla.

```
spark.sql("SELECT * FROM {}.{}".format(DB_NAME, TABLE_NAME)).show()
```

## Compatibilidad con las bibliotecas Python en Amazon Athena para Apache Spark

En esta página, se describe la terminología utilizada y la administración del ciclo de vida de los tiempos de ejecución, las bibliotecas y los paquetes que se utilizan en Amazon Athena para Apache Spark.

## Definiciones

- Amazon Athena para Apache Spark es una versión personalizada de Apache Spark de código abierto. Para ver la versión actual, ejecute el comando `print(f' {spark.version}')` en una celda de un cuaderno.
- El tiempo de ejecución de Athena es el entorno en el que se ejecuta su código. El entorno incluye un intérprete de Python y bibliotecas de PySpark.
- Una biblioteca o paquete externo es una biblioteca JAR o Python de Java o Scala que no forma parte del tiempo de ejecución de Athena, pero que se puede incluir en los trabajos de Athena para Spark. Tanto usted como Amazon pueden crear los paquetes externos.
- Un paquete de conveniencia es un conjunto de paquetes externos seleccionados por Athena que puede elegir incluir en sus aplicaciones de Spark.
- Una agrupación combina el tiempo de ejecución de Athena y un paquete de conveniencia.
- Una biblioteca de usuario es una biblioteca o paquete externo que agrega explícitamente a su trabajo de Athena para Spark.
  - Una biblioteca de usuario es un paquete externo que no forma parte de un paquete de conveniencia. Es necesario cargar e instalar una biblioteca de usuario, como cuando escribe algunos archivos `.py`, los comprime y, a continuación, agrega los archivos `.zip` a la aplicación.
- Una aplicación de Athena para Spark es un trabajo o una consulta que se envía a Athena para Spark.

## Administración del ciclo de vida

### Control de versiones y obsolescencia del tiempo de ejecución

El componente principal del tiempo de ejecución de Athena es el intérprete de Python. Dado que Python es un lenguaje en evolución, se lanzan nuevas versiones con regularidad y se elimina el soporte para las versiones anteriores. Athena no recomienda ejecutar programas con versiones obsoletas del intérprete de Python y recomienda encarecidamente que utilice el último tiempo de ejecución de Athena siempre que sea posible.

La programación de obsolescencia del tiempo de ejecución de Athena es la siguiente:

1. Una vez que Athena proporcione un nuevo tiempo de ejecución, Athena seguirá admitiendo el tiempo de ejecución anterior durante 6 meses. Durante ese tiempo, Athena aplicará revisiones y actualizaciones de seguridad al tiempo de ejecución anterior.

2. Transcurridos 6 meses, Athena dejará de ofrecer soporte para el tiempo de ejecución anterior. Athena ya no aplicará revisiones de seguridad ni otras actualizaciones al tiempo de ejecución anterior. Las aplicaciones de Spark que utilicen el tiempo de ejecución anterior ya no podrán recibir soporte técnico.
3. Transcurridos 12 meses, ya no podrá actualizar ni editar las aplicaciones de Spark en un grupo de trabajo que utilice el tiempo de ejecución anterior. Le recomendamos actualizar sus aplicaciones de Spark antes de finalizar este periodo. Una vez finalizado el periodo de tiempo, podrá seguir ejecutando los cuadernos existentes, pero los cuadernos que sigan utilizando el tiempo de ejecución anterior registrarán una advertencia en ese sentido.
4. Transcurridos 18 meses, ya no podrá ejecutar los trabajos en el grupo de trabajo utilizando el tiempo de ejecución anterior.

## Control de versiones y obsolescencia de los paquetes de conveniencia

El contenido de los paquetes de conveniencia cambia con el tiempo. Athena ocasionalmente agrega, elimina o actualiza estos paquetes de conveniencia.

Athena utiliza las siguientes pautas para los paquetes de conveniencia:

- Los paquetes de conveniencia tienen un esquema de control de versiones simple, como 1, 2, 3.
- Cada versión del paquete de conveniencia incluye versiones específicas de paquetes externos. Una vez que Athena crea un paquete de conveniencia, el conjunto de paquetes externos del paquete de conveniencia y sus versiones correspondientes no cambian.
- Athena crea una nueva versión del paquete de conveniencia cuando incluye un nuevo paquete externo, elimina un paquete externo o actualiza la versión de uno o más paquetes externos.

Athena descarta un paquete de conveniencia cuando descarta el tiempo de ejecución de Athena que utiliza el paquete. Athena puede descartar los paquetes antes para limitar la cantidad de paquetes que admite.

La programación de obsolescencia de los paquetes de conveniencia sigue la programación de obsolescencia del tiempo de ejecución de Athena.

## Lista de bibliotecas de Python preinstaladas

Las bibliotecas de Python preinstaladas incluyen lo siguiente.

```
boto3==1.24.31
botocore==1.27.31
certifi==2022.6.15
charset-normalizer==2.1.0
cyclers==0.11.0
cython==0.29.30
docutils==0.19
fonttools==4.34.4
idna==3.3
jmespath==1.0.1
joblib==1.1.0
kiwisolver==1.4.4
matplotlib==3.5.2
mpmath==1.2.1
numpy==1.23.1
packaging==21.3
pandas==1.4.3
patsy==0.5.2
pillow==9.2.0
plotly==5.9.0
pmdarima==1.8.5
pyathena==2.9.6
pyparsing==3.0.9
python-dateutil==2.8.2
pytz==2022.1
requests==2.28.1
s3transfer==0.6.0
scikit-learn==1.1.1
scipy==1.8.1
seaborn==0.11.2
six==1.16.0
statsmodels==0.13.2
sympy==1.10.1
tenacity==8.0.1
threadpoolctl==3.1.0
urllib3==1.26.10
pyarrow==9.0.0
```

## Notas

- No se admiten MLlib (biblioteca de machine learning de Apache Spark) y el paquete `pyspark.ml`.
- Actualmente, `pip install` no es compatible con Athena para las sesiones de Spark.



Para de obtener más información sobre cómo importar bibliotecas de Python a Amazon Athena para Apache Spark, consulte. [Importación de archivos y bibliotecas de Python a Amazon Athena para Apache Spark](#)

## Importación de archivos y bibliotecas de Python a Amazon Athena para Apache Spark

En este documento, se proporcionan ejemplos de cómo importar archivos y bibliotecas de Python a Amazon Athena para Apache Spark.

### Condiciones y limitaciones

- Versión de Python: actualmente, Athena para Spark usa la versión 3.9.16 de Python. Tenga en cuenta que los paquetes de Python son sensibles a las versiones inferiores de Python.
- Athena para la arquitectura Spark: Athena para Spark utiliza Amazon Linux 2 en la arquitectura ARM64. Tenga en cuenta que algunas bibliotecas de Python no distribuyen binarios para esta arquitectura.
- Objetos binarios compartidos (SO): dado que el método [addPyFile](#) de SparkContext no detecta objetos binarios compartidos, no se puede usar en Athena para que Spark agregue paquetes de Python que dependen de objetos compartidos.
- Conjuntos de datos distribuidos resilientes (RDD): no se admiten los [RDD](#).
- Dataframe.foreach: no se admite el método [DataFrame.foreach](#) de PySpark.

### Ejemplos

En los ejemplos, se utilizan las siguientes convenciones.

- La ubicación `s3://DOC-EXAMPLE-BUCKET` de los marcadores de posición de Amazon S3. Reemplácela por la ubicación de un bucket de S3 propio.
- Todos los bloques de código que se ejecutan desde un intérprete de comandos de Unix se muestran como `directory_name $`. Por ejemplo, el comando `ls` del directorio `/tmp` y su resultado se muestran de la siguiente manera:

```
/tmp $ ls
```

#### Salida

```
file1 file2
```

- [Agregación de un archivo a un cuaderno después de escribirlo en un directorio temporal local](#)
- [Importación de un archivo desde Amazon S3](#)
- [Agregación de archivos de Python y registro de una UDF](#)
- [Importación de un archivo .zip de Python](#)
- [Importación de dos versiones de una biblioteca de Python como módulos independientes](#)
- [Importación de un archivo .zip de Python desde PyPI](#)
- [Importación de un archivo .zip de Python desde PyPI con dependencias](#)

## Importación de archivos de texto para usarlos en cálculos

En los ejemplos de esta sección, se muestra cómo importar archivos de texto para usarlos en los cálculos de sus cuadernos de Athena para Spark.

### Agregación de un archivo a un cuaderno después de escribirlo en un directorio temporal local

En el siguiente ejemplo, se muestra cómo escribir un archivo en un directorio temporal local, agregarlo a un cuaderno y probarlo.

```
import os
from pyspark import SparkFiles
tempdir = '/tmp/'
path = os.path.join(tempdir, "test.txt")
with open(path, "w") as testFile:
    _ = testFile.write("5")
sc.addFile(path)

def func(iterator):
    with open(SparkFiles.get("test.txt")) as testFile:
        fileVal = int(testFile.readline())
        return [x * fileVal for x in iterator]

#Test the file
from pyspark.sql.functions import udf
from pyspark.sql.functions import col
```

```
udf_with_import = udf(func)
df = spark.createDataFrame([(1, "a"), (2, "b")])
df.withColumn("col", udf_with_import(col('_2'))).show()
```

## Salida

```
Calculation completed.
+---+---+-----+
| _1| _2|    col|
+---+---+-----+
|  1| a|[aaaaa]|
|  2| b|[bbbbbb]|
+---+---+-----+
```

## Importación de un archivo desde Amazon S3

En el siguiente ejemplo, se muestra cómo importar un archivo de Amazon S3 a un cuaderno y cómo probarlo.

Para importar un archivo de Amazon S3 a un cuaderno

1. Cree un archivo llamado `test.txt` que tenga una sola línea con el valor 5.
2. Agregue el archivo a un bucket de Amazon S3. En este ejemplo, se utiliza la ubicación `s3://DOC-EXAMPLE-BUCKET`.
3. Utilice el siguiente código para importar el archivo a su cuaderno y probarlo.

```
from pyspark import SparkFiles
sc.addFile('s3://DOC-EXAMPLE-BUCKET/test.txt')

def func(iterator):
    with open(SparkFiles.get("test.txt")) as testFile:
        fileVal = int(testFile.readline())
        return [x * fileVal for x in iterator]

#Test the file
from pyspark.sql.functions import udf
from pyspark.sql.functions import col

udf_with_import = udf(func)
df = spark.createDataFrame([(1, "a"), (2, "b")])
```

```
df.withColumn("col", udf_with_import(col('_2'))).show()
```

## Salida

```
Calculation completed.
+---+---+-----+
| _1| _2|    col|
+---+---+-----+
|  1|  a|[aaaaa]|
|  2|  b|[bbbbbb]|
+---+---+-----+
```

## Agregación de archivos de Python

En los ejemplos de esta sección, se muestra cómo agregar archivos y bibliotecas de Python a sus cuadernos de Spark en Athena.

### Agregación de archivos de Python y registro de una UDF

En el siguiente ejemplo, se muestra cómo agregar archivos de Python desde Amazon S3 a su cuaderno y cómo registrar una UDF.

Para agregar archivos de Python a su cuaderno y registrar una UDF

1. Con una ubicación propia de Amazon S3, cree el archivo `s3://DOC-EXAMPLE-BUCKET/file1.py` con el siguiente contenido:

```
def xyz(input):
    return 'xyz - udf ' + str(input);
```

2. En la misma ubicación de S3, cree el archivo `s3://DOC-EXAMPLE-BUCKET/file2.py` con el siguiente contenido:

```
from file1 import xyz
def uvw(input):
    return 'uvw -> ' + xyz(input);
```

3. En su cuaderno de Athena para Spark, ejecute los siguientes comandos.

```
sc.addPyFile('s3://DOC-EXAMPLE-BUCKET/file1.py')
sc.addPyFile('s3://DOC-EXAMPLE-BUCKET/file2.py')
```

```
def func(iterator):
    from file2 import uvw
    return [uvw(x) for x in iterator]

from pyspark.sql.functions import udf
from pyspark.sql.functions import col

udf_with_import = udf(func)

df = spark.createDataFrame([(1, "a"), (2, "b")])

df.withColumn("col", udf_with_import(col('_2'))).show(10)
```

## Salida

```
Calculation started (calculation_id=1ec09e01-3dec-a096-00ea-57289cdb8ce7) in
(session=c8c09e00-6f20-41e5-98bd-4024913d6cee). Checking calculation status...
Calculation completed.
+---+---+-----+
| _1| _2|          col|
+---+---+-----+
| 1 |  a|[uvw -> xyz - ud... |
| 2 |  b|[uvw -> xyz - ud... |
+---+---+-----+
```

## Importación de un archivo .zip de Python

Puede utilizar los métodos `addPyFile` y `import` de Python para importar un archivo .zip de Python a su cuaderno.

### Note

Los archivos .zip que importe a Athena para Spark pueden incluir solo paquetes de Python. Por ejemplo, no se admite la inclusión de paquetes con archivos basados en C.

Para importar un archivo **.zip** de Python a su cuaderno

1. En su equipo local, en un directorio de escritorio, por ejemplo `\tmp`, cree un directorio llamado `moduletest`.

2. En el directorio `moduletest`, cree un archivo denominado `hello.py` con el contenido siguiente:

```
def hi(input):
    return 'hi ' + str(input);
```

3. En el mismo directorio, agregue un archivo vacío con el nombre `__init__.py`.

Si saca un listado del contenido del directorio, ahora debería tener el siguiente aspecto.

```
/tmp $ ls moduletest
__init__.py      hello.py
```

4. Utilice el comando `zip` para agregar los dos archivos del módulo en un archivo llamado `moduletest.zip`.

```
moduletest $ zip -r9 ../moduletest.zip *
```

5. Cargue el archivo `.zip` en el bucket de Amazon S3.
6. Use el siguiente código para importar el archivo `.zip` de Python a su cuaderno.

```
sc.addPyFile('s3://DOC-EXAMPLE-BUCKET/moduletest.zip')

from moduletest.hello import hi

from pyspark.sql.functions import udf
from pyspark.sql.functions import col

hi_udf = udf(hi)

df = spark.createDataFrame([(1, "a"), (2, "b")])

df.withColumn("col", hi_udf(col('_2'))).show()
```

## Salida

```
Calculation started (calculation_id=6ec09e8c-6fe0-4547-5f1b-6b01adb2242c) in
(session=dcc09e8c-3f80-9cdc-bfc5-7effa1686b76). Checking calculation status...
Calculation completed.
+---+---+---+
| _1| _2| col|
```

```
+---+---+---+
| 1| a|hi a|
| 2| b|hi b|
+---+---+---+
```

## Importación de dos versiones de una biblioteca de Python como módulos independientes

En los siguientes ejemplos de código, se muestra cómo agregar e importar dos versiones diferentes de una biblioteca de Python desde una ubicación de Amazon S3 como dos módulos independientes. El código agrega cada uno de los archivos de la biblioteca de S3, los importa y, a continuación, imprime la versión de la biblioteca para verificar la importación.

```
sc.addPyFile('s3://DOC-EXAMPLE-BUCKET/python-third-party-libs-test/
simplejson_v3_15.zip')
sc.addPyFile('s3://DOC-EXAMPLE-BUCKET/python-third-party-libs-test/
simplejson_v3_17_6.zip')

import simplejson_v3_15
print(simplejson_v3_15.__version__)
```

### Salida

```
3.15.0
```

```
import simplejson_v3_17_6
print(simplejson_v3_17_6.__version__)
```

### Salida

```
3.17.6
```

## Importación de un archivo .zip de Python desde PyPI

En este ejemplo, se utiliza el comando `pip` para descargar un archivo .zip de Python del proyecto [bpabel/piglatin](#) desde el [Índice de paquetes de Python \(PyPI\)](#).

Para importar un archivo .zip de Python desde PyPI

1. En el escritorio local, utilice los siguientes comandos para crear un directorio llamado `testpiglatin` y crear un entorno virtual.

```
/tmp $ mkdir testpigliatin
/tmp $ cd testpigliatin
testpigliatin $ virtualenv .
```

## Salida

```
created virtual environment CPython3.9.6.final.0-64 in 410ms
creator CPython3Posix(dest=/private/tmp/testpigliatin, clear=False,
  no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle,
  via=copy, app_data_dir=/Users/user1/Library/Application Support/virtualenv)
added seed packages: pip==22.0.4, setuptools==62.1.0, wheel==0.37.1
activators
  BashActivator, CShellActivator, FishActivator, NushellActivator, PowerShellActivator, PythonAct
```

2. Cree un subdirectorio llamado `unpacked` para albergar el proyecto.

```
testpigliatin $ mkdir unpacked
```

3. Utilice el comando `pip` para instalar el proyecto en el directorio `unpacked`.

```
testpigliatin $ bin/pip install -t $PWD/unpacked piglatin
```

## Salida

```
Collecting piglatin
Using cached piglatin-1.0.6-py2.py3-none-any.whl (3.1 kB)
Installing collected packages: piglatin
Successfully installed piglatin-1.0.6
```

4. Compruebe el contenido del directorio.

```
testpigliatin $ ls
```

## Salida

```
bin lib pyvenv.cfg unpacked
```

5. Cambie al directorio `unpacked` y muestre el contenido.



```
testpigliatin $ cd unpacked
unpacked $ ls
```

### Salida

```
pigliatin piglatin-1.0.6.dist-info
```

6. Utilice el comando `zip` para insertar el contenido del proyecto `pigliatin` en un archivo llamado `library.zip`.

```
unpacked $ zip -r9 ../library.zip *
```

### Salida

```
adding: piglatin/ (stored 0%)
adding: piglatin/__init__.py (deflated 56%)
adding: piglatin/__pycache__/ (stored 0%)
adding: piglatin/__pycache__/__init__.cpython-39.pyc (deflated 31%)
adding: piglatin-1.0.6.dist-info/ (stored 0%)
adding: piglatin-1.0.6.dist-info/RECORD (deflated 39%)
adding: piglatin-1.0.6.dist-info/LICENSE (deflated 41%)
adding: piglatin-1.0.6.dist-info/WHEEL (deflated 15%)
adding: piglatin-1.0.6.dist-info/REQUESTED (stored 0%)
adding: piglatin-1.0.6.dist-info/INSTALLER (stored 0%)
adding: piglatin-1.0.6.dist-info/METADATA (deflated 48%)
```

7. (Opcional) Utilice los siguientes comandos para probar la importación localmente.
  - a. Establezca la ruta de Python en la ubicación del archivo `library.zip` e inicie Python.

```
/home $ PYTHONPATH=/tmp/testpigliatin/library.zip
/home $ python3
```

### Salida

```
Python 3.9.6 (default, Jun 29 2021, 06:20:32)
[Clang 12.0.0 (clang-1200.0.32.29)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
```

- b. Importe la biblioteca y ejecute un comando de prueba.

```
>>> import piglatin
>>> piglatin.translate('hello')
```

### Salida

```
'ello-hay'
```

8. Utilice comandos como los siguientes para agregar el archivo `.zip` desde Amazon S3, importarlo a su cuaderno de Athena y probarlo.

```
sc.addPyFile('s3://DOC-EXAMPLE-BUCKET/library.zip')

import piglatin
piglatin.translate('hello')

from pyspark.sql.functions import udf
from pyspark.sql.functions import col

hi_udf = udf(piglatin.translate)

df = spark.createDataFrame([(1, "hello"), (2, "world")])

df.withColumn("col", hi_udf(col('_2'))).show()
```

### Salida

```
Calculation started (calculation_id=e2c0a06e-f45d-d96d-9b8c-ff6a58b2a525) in
(session=82c0a06d-d60e-8c66-5d12-23bcd55a6457). Checking calculation status...
Calculation completed.
+---+-----+-----+
| _1|  _2|    col|
+---+-----+-----+
|  1|hello|ello-hay|
|  2|world|orld-way|
+---+-----+-----+
```

## Importación de un archivo .zip de Python desde PyPI con dependencias

En este ejemplo, se importa desde PyPI el paquete [md2gemini](#), que convierte el texto en Markdown al formato de texto Gemini. El paquete tiene las siguientes [dependencias](#):

```
cjkrwrap
mistune
wcwidth
```

### Para importar un archivo .zip de Python con dependencias

1. En su equipo local, utilice los siguientes comandos para crear un directorio llamado `testmd2gemini` y crear un entorno virtual.

```
/tmp $ mkdir testmd2gemini
/tmp $ cd testmd2gemini
testmd2gemini$ virtualenv .
```

2. Cree un subdirectorio llamado `unpacked` para albergar el proyecto.

```
testmd2gemini $ mkdir unpacked
```

3. Utilice el comando `pip` para instalar el proyecto en el directorio `unpacked`.

```
/testmd2gemini $ bin/pip install -t $PWD/unpacked md2gemini
```

### Salida

```
Collecting md2gemini
  Downloading md2gemini-1.9.0-py3-none-any.whl (31 kB)
Collecting wcwidth
  Downloading wcwidth-0.2.5-py2.py3-none-any.whl (30 kB)
Collecting mistune<3,>=2.0.0
  Downloading mistune-2.0.2-py2.py3-none-any.whl (24 kB)
Collecting cjkrwrap
  Downloading CJKwrap-2.2-py2.py3-none-any.whl (4.3 kB)
Installing collected packages: wcwidth, mistune, cjkrwrap, md2gemini
Successfully installed cjkrwrap-2.2 md2gemini-1.9.0 mistune-2.0.2 wcwidth-0.2.5
...
```

4. Cambie al directorio `unpacked` y compruebe el contenido.

```
testmd2gemini $ cd unpacked
unpacked $ ls -lah
```

## Salida

```
total 16
drwxr-xr-x 13 user1 wheel 416B Jun 7 18:43 .
drwxr-xr-x 8 user1 wheel 256B Jun 7 18:44 ..
drwxr-xr-x 9 user1 staff 288B Jun 7 18:43 CJKwrap-2.2.dist-info
drwxr-xr-x 3 user1 staff 96B Jun 7 18:43 __pycache__
drwxr-xr-x 3 user1 staff 96B Jun 7 18:43 bin
-rw-r--r-- 1 user1 staff 5.0K Jun 7 18:43 cjkwrap.py
drwxr-xr-x 7 user1 staff 224B Jun 7 18:43 md2gemini
drwxr-xr-x 10 user1 staff 320B Jun 7 18:43 md2gemini-1.9.0.dist-info
drwxr-xr-x 12 user1 staff 384B Jun 7 18:43 mistune
drwxr-xr-x 8 user1 staff 256B Jun 7 18:43 mistune-2.0.2.dist-info
drwxr-xr-x 16 user1 staff 512B Jun 7 18:43 tests
drwxr-xr-x 10 user1 staff 320B Jun 7 18:43 wcwidth
drwxr-xr-x 9 user1 staff 288B Jun 7 18:43 wcwidth-0.2.5.dist-info
```

5. Utilice el comando `zip` para insertar el contenido del proyecto `md2gemini` en un archivo llamado `md2gemini.zip`.

```
unpacked $ zip -r9 ../md2gemini *
```

## Salida

```
adding: CJKwrap-2.2.dist-info/ (stored 0%)
adding: CJKwrap-2.2.dist-info/RECORD (deflated 37%)
....
adding: wcwidth-0.2.5.dist-info/INSTALLER (stored 0%)
adding: wcwidth-0.2.5.dist-info/METADATA (deflated 62%)
```

6. (Opcional) Utilice los siguientes comandos para comprobar que la biblioteca funciona en su equipo local.
  - a. Establezca la ruta de Python en la ubicación del archivo `md2gemini.zip` e inicie Python.

```
/home $ PYTHONPATH=/tmp/testmd2gemini/md2gemini.zip
/home python3
```

**b. Importe la biblioteca y realice una prueba.**

```
>>> from md2gemini import md2gemini
>>> print(md2gemini('[abc](https://abc.def)'))
```

**Salida**

```
https://abc.def abc
```

**7. Utilice los siguientes comandos para agregar el archivo .zip desde Amazon S3, importarlo a su cuaderno de Athena y realizar una prueba no UDF.**

```
# (non udf test)
sc.addPyFile('s3://DOC-EXAMPLE-BUCKET/md2gemini.zip')
from md2gemini import md2gemini
print(md2gemini('[abc](https://abc.def)'))
```

**Salida**

```
Calculation started (calculation_id=0ac0a082-6c3f-5a8f-eb6e-f8e9a5f9bc44) in
(session=36c0a082-5338-3755-9f41-0cc954c55b35). Checking calculation status...
Calculation completed.
=> https://abc.def (https://abc.def/) abc
```

**8. Utilice los siguientes comandos para realizar una prueba UDF.**

```
# (udf test)

from pyspark.sql.functions import udf
from pyspark.sql.functions import col
from md2gemini import md2gemini

hi_udf = udf(md2gemini)
df = spark.createDataFrame([(1, "[first website](https://abc.def)"), (2, "[second
website](https://aws.com)")]])
df.withColumn("col", hi_udf(col('_2'))).show()
```

**Salida**

```
Calculation started (calculation_id=60c0a082-f04d-41c1-a10d-d5d365ef5157) in
(session=36c0a082-5338-3755-9f41-0cc954c55b35). Checking calculation status...
Calculation completed.
```

```
+---+-----+-----+
| _1|          _2|      col|
+---+-----+-----+
|  1|[first website](h...|=> https://abc.de...|
|  2|[second website](...|=> https://aws.co...|
+---+-----+-----+
```

## Adición de archivos JAR y configuración personalizada de Spark

Al crear o editar una sesión en Amazon Athena para Apache Spark, puede usar las [propiedades de Spark](#) a fin de especificar los archivos `.jar`, los paquetes u otra configuración personalizada para la sesión. Para especificar las propiedades de Spark, puede usar la consola, la AWS CLI o la API de Athena.

### Uso de la consola de Athena para especificar propiedades de Spark

En la consola de Athena, puede especificar sus propiedades de Spark al [crear un cuaderno](#) o [editar una sesión actual](#).

Para agregar propiedades en el cuadro de diálogo Crear cuaderno o Editar detalles de la sesión

1. Amplíe Propiedades de Spark.
2. Para agregar sus propiedades, use la opción Editar en la tabla o Editar en JSON.
  - Para la opción Editar en la tabla, seleccione Agregar propiedad si desea agregar una propiedad o Eliminar si desea eliminar una propiedad. Utilice los cuadros Clave y Valor para introducir los nombres de las propiedades y sus valores.
    - Para agregar un archivo `.jar` personalizado, utilice la propiedad `spark.jars`.
    - Utilice la propiedad `spark.jars.packages` para especificar un archivo de paquete.
  - Para introducir y editar la configuración directamente, elija la opción Editar en JSON. En el editor de texto JSON, puede llevar a cabo las siguientes tareas:
    - Seleccione Copiar para copiar el texto JSON en el portapapeles.
    - Seleccione Borrar para eliminar todo el texto del editor JSON.

- Elija el icono de ajustes (engranaje) a fin de configurar el ajuste de líneas o seleccionar un tema de color para el editor JSON.

## Notas

- Puede configurar las propiedades en Athena para Spark, que es lo mismo que configurar las [propiedades de Spark](#) directamente en un objeto [SparkConf](#).
- Inicie todas las propiedades de Spark con el prefijo `spark.`. Se ignoran las propiedades con otros prefijos.
- No todas las propiedades de Spark están disponibles para la configuración personalizada en Athena. Si envía una solicitud `StartSession` que tiene una configuración restringida, la sesión no podrá iniciarse.
  - No puede usar el prefijo `spark.athena.` porque está reservado.

## Uso de la AWS CLI o la API de Athena para proporcionar una configuración personalizada

Para utilizar la AWS CLI o la API de Athena a fin de proporcionar la configuración de la sesión, utilice la acción de la API [StartSession](#) o el comando de la CLI [start-session](#). En la solicitud `StartSession`, utilice el campo `SparkProperties` del objeto [EngineConfiguration](#) para pasar la información de configuración en formato JSON. Esto inicia una sesión con la configuración especificada. Para ver la sintaxis de la solicitud, consulte [StartSession](#) en la Referencia de la API de Amazon Athena.

## Solución de errores de inicio de sesión

Cuando se produce un error de configuración personalizada durante el inicio de una sesión, la consola de Athena para Spark muestra un mensaje de error. Para solucionar los errores de inicio de sesión, puede comprobar los cambios de estado de la sesión o la información de registro.

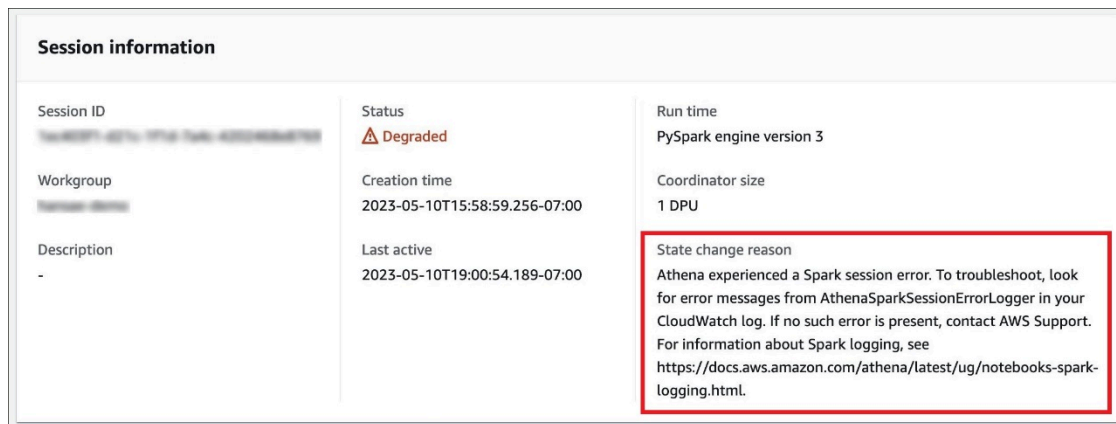
## Visualización de la información sobre los cambios de estado de la sesión

Puede obtener detalles sobre un cambio de estado de sesión en el editor de cuadernos de Athena o en la API de Athena.

## Para visualizar la información del estado de la sesión en la consola de Athena

1. En el editor de cuadernos de Athena, en el menú Sesión de la esquina superior derecha, elija Ver detalles.
2. Vea la pestaña Sesión actual. En la sección Información de la sesión, se muestra información como el identificador de la sesión, el grupo de trabajo, el estado y el motivo del cambio de estado.

En la siguiente de captura de pantalla de ejemplo, se muestra información en la sección Motivo del cambio de estado del cuadro de diálogo Información de la sesión sobre un error de sesión de Spark en Athena.



**Session information**

Session ID	Status	Run time
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx	<span style="color: red;">⚠ Degraded</span>	PySpark engine version 3
Workgroup	Creation time	Coordinator size
xxxxxxxx-xxxx	2023-05-10T15:58:59.256-07:00	1 DPU
Description	Last active	State change reason
-	2023-05-10T19:00:54.189-07:00	Athena experienced a Spark session error. To troubleshoot, look for error messages from AthenaSparkSessionErrorLogger in your CloudWatch log. If no such error is present, contact AWS Support. For information about Spark logging, see <a href="https://docs.aws.amazon.com/athena/latest/ug/notebooks-spark-logging.html">https://docs.aws.amazon.com/athena/latest/ug/notebooks-spark-logging.html</a> .

## Para visualizar la información del estado de la sesión con la API de Athena

- En la API de Athena, puede encontrar información sobre el cambio de estado de la sesión en el campo `StateChangeReason` del objeto [SessionStatus](#).

### Note

Tras detener manualmente una sesión, o si la sesión se detiene tras un tiempo de espera de inactividad (el valor predeterminado es de 20 minutos), el valor de `StateChangeReason` cambia a Se cerró la sesión por solicitud.



## Uso del registro para solucionar los errores de inicio de sesión

[Amazon CloudWatch](#) registra los errores de configuración personalizados que se producen durante el inicio de una sesión. En sus registros de CloudWatch, busque los mensajes de error de `AthenaSparkSessionErrorLogger` para solucionar un error de inicio de sesión.

Para obtener más información sobre el registro de Spark, consulte [Registro de eventos de aplicaciones de Spark en Athena](#).

Para obtener más información sobre cómo solucionar problemas con las sesiones en Athena para Spark, consulte [Solución de problemas de las sesiones](#).

## Formatos de datos y almacenamiento compatibles

En la siguiente tabla se muestran los formatos que se admiten de manera nativa en Athena para Apache Spark.

Formato de los datos	Lectura	Escritura	Compresión de escritura
parquet	yes	sí	ninguna, sin comprimir, snappy, gzip
orc	yes	sí	ninguna, snappy, zlib, lzo
json	yes	sí	bzip2, gzip, deflate
csv	yes	sí	bzip2, gzip, deflate
texto	yes	sí	ninguna, bzip2, gzip, deflate
archivo binario	yes	N/A	N/A

# Supervisión de los cálculos de Apache Spark con métricas de CloudWatch

Athena publica las métricas relacionadas con el cálculo en Amazon CloudWatch cuando se selecciona la opción [Publish CloudWatch metrics](#) para su grupo de trabajo habilitado para Spark. En la consola de CloudWatch puede crear paneles personalizados y establecer alarmas y activadores de métricas.

Athena publica las siguientes métricas en la consola de CloudWatch bajo el espacio de nombres AmazonAthenaForApacheSpark:

- `DPUCount`: número de DPU consumidas durante la sesión para ejecutar los cálculos.

Esta métrica tiene las siguientes dimensiones:

- `SessionId`: ID de la sesión en la que se envían los cálculos.
- `WorkGroup`: nombre del grupo de trabajo.

Para consultar las métricas de los grupos de trabajo habilitados para Spark en la consola de Amazon CloudWatch

1. Abra la consola de CloudWatch en <https://console.aws.amazon.com/cloudwatch/>.
2. En el panel de navegación, seleccione Métricas y, a continuación, Todas las métricas.
3. Seleccione el espacio de nombres AmazonAthenaForApacheSpark.

Para ver métricas mediante la CLI

- Realice una de las acciones siguientes:
  - Para enumerar las métricas de los grupos de trabajo habilitados para Spark de Athena, abra un símbolo del sistema y utilice el siguiente comando:

```
aws cloudwatch list-metrics --namespace "AmazonAthenaForApacheSpark"
```

- Para mostrar todas las métricas disponibles, utilice el siguiente comando:

```
aws cloudwatch list-metrics
```

## Lista de métricas y dimensiones de CloudWatch para cálculos de Apache Spark en Athena

Si habilitó métricas de CloudWatch en su grupo de trabajo de Athena habilitado para Spark, Athena envía las siguientes métricas a CloudWatch por grupo de trabajo. La métrica usa el espacio de nombres AmazonAthenaForApacheSpark.

Nombre de métrica	Descripción
DPUCount	Número de DPU (unidades de procesamiento de datos) consumidas durante la sesión para ejecutar los cálculos. Una DPU es una medida relativa de la potencia de procesamiento que consta de 4 vCPU de capacidad de cómputo y 16 GB de memoria.

Esta métrica tiene las siguientes dimensiones.

Dimensión	Descripción
SessionId	El ID de la sesión en la que se envían los cálculos.
WorkGroup	El nombre del grupo de trabajo.

## Habilitación de buckets de Amazon S3 de pagos por solicitante en Athena para Spark

Cuando un bucket de Amazon S3 se configura para pagos por solicitante, a la cuenta del usuario que ejecuta la consulta se le cobran las tarifas de acceso y transferencia de datos asociadas a la consulta. Para obtener más información, consulte [Uso de buckets de pagos por solicitante para transferencias de almacenamiento y uso](#) en la Guía del usuario de Amazon S3.

En Athena para Spark, los buckets de pagos por solicitante están habilitados por sesión, no por grupo de trabajo. En un nivel superior, habilitar los buckets de pagos por solicitante incluye los siguientes pasos:

1. En la consola de Amazon S3, habilite los pagos por solicitante en las propiedades del bucket y agregue una política de bucket para especificar el acceso.
2. En la consola de IAM, cree una política de IAM para permitir el acceso al bucket y, a continuación, adjunte la política al rol de IAM que se utilizará para acceder al bucket de pagos por solicitante.
3. En Athena para Spark, agregue una propiedad de sesión para habilitar la característica de pagos por solicitante.

## 1. Habilite los pagos por solicitante en un bucket de Amazon S3 y agregue una política de bucket.

Para habilitar los pagos por solicitante en un bucket de S3

1. Abra la consola de Amazon S3 en <https://console.aws.amazon.com/s3/>.
2. En la lista de buckets, elija el enlace del bucket para el que desea habilitar los pagos por solicitante.
3. En la página del bucket, elija la pestaña Propiedades.
4. Desplácese hacia abajo hasta la sección Pagos por solicitante y, a continuación, elija Editar.
5. En la página Editar pagos por solicitante, seleccione Habilitar y, a continuación, elija Guardar cambios.
6. Elija la pestaña Permisos.
7. Elija Editar en la sección Política de bucket.
8. En la página Editar política de bucket, aplique la política de bucket que desee para el bucket de origen. El siguiente ejemplo de política da acceso a todas las entidades principales de AWS ("AWS": "\*"), pero su acceso puede ser más detallado. Por ejemplo, puede que desee especificar solo un rol de IAM determinado en otra cuenta.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "s3:*",
```

```

        "Resource": [
            "arn:aws:s3:::account_number-us-east-1-my-s3-requester-pays-
bucket",
            "arn:aws:s3:::account_number-us-east-1-my-s3-requester-pays-bucket/
*"
        ]
    }
]
}

```

## 2. Cree una política de IAM y adjúntela al rol de IAM.

Luego, cree una política de IAM que permita el acceso al bucket. A continuación, adjunte la política al rol que se utilizará para acceder al bucket de pagos por solicitante.

Para crear una política de IAM para el bucket de pagos por solicitante y adjuntar la política a un rol

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación de la consola de IAM, elija Políticas.
3. Elija Crear política.
4. Elija JSON.
5. En el Editor de políticas, agregue una política como la que sigue:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:*"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::account_number-us-east-1-my-s3-requester-pays-
bucket",
        "arn:aws:s3:::account_number-us-east-1-my-s3-requester-pays-bucket/
*"
      ]
    }
  ]
}

```

6. Elija Siguiente.
7. En la página Revisar y crear, escriba un nombre y una descripción opcional para la política y, a continuación, elija Crear política.
8. Seleccione Roles en el panel de navegación.
9. En la página Roles, busque el rol que desea usar y, a continuación, elija el enlace del nombre del rol.
10. En la sección Políticas de permisos, elija Agregar permisos, Adjuntar políticas.
11. En la sección Otras políticas de permisos, active la casilla de verificación de la política que creó y, a continuación, elija Agregar permisos.

### 3. Agregue una propiedad de sesión de Athena para Spark.

Una vez que haya configurado el bucket de Amazon S3 y los permisos asociados para los pagos por solicitante, podrá habilitar la característica en una sesión de Athena para Spark.

Para habilitar buckets de pagos por solicitante en una sesión de Athena para Spark

1. En el editor de cuadernos, en el menú Session (Sesión) de la esquina superior derecha, elija Edit session (Editar sesión).
2. Amplíe Propiedades de Spark.
3. Seleccione Editar en JSON.
4. En el editor de texto JSON, introduzca lo siguiente:

```
{  
  "spark.hadoop.fs.s3.useRequesterPaysHeader": "true"  
}
```

5. Elija Guardar.

## Habilitación del cifrado de Apache Spark

Puede habilitar el cifrado de Apache Spark en Athena. Al hacerlo, se cifran los datos en tránsito entre los nodos de Spark y también se cifran los datos en reposo que Spark almacena de manera local. Para mejorar la seguridad de estos datos, Athena utiliza la siguiente configuración de cifrado:

```
spark.io.encryption.keySizeBits="256"
```

```
spark.io.encryption.keygen.algorithm="HmacSHA384"
```

Para habilitar el cifrado de Spark, puede usar la consola, la AWS CLI o la API de Athena.

## Uso de la consola de Athena para habilitar el cifrado de Spark

Para crear un cuaderno nuevo con el cifrado de Spark habilitado

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.
3. Realice una de las acciones siguientes:
  - En Notebook explorer (Explorador de cuadernos), elija Create notebook (Crear cuaderno).
  - En Notebook editor (Editor de cuadernos), elija Create notebook (Crear cuaderno) o elija el icono con el signo más (+) para agregar un cuaderno.
4. En Nombre del cuaderno, ingrese un nombre para el cuaderno.
5. Amplíe la opción Propiedades de Spark.
6. Seleccione Activar el cifrado de Spark.
7. Seleccione Crear.

La sesión del cuaderno que cree estará cifrada. Use el nuevo cuaderno como lo haría normalmente. Cuando más adelante inicie nuevas sesiones que usen el cuaderno, las nuevas sesiones también estarán cifradas.

También puede utilizar la consola de Athena para habilitar el cifrado de Spark en un cuaderno existente.

Para habilitar el cifrado en un cuaderno existente

1. [Abra una nueva sesión](#) para un cuaderno ya creado.
2. En el editor de cuadernos, en el menú Session (Sesión) de la esquina superior derecha, elija Edit session (Editar sesión).
3. En el cuadro de diálogo Editar detalles de la sesión, amplíe Propiedades de Spark.
4. Seleccione Activar el cifrado de Spark.
5. Elija Guardar.

La consola inicia una nueva sesión con el cifrado habilitado. Las sesiones posteriores que cree para este cuaderno también tendrán el cifrado habilitado.

## Uso de la AWS CLI para habilitar el cifrado de Spark

Puede usar la AWS CLI para habilitar el cifrado al iniciar una sesión mediante la especificación de las propiedades de Spark correspondientes.

Para usar la AWS CLI a fin de habilitar el cifrado de Spark

1. Use un comando como el siguiente para crear un objeto JSON de configuración del motor que especifique las propiedades de cifrado de Spark.

```
ENGINE_CONFIGURATION_JSON=$(
  cat <<EOF
{
  "CoordinatorDpuSize": 1,
  "MaxConcurrentDpus": 20,
  "DefaultExecutorDpuSize": 1,
  "SparkProperties": {
    "spark.authenticate": "true",
    "spark.io.encryption.enabled": "true",
    "spark.network.crypto.enabled": "true"
  }
}
EOF
)
```

2. En la AWS CLI, use el comando `athena start-session` y pase el objeto JSON que creó al argumento `--engine-configuration`, como en el siguiente ejemplo:

```
aws athena start-session \
  --region "region" \
  --work-group "your-work-group" \
  --engine-configuration "$ENGINE_CONFIGURATION_JSON"
```



## Uso de la API de Athena para habilitar el cifrado de Spark

Para habilitar el cifrado de Spark con la API de Athena, use la acción [StartSession](#) y el parámetro [EngineConfiguration](#) de `SparkProperties` a fin de especificar la configuración de cifrado en su solicitud `StartSession`.

## Configuración del acceso entre cuentas de AWS Glue en Athena para Spark

En este tema, se muestra cómo se pueden configurar la cuenta de consumidor `666666666666` y la cuenta de propietario `999999999999` para el acceso entre cuentas de AWS Glue. Cuando se configuran las cuentas, la cuenta del consumidor puede ejecutar consultas desde Athena para Spark en las bases de datos y tablas de AWS Glue del propietario.

### 1. Proporcionar acceso a los roles de consumidor en AWS Glue

En AWS Glue, el propietario crea una política que proporciona a los roles de consumidor acceso al catálogo de datos de AWS Glue del propietario.

Para agregar una política de AWS Glue que permita a un rol de consumidor acceder al catálogo de datos del propietario

1. Con la cuenta del propietario del catálogo, inicie sesión en la AWS Management Console.
2. Abra la consola de AWS Glue en <https://console.aws.amazon.com/glue/>.
3. En el panel de navegación, amplíe Catálogo de datos y, a continuación, elija Configuración del catálogo.
4. En la página Configuración del catálogo de datos, en la sección de Permisos, agregue una política como la siguiente. Esta política proporciona roles para que la cuenta de consumidor `666666666666` acceda al catálogo de datos en la cuenta de propietario `999999999999`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Cataloguers",
      "Effect": "Allow",
      "Principal": {
```

```

    "AWS": [
      "arn:aws:iam::666666666666:role/Admin",
      "arn:aws:iam::666666666666:role/AWSAthenaSparkExecutionRole"
    ]
  },
  "Action": "glue:*",
  "Resource": [
    "arn:aws:glue:us-west-2:999999999999:catalog",
    "arn:aws:glue:us-west-2:999999999999:database/*",
    "arn:aws:glue:us-west-2:999999999999:table/*"
  ]
}
]
}

```

## 2. Configurar la cuenta de consumidor para el acceso

En la cuenta de consumidor, cree una política que permita el acceso a las bases de datos, las tablas y el AWS Glue Data Catalog del propietario, y adjunte la política a un rol. En el siguiente ejemplo, se utiliza la cuenta de consumidor **666666666666**.

Para crear una política de AWS Glue a fin de acceder al AWS Glue Data Catalog

1. Con la cuenta de consumidor, inicia sesión en la AWS Management Console.
2. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
3. En el panel de navegación, amplíe Administración de acceso y, a continuación, seleccione Políticas.
4. Elija Crear política.
5. En la página Especificar permisos, seleccione JSON.
6. En el Editor de políticas, ingrese una instrucción JSON como la siguiente que permita realizar acciones de AWS Glue en el catálogo de datos de la cuenta de propietario.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "glue:*",
      "Resource": [

```

```
        "arn:aws:glue:us-east-1:999999999999:catalog",
        "arn:aws:glue:us-east-1:999999999999:database/*",
        "arn:aws:glue:us-east-1:999999999999:table/*"
    ]
}
}
```

7. Elija Siguiente.
8. En la página Revisar y crear, en Nombre de la política, ingrese un nombre para la política.
9. Elija Crear política.

Luego, utilice la consola de IAM de la cuenta de consumidor para adjuntar la política que acaba de crear al rol o los roles de IAM que utilizará la cuenta de consumidor a fin de acceder al catálogo de datos del propietario.

Para adjuntar la política de AWS Glue a los roles de la cuenta de consumidor

1. En el panel de navegación de la consola de IAM de la cuenta de consumidor, elija Roles.
2. En la página Roles, busque el rol al que desee adjuntar la política.
3. Elija Agregar permisos y luego Adjuntar políticas.
4. Busque la política que acaba de crear.
5. Seleccione la casilla de verificación de la política y, a continuación, elija Agregar permisos.
6. Repita los pasos para agregar la política a los demás roles que desee utilizar.

### 3. Configurar una sesión y crear una consulta

En Athena para Spark, en la cuenta de solicitante y con el rol especificado, cree una sesión para probar el acceso al [crear un cuaderno](#) o [editar una sesión actual](#). Al [configurar las propiedades de la sesión](#), especifique una de las siguientes opciones:

- El separador de catálogos de Glue: con este enfoque, se incluye el ID de cuenta del propietario en las consultas. Utilice este método si va a utilizar la sesión para consultar catálogos de datos de distintos propietarios.
- El ID de catálogo de Glue: con este enfoque, se consulta directamente la base de datos. Este método es más práctico si va a utilizar la sesión para consultar solo el catálogo de datos de un único propietario.

## Uso del enfoque de separador de catálogos de AWS Glue

Al editar las propiedades de la sesión, agregue lo siguiente:

```
{
  "spark.hadoop.aws.glue.catalog.separator": "/"
}
```

Al ejecutar una consulta en una celda, utilice una sintaxis como la del siguiente ejemplo. Tenga en cuenta que en la cláusula FROM se requieren el ID de catálogo y el separador antes del nombre de la base de datos.

```
df = spark.sql('SELECT requestip, uri, method, status FROM `999999999999/
mydatabase`.cloudfront_logs LIMIT 5')
df.show()
```

## Uso del enfoque de ID de catálogo de AWS Glue

Al editar las propiedades de la sesión, ingrese la siguiente propiedad. Sustituya **999999999999** por el ID de cuenta del propietario.

```
{
  "spark.hadoop.hive.metastore.glue.catalogid": "999999999999"
}
```

Al ejecutar una consulta en una celda, utilice una sintaxis como la siguiente. Tenga en cuenta que en la cláusula FROM no se requieren el ID de catálogo y el separador antes del nombre de la base de datos.

```
df = spark.sql('SELECT * FROM mydatabase.cloudfront_logs LIMIT 10')
df.show()
```

## Recursos adicionales de

[Acceso entre cuentas a los catálogos de datos de AWS Glue](#)

[Administración de los permisos entre cuentas mediante AWS Glue y Lake Formation](#) en la Guía para desarrolladores de AWS Lake Formation.

[Configurar el acceso entre cuentas a un AWS Glue Data Catalog compartido mediante Amazon Athena](#) en Patrones de recomendaciones de AWS.

## Cuotas de servicio de Amazon Athena para Apache Spark

Las cuotas de servicio, que también se denominan límites, establecen el número máximo de recursos u operaciones de servicio que puede usar en su Cuenta de AWS. Para obtener más información sobre las cuotas de otros servicios de AWS que puede utilizar con Amazon Athena para Spark, consulte [cuotas de servicio de AWS](#) en la Referencia general de Amazon Web Services.

### Note

Las nuevas Cuentas de AWS pueden tener cuotas iniciales más bajas que se pueden aumentar con el tiempo. Amazon Athena para Apache Spark supervisa el uso de la cuenta dentro de cada Región de AWS y luego aumenta automáticamente las cuotas en función de su uso. Si sus requisitos superan los límites establecidos, póngase en contacto con el servicio de atención al cliente.

En la siguiente tabla, se enumeran las cuotas de servicio de Amazon Athena para Apache Spark.

Nombre	Valor predeterminado	Ajustable	Descripción
Simultaneidad de DPU de Apache Spark	160	No	El número máximo de unidades de procesamiento de datos (DPU) que se pueden consumir de forma simultánea para los cálculos de Apache Spark de una sola cuenta en la Región de AWS actual. Una DPU es una medida relativa de la potencia de procesamiento que consta de 4 vCPU de capacidad de cómputo y 16 GB de memoria.
Simultaneidad de DPU en una sesión de Apache Spark	60	No	El número máximo de DPU que se pueden consumir de forma simultánea para un cálculo de Apache Spark en una sesión.

# API de cuadernos de Athena

La siguiente lista contiene vínculos de referencia a las acciones de la API de cuadernos de Athena. Para ver las estructuras de datos y otras acciones de la API de Athena, consulte [Amazon Athena API Reference](#) (Referencia de la API de Amazon Athena).

- [CreateNotebook](#)
- [CreatePresignedNotebookUrl](#)
- [DeleteNotebook](#)
- [ExportNotebook](#)
- [GetCalculationExecution](#)
- [GetCalculationExecutionCode](#)
- [GetCalculationExecutionStatus](#)
- [GetNotebookMetadata](#)
- [GetSession](#)
- [GetSessionStatus](#)
- [ImportNotebook](#)
- [ListApplicationDPUSizes](#)
- [ListCalculationExecutions](#)
- [ListExecutors](#)
- [ListNotebookMetadata](#)
- [ListNotebookSessions](#)
- [ListSessions](#)
- [StartCalculationExecution](#)
- [StartSession](#)
- [StopCalculationExecution](#)
- [TerminateSession](#)
- [UpdateNotebook](#)
- [UpdateNotebookMetadata](#)

# Problemas conocidos de Athena para Spark

Esta página documenta algunos de los problemas conocidos de Athena para Apache Spark.

## Excepción de argumento ilegal al crear una tabla

Aunque Spark no permite crear bases de datos con una propiedad de ubicación vacía, las bases de datos en AWS Glue pueden tener una propiedad LOCATION vacía si se crean fuera de Spark.

Si crea una tabla y especifica una base de datos de AWS Glue que tiene un campo LOCATION vacío, puede producirse una excepción como la siguiente: `IllegalArgumentException: Cannot create a path from an empty string.` (`IllegalArgumentException`: no se puede crear una ruta a partir de una cadena vacía).

Por ejemplo, el siguiente comando lanza una excepción si la base de datos predeterminada en AWS Glue contiene un campo LOCATION vacío:

```
spark.sql("create table testTable (firstName STRING)")
```

Solución sugerida A: use AWS Glue para agregar una ubicación a la base de datos que está utilizando.

Para agregar una ubicación a una base de datos de AWS Glue

1. Inicie sesión en la AWS Management Console y abra la consola de AWS Glue en <https://console.aws.amazon.com/glue/>.
2. En el panel de navegación, seleccione Databases (Bases de datos).
3. En la lista de bases de datos, elija la base de datos que desea editar.
4. En la página de detalles de la base de datos, elija Edit (Editar).
5. En la página Update a database (Actualizar una base de datos), en Location (Ubicación), ingrese una ubicación de Amazon S3.
6. Elija Update Database (Actualizar base de datos).

Solución sugerida B: use una base de datos de AWS Glue diferente que tenga una ubicación válida y existente en Amazon S3. Por ejemplo, si tiene una base de datos denominada `dbWithLocation`, use el comando `spark.sql("use dbWithLocation")` para cambiar a esa base de datos.

Solución sugerida C: cuando use Spark SQL para crear la tabla, especifique un valor para `location`, como en el siguiente ejemplo.

```
spark.sql("create table testTable (firstName STRING)
         location 's3://DOC-EXAMPLE-BUCKET/').
```

Solución sugerida D: si especificó una ubicación al crear la tabla, pero el problema persiste, asegúrese de que la ruta de Amazon S3 que proporcione tenga una barra diagonal al final. Por ejemplo, el siguiente comando lanza una excepción de argumento ilegal:

```
spark.sql("create table testTable (firstName STRING)
         location 's3://DOC-EXAMPLE-BUCKET'")
```

Para corregir este problema, agregue una barra al final de la ubicación (por ejemplo, 's3:// DOC-EXAMPLE-BUCKET/').

## Base de datos creada en una ubicación de grupo de trabajo

Si usa un comando como `spark.sql('create database db')` para crear una base de datos y no especifica una ubicación para la base de datos, Athena crea un subdirectorío en la ubicación de su grupo de trabajo y usa esa ubicación para la base de datos recién creada.

## Problemas con las tablas administradas por Hive en la base de datos de AWS Glue predeterminada

Si la propiedad `Location` de su base de datos predeterminada en AWS Glue no está vacía y especifica una ubicación válida en Amazon S3, y utiliza Athena para Spark a fin de crear una tabla administrada por Hive en su base de datos de AWS Glue predeterminada, los datos se escriben en la ubicación de Amazon S3 especificada en su grupo de trabajo de Athena para Spark en lugar de en la ubicación especificada en la base de datos de AWS Glue.

Este problema se debe a la forma en que Apache Hive gestiona su base de datos predeterminada. Apache Hive crea datos de tablas en la ubicación raíz del almacén de Hive, que puede ser diferente de la ubicación real predeterminada de la base de datos.

Cuando utiliza Athena para Spark a fin de crear una tabla administrada por Hive en la base de datos predeterminada de AWS Glue, los metadatos de la tabla de AWS Glue pueden apuntar a dos ubicaciones diferentes. Esto puede provocar un comportamiento inesperado al intentar realizar una operación `INSERT` o `DROP TABLE`.



Los pasos para reproducir el problema son los siguientes:

1. En Athena para Spark, utilice uno de los siguientes métodos para crear o guardar una tabla administrada por Hive:
  - Una instrucción SQL como `CREATE TABLE $tableName`.
  - Un comando de PySpark como `df.write.mode("overwrite").saveAsTable($tableName)` que no especifica la opción `path` en la API de Dataframe.

En este punto, es posible que la consola de AWS Glue muestre una ubicación incorrecta en Amazon S3 para la tabla.

2. En Athena para Spark, utilice la instrucción `DROP TABLE $table_name` para eliminar la tabla que ha creado.
3. Tras ejecutar la instrucción `DROP TABLE`, observará que los archivos subyacentes de Amazon S3 siguen presentes.

Para resolver este problema, siga uno de estos pasos:

Solución A: utilice una base de datos de AWS Glue diferente al crear tablas administradas por Hive.

Solución B: especifique una ubicación vacía para la base de datos predeterminada en AWS Glue. A continuación, cree las tablas administradas en la base de datos predeterminada.

## Incompatibilidad de formatos de archivos CSV y JSON entre Athena para Spark y Athena SQL

Debido a un problema conocido con el código abierto de Spark, al crear una tabla en Athena para Spark con datos CSV o JSON, es posible que la tabla no se pueda leer desde Athena SQL y viceversa.

Por ejemplo, puede crear una tabla en Athena para Spark de una de las siguientes maneras:

- Con la siguiente sintaxis `USING csv`:

```
spark.sql('''CREATE EXTERNAL TABLE $tableName (  
  $colName1 $colType1,  
  $colName2 $colType2,  
  $colName3 $colType3)
```

```
USING csv
PARTITIONED BY ($colName1)
LOCATION $s3_location''')
```

- Con la siguiente sintaxis de la API de [DataFrame](#):

```
df.write.format('csv').saveAsTable($table_name)
```

Debido a un problema conocido con el código abierto de Spark, es posible que las consultas de Athena SQL en las tablas resultantes no se realicen correctamente.

Solución sugerida: intente crear la tabla en Athena para Spark con la sintaxis de Apache Hive. Para obtener más información, consulte [CREATE HIVEFORMAT TABLE](#) (Crear tabla con formato Hive) en la documentación de Apache Spark.

## Solución de problemas de Athena para Spark

Utilice la siguiente información para solucionar los problemas que pueda tener al utilizar cuadernos y sesiones en Athena.

### Temas

- [Solución de problemas de grupos de trabajo habilitados para Spark](#)
- [Uso de la instrucción EXPLAIN de Spark para solucionar problemas de Spark SQL](#)
- [Registro de eventos de aplicaciones de Spark en Athena](#)
- [Uso de CloudTrail para solucionar problemas con las llamadas a la API de cuadernos de Athena](#)
- [Superar el límite de tamaño para bloques de código de 68 000](#)
- [Solución de problemas de las sesiones](#)
- [Solución de problemas de las tablas](#)
- [Cómo obtener asistencia](#)

## Solución de problemas de grupos de trabajo habilitados para Spark

Utilice la siguiente información para solucionar problemas de grupos de trabajo habilitados para Spark en Athena.

## La sesión deja de responder cuando se usa un rol de IAM existente

Si no se creó un nuevo rol `AWSAthenaSparkExecutionRole` para su grupo de trabajo habilitado para Spark y, en su lugar, se actualizó o eligió un rol de IAM existente, es posible que su sesión deje de responder. En este caso, puede que necesite agregar las siguientes políticas de confianza y permisos al rol de ejecución de su grupo de trabajo habilitado para Spark.

A continuación, se muestra un ejemplo de una política de confianza. La política incluye una comprobación de suplente confuso para el rol de ejecución. Sustituya los valores de `111122223333`, `aws-region` y `workgroup-name` por el ID de Cuenta de AWS, la Región de AWS y el grupo de trabajo que esté utilizando.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "athena.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:athena:aws-
region:111122223333:workgroup/workgroup-name"
        }
      }
    }
  ]
}
```

Agregue una política de permisos como la siguiente política predeterminada para grupos de trabajo habilitados para cuadernos. Modifique las ubicaciones de Amazon S3 y los ID de Cuenta de AWS de los marcadores de posición para que se correspondan con los que está utilizando. Sustituya los valores de `DOC-EXAMPLE-BUCKET`, `aws-region`, `111122223333` y `workgroup-name` por el bucket de Amazon S3, la Región de AWS, el ID de Cuenta de AWS y el grupo de trabajo que esté utilizando.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:ListBucket",
        "s3:DeleteObject",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "athena:GetWorkGroup",
        "athena:CreatePresignedNotebookUrl",
        "athena:TerminateSession",
        "athena:GetSession",
        "athena:GetSessionStatus",
        "athena:ListSessions",
        "athena:StartCalculationExecution",
        "athena:GetCalculationExecutionCode",
        "athena:StopCalculationExecution",
        "athena:ListCalculationExecutions",
        "athena:GetCalculationExecution",
        "athena:GetCalculationExecutionStatus",
        "athena:ListExecutors",
        "athena:ExportNotebook",
        "athena:UpdateNotebook"
      ],
      "Resource": "arn:aws:athena:aws-region:111122223333:workgroup/workgroup-
name"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",

```

```

        "logs:CreateLogGroup",
        "logs:PutLogEvents"
    ],
    "Resource": [
        "arn:aws:logs:aws-region:111122223333:log-group:/aws-athena:*",
        "arn:aws:logs:aws-region:111122223333:log-group:/aws-athena*:log-
stream:*"
    ]
},
{
    "Effect": "Allow",
    "Action": "logs:DescribeLogGroups",
    "Resource": "arn:aws:logs:aws-region:111122223333:log-group:*"
},
{
    "Effect": "Allow",
    "Action": [
        "cloudwatch:PutMetricData"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "cloudwatch:namespace": "AmazonAthenaForApacheSpark"
        }
    }
}
]
}

```

## Uso de la instrucción EXPLAIN de Spark para solucionar problemas de Spark SQL

Puede usar la instrucción EXPLAIN de Spark con Spark SQL para solucionar problemas de su código Spark. Los siguientes ejemplos de código y salida muestran este uso.

### Example – Instrucción SELECT de Spark

```
spark.sql("select * from select_taxi_table").explain(True)
```

### Salida

```
Calculation started (calculation_id=20c1ebd0-1ccf-ef14-db35-7c1844876a7e) in
```

```
(session=24c1ebcb-57a8-861e-1023-736f5ae55386).
Checking calculation status...

Calculation completed.
== Parsed Logical Plan ==
'Project [*]
+- 'UnresolvedRelation [select_taxi_table], [], false

== Analyzed Logical Plan ==
VendorID: bigint, passenger_count: bigint, count: bigint
Project [VendorID#202L, passenger_count#203L, count#204L]
+- SubqueryAlias spark_catalog.spark_demo_database.select_taxi_table
  +- Relation spark_demo_database.select_taxi_table[VendorID#202L,
    passenger_count#203L,count#204L] csv

== Optimized Logical Plan ==
Relation spark_demo_database.select_taxi_table[VendorID#202L,
passenger_count#203L,count#204L] csv

== Physical Plan ==
FileScan csv spark_demo_database.select_taxi_table[VendorID#202L,
passenger_count#203L,count#204L]
Batched: false, DataFilters: [], Format: CSV,
Location: InMemoryFileIndex(1 paths)
[s3://DOC-EXAMPLE-BUCKET/select_taxi],
PartitionFilters: [], PushedFilters: [],
ReadSchema: struct<VendorID:bigint,passenger_count:bigint,count:bigint>
```

## Example – Marco de datos de Spark

En el siguiente ejemplo de código, se muestra cómo utilizar EXPLAIN con un marco de datos de Spark.

```
taxi1_df=taxi_df.groupBy("VendorID", "passenger_count").count()
taxi1_df.explain("extended")
```

## Salida

```
Calculation started (calculation_id=d2c1ebd1-f9f0-db25-8477-3effc001b309) in
(session=24c1ebcb-57a8-861e-1023-736f5ae55386).
Checking calculation status...

Calculation completed.
```

```

== Parsed Logical Plan ==
'Aggregate ['VendorID, 'passenger_count],
['VendorID, 'passenger_count, count(1) AS count#321L]
+- Relation [VendorID#49L,tpep_pickup_datetime#50,tpep_dropoff_datetime#51,
passenger_count#52L,trip_distance#53,RatecodeID#54L,store_and_fwd_flag#55,
PULocationID#56L,DOLocationID#57L,payment_type#58L,fare_amount#59,
extra#60,mta_tax#61,tip_amount#62,tolls_amount#63,improvement_surcharge#64,
total_amount#65,congestion_surcharge#66,airport_fee#67] parquet

== Analyzed Logical Plan ==
VendorID: bigint, passenger_count: bigint, count: bigint
Aggregate [VendorID#49L, passenger_count#52L],
[VendorID#49L, passenger_count#52L, count(1) AS count#321L]
+- Relation [VendorID#49L,tpep_pickup_datetime#50,tpep_dropoff_datetime#51,
passenger_count#52L,trip_distance#53,RatecodeID#54L,store_and_fwd_flag#55,
PULocationID#56L,DOLocationID#57L,payment_type#58L,fare_amount#59,extra#60,
mta_tax#61,tip_amount#62,tolls_amount#63,improvement_surcharge#64,
total_amount#65,congestion_surcharge#66,airport_fee#67] parquet

== Optimized Logical Plan ==
Aggregate [VendorID#49L, passenger_count#52L],
[VendorID#49L, passenger_count#52L, count(1) AS count#321L]
+- Project [VendorID#49L, passenger_count#52L]
  +- Relation [VendorID#49L,tpep_pickup_datetime#50,tpep_dropoff_datetime#51,
passenger_count#52L,trip_distance#53,RatecodeID#54L,store_and_fwd_flag#55,
PULocationID#56L,DOLocationID#57L,payment_type#58L,fare_amount#59,extra#60,
mta_tax#61,tip_amount#62,tolls_amount#63,improvement_surcharge#64,
total_amount#65,congestion_surcharge#66,airport_fee#67] parquet

== Physical Plan ==
AdaptiveSparkPlan isFinalPlan=false
+- HashAggregate(keys=[VendorID#49L, passenger_count#52L], functions=[count(1)],
output=[VendorID#49L, passenger_count#52L, count#321L])
  +- Exchange hashpartitioning(VendorID#49L, passenger_count#52L, 1000),
ENSURE_REQUIREMENTS, [id=#531]
    +- HashAggregate(keys=[VendorID#49L, passenger_count#52L],
functions=[partial_count(1)], output=[VendorID#49L,
passenger_count#52L, count#326L])
      +- FileScan parquet [VendorID#49L,passenger_count#52L] Batched: true,
DataFilters: [], Format: Parquet,
Location: InMemoryFileIndex(1 paths)[s3://DOC-EXAMPLE-BUCKET/
notebooks/yellow_tripdata_2016-01.parquet], PartitionFilters: [],
PushedFilters: [],

```

```
ReadSchema: struct<VendorID:bigint,passenger_count:bigint>
```

## Registro de eventos de aplicaciones de Spark en Athena

El editor de cuadernos Athena permite el registro estándar de Jupyter, Spark y Python. Puede usar `df.show()` para mostrar el contenido de PySpark DataFrame o `print("Output")` para mostrar valores en la salida de la celda. Las salidas `stdout`, `stderr` y `results` de los cálculos se escriben en la ubicación del bucket de resultados de la consulta en Amazon S3.

## Registro de eventos de aplicaciones de Spark en Amazon CloudWatch

Sus sesiones de Athena también pueden introducir registros en [Amazon CloudWatch](#) en la cuenta que esté utilizando.

### Descripción de los flujos de registro y los grupos de registros

CloudWatch organiza la actividad de registro en flujos de registro y grupos de registros.

**Flujos de registro:** un flujo de registro de CloudWatch es una secuencia de eventos de registro que comparten el mismo origen. Cada fuente independiente de registros en Registros de CloudWatch constituye un flujo de registros independiente.

**Grupos de registro:** en Registros de CloudWatch, un grupo de registros es un grupo de flujos de registro que comparten la misma configuración de retención, supervisión y control de acceso.

No hay límites en el número de flujos de registros que pueden pertenecer a un grupo de registros.

En Athena, cuando inicia una sesión de cuaderno por primera vez, Athena crea un grupo de registro en CloudWatch que utiliza el nombre de su grupo de trabajo habilitado para Spark, como en el siguiente ejemplo.

```
/aws-athena/workgroup-name
```

Este grupo de registro recibe un flujo de registro por cada ejecutor de su sesión que produzca al menos un evento de registro. Un ejecutor es la unidad de cálculo más pequeña que una sesión de cuaderno puede solicitar a Athena. En CloudWatch, el nombre del flujo de registro comienza con el ID de sesión y el ID de ejecutor.

Para obtener más información sobre los flujos de registro y los grupos de registro de CloudWatch, consulte [Working with log groups and log streams](#) (Trabajo con grupos de registro y flujos de registro) en la guía del usuario de Registros de Amazon CloudWatch.



## Uso de objetos de registro estándar en Athena para Spark

En una sesión de Athena para Spark, puede utilizar los dos objetos de registro estándar globales siguientes para escribir registros en Amazon CloudWatch:

- `athena_user_logger`: envía registros únicamente a CloudWatch. Utilice este objeto cuando desee registrar la información de sus aplicaciones de Spark directamente en CloudWatch, como en el siguiente ejemplo.

```
athena_user_logger.info("CloudWatch log line.")
```

El ejemplo escribe un evento de registro en CloudWatch como el siguiente:

```
AthenaForApacheSpark: 2022-01-01 12:00:00,000 INFO builtins: CloudWatch log line.
```

- `athena_shared_logger`: envía el mismo registro tanto a CloudWatch como a AWS para fines de soporte. Puede utilizar este objeto para compartir registros con los equipos de servicio de AWS a fin de solucionar problemas, como en el siguiente ejemplo.

```
athena_shared_logger.info("Customer debug line.")  
var = [...some variable holding customer data...]  
athena_shared_logger.info(var)
```

El ejemplo registra la línea debug y el valor de la variable `var` en Registros de CloudWatch y envía una copia de cada línea a AWS Support.

### Note

Por motivos de privacidad, su código de cálculo y sus resultados no se comparten con AWS. Asegúrese de que sus llamadas a `athena_shared_logger` escriban solo la información para la que desea que sea visible AWS Support.

Los registradores proporcionados escriben eventos a través de [Apache Log4j](#) y heredan los niveles de registro de esta interfaz. Los valores posibles del nivel de registro son DEBUG, ERROR, FATAL, INFO y WARN o WARNING. Puede utilizar la función con nombre correspondiente en el registrador para generar estos valores.

**Note**

No vuelva a unir los nombres `athena_user_logger` ni `athena_shared_logger`. Al hacerlo, los objetos de registro no podrán escribir en CloudWatch durante el resto de la sesión.

**Ejemplo: registro de eventos de cuaderno en CloudWatch**

El siguiente procedimiento muestra cómo registrar los eventos de cuaderno de Athena en Registros de Amazon CloudWatch.

Para registrar los eventos de cuaderno de Athena en Registros de Amazon CloudWatch

1. Siga [Introducción a Apache Spark en Amazon Athena](#) para crear un grupo de trabajo compatible con Spark en Athena con un nombre único. Este tutorial utiliza el nombre del grupo de trabajo `athena-spark-example`.
2. Siga los pasos que se indican en [Creación de un cuaderno propio](#) para crear un cuaderno e iniciar una nueva sesión.
3. En el editor de cuadernos de Athena, en una nueva celda de cuaderno, introduzca el siguiente comando:

```
athena_user_logger.info("Hello world.")
```

4. Ejecute la celda.
5. Para recuperar el ID de sesión actual, realice una de las siguientes acciones:
  - Consulte la salida de la celda (por ejemplo, `. . . session=72c24e73-2c24-8b22-14bd-443bdcd72de4`).
  - En una celda nueva, ejecute el comando [mágico](#) `%session_id`.
6. Guarde el ID de sesión.
7. Con la misma Cuenta de AWS que está utilizando para ejecutar la sesión del cuaderno, abra la consola de CloudWatch en <https://console.aws.amazon.com/cloudwatch/>.
8. En el panel de navegación de la consola de CloudWatch, elija Log groups (Grupos de registro).
9. En la lista de grupos de registro, seleccione el grupo de registro que tenga el nombre de su grupo de trabajo de Athena habilitado para Spark, como en el siguiente ejemplo.

```
/aws-athena/athena-spark-example
```

La sección Log streams (Flujos de registro) contiene una lista de uno o más enlaces de flujos de registro para el grupo de trabajo. Cada nombre de flujo de registro contiene el ID de sesión, el ID del ejecutor y el UUID único separados por caracteres de barra diagonal.

Por ejemplo, si el ID de sesión es 5ac22d11-9fd8-ded7-6542-0412133d3177 y el ID del ejecutor es f8c22d11-9fd8-ab13-8aba-c4100bfba7e2, el nombre del flujo de registro se parece al siguiente ejemplo.

```
5ac22d11-9fd8-ded7-6542-0412133d3177/f8c22d11-9fd8-ab13-8aba-c4100bfba7e2/f012d7cb-cefd-40b1-90b9-67358f003d0b
```

10. Elija el enlace de flujo de registro de su sesión.
11. En la página Log events (Eventos de registro), consulte la columna Message (Mensaje).

El evento de registro de la celda que ejecutó es similar al siguiente:

```
AthenaForApacheSpark: 2022-01-01 12:00:00,000 INFO builtins: Hello world.
```

12. Vuelva al editor de cuadernos de Athena.
13. En una celda nueva, introduzca el siguiente código. El código registra una variable en CloudWatch:

```
x = 6  
athena_user_logger.warn(x)
```

14. Ejecute la celda.
15. Vuelva a la página Log events (Eventos de registro) de la consola de CloudWatch para ver el mismo flujo de registro.
16. El flujo de registro ahora contiene una entrada de eventos de registro con un mensaje como el siguiente:

```
AthenaForApacheSpark: 2022-01-01 12:00:00,000 WARN builtins: 6
```

## Uso de CloudTrail para solucionar problemas con las llamadas a la API de cuadernos de Athena

Para solucionar problemas con las llamadas a la API de cuadernos, puede examinar los registros de CloudTrail de Athena para buscar anomalías o descubrir las acciones iniciadas por los usuarios. Para obtener información sobre el uso de CloudTrail con Athena, consulte [Registro de las llamadas a la API de Amazon Athena con AWS CloudTrail](#).

Los siguientes son ejemplos de entradas de registro de CloudTrail para las API de cuadernos de Athena:

- [StartSession](#)
- [TerminateSession](#)
- [ImportNotebook](#)
- [UpdateNotebook](#)
- [StartCalculationExecution](#)

### StartSession

El siguiente ejemplo muestra el registro de CloudTrail de un evento [StartSession](#) de cuaderno.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID:alias",
    "arn": "arn:aws:sts::123456789012:assumed-role/Admin/alias",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "EXAMPLE_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:role/Admin",
        "accountId": "123456789012",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
```

```
        "creationDate": "2022-10-14T16:41:51Z",
        "mfaAuthenticated": "false"
    }
}
},
"eventTime": "2022-10-14T17:05:36Z",
"eventSource": "athena.amazonaws.com",
"eventName": "StartSession",
"awsRegion": "us-east-1",
"sourceIPAddress": "203.0.113.10",
"userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36",
"requestParameters": {
    "workGroup": "notebook-workgroup",
    "engineConfiguration": {
        "coordinatorDpuSize": 1,
        "maxConcurrentDpus": 20,
        "defaultExecutorDpuSize": 1,
        "additionalConfigs": {
            "NotebookId": "b8f5854b-1042-4b90-9d82-51d3c2fd5c04",
            "NotebookIframeParentUrl": "https://us-east-1.console.aws.amazon.com"
        }
    }
},
"notebookVersion": "KeplerJupyter-1.x",
"sessionIdleTimeoutInMinutes": 20,
"clientRequestToken": "d646ff46-32d2-42f0-94d1-d060ec3e5d78"
},
"responseElements": {
    "sessionId": "a2c1ebba-ad01-865f-ed2d-a142b7451f7e",
    "state": "CREATED"
},
"requestID": "d646ff46-32d2-42f0-94d1-d060ec3e5d78",
"eventID": "b58ce998-eb89-43e9-8d67-d3d8e30561c9",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management",
"tlsDetails": {
    "tlsVersion": "TLSv1.2",
    "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
    "clientProvidedHostHeader": "athena.us-east-1.amazonaws.com"
},
"sessionCredentialFromConsole": "true"
```

}

## TerminateSession

El siguiente ejemplo muestra el registro de CloudTrail de un evento [TerminateSession](#) de cuaderno.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID:alias",
    "arn": "arn:aws:sts::123456789012:assumed-role/Admin/alias",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "EXAMPLE_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:role/Admin",
        "accountId": "123456789012",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-10-14T16:41:51Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2022-10-14T17:21:03Z",
  "eventSource": "athena.amazonaws.com",
  "eventName": "TerminateSession",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.11",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36",
  "requestParameters": {
    "sessionId": "a2c1ebba-ad01-865f-ed2d-a142b7451f7e"
  },
  "responseElements": {
    "state": "TERMINATING"
  },
  "requestID": "438ea37e-b704-4cb3-9a76-391997cf42ee",
}
```

```
"eventID": "49026c5a-bf58-4cdb-86ca-978e711ad238",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management",
"tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "clientProvidedHostHeader": "athena.us-east-1.amazonaws.com"
},
"sessionCredentialFromConsole": "true"
}
```

## ImportNotebook

El siguiente ejemplo muestra el registro de CloudTrail de un evento [ImportNotebook](#) de cuaderno. Por motivos de seguridad, parte del contenido está oculto.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID:alias",
    "arn": "arn:aws:sts::123456789012:assumed-role/Admin/alias",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "EXAMPLE_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:role/Admin",
        "accountId": "123456789012",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-10-14T16:41:51Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2022-10-14T17:08:54Z",
```

```

"eventSource": "athena.amazonaws.com",
"eventName": "ImportNotebook",
"awsRegion": "us-east-1",
"sourceIPAddress": "203.0.113.12",
"userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36",
"requestParameters": {
  "workGroup": "notebook-workgroup",
  "name": "example-notebook-name",
  "payload": "HIDDEN_FOR_SECURITY_REASONS",
  "type": "IPYNB",
  "contentMD5": "HIDDEN_FOR_SECURITY_REASONS"
},
"responseElements": {
  "notebookId": "05f6225d-bdcc-4935-bc25-a8e19434652d"
},
"requestID": "813e777f-6dac-41f4-82a7-e99b7b33f319",
"eventID": "4abec837-143b-4458-9c1f-fa9fb88ab69b",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management",
"tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "clientProvidedHostHeader": "athena.us-east-1.amazonaws.com"
},
"sessionCredentialFromConsole": "true"
}

```

## UpdateNotebook

El siguiente ejemplo muestra el registro de CloudTrail de un evento [UpdateNotebook](#) de cuaderno. Por motivos de seguridad, parte del contenido está oculto.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID:AthenaExecutor-9cc1ebb2-aac5-
b1ca-8247-5d827bd8232f",

```



```
    "arn": "arn:aws:sts::123456789012:assumed-role/AWSAthenaSparkExecutionRole-om0yj71w5l/AthenaExecutor-9cc1ebb2-aac5-b1ca-8247-5d827bd8232f",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "EXAMPLE_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:role/service-role/AWSAthenaSparkExecutionRole-om0yj71w5l",
        "accountId": "123456789012",
        "userName": "AWSAthenaSparkExecutionRole-om0yj71w5l"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-10-14T16:48:06Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2022-10-14T16:52:22Z",
  "eventSource": "athena.amazonaws.com",
  "eventName": "UpdateNotebook",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.13",
  "userAgent": "Boto3/1.24.84 Python/3.8.14 Linux/4.14.225-175.364.amzn2.aarch64 Botocore/1.27.84",
  "requestParameters": {
    "notebookId": "c87553ff-e740-44b5-884f-a70e575e08b9",
    "payload": "HIDDEN_FOR_SECURITY_REASONS",
    "type": "IPYNB",
    "contentMD5": "HIDDEN_FOR_SECURITY_REASONS",
    "sessionId": "9cc1ebb2-aac5-b1ca-8247-5d827bd8232f"
  },
  "responseElements": null,
  "requestID": "baaba1d2-f73d-4df1-a82b-71501e7374f1",
  "eventID": "745cdd6f-645d-4250-8831-d0ffd2fe3847",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.2",
```

```

    "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
    "clientProvidedHostHeader": "athena.us-east-1.amazonaws.com"
  }
}

```

## StartCalculationExecution

El siguiente ejemplo muestra el registro de CloudTrail de un evento [StartCalculationExecution](#) de cuaderno. Por motivos de seguridad, parte del contenido está oculto.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLE_PRINCIPAL_ID:AthenaExecutor-9cc1ebb2-aac5-b1ca-8247-5d827bd8232f",
    "arn": "arn:aws:sts::123456789012:assumed-role/AWSAthenaSparkExecutionRole-om0yj71w5l/AthenaExecutor-9cc1ebb2-aac5-b1ca-8247-5d827bd8232f",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "EXAMPLE_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:role/service-role/AWSAthenaSparkExecutionRole-om0yj71w5l",
        "accountId": "123456789012",
        "userName": "AWSAthenaSparkExecutionRole-om0yj71w5l"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-10-14T16:48:06Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2022-10-14T16:52:37Z",
  "eventSource": "athena.amazonaws.com",
  "eventName": "StartCalculationExecution",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.14",
  "userAgent": "Boto3/1.24.84 Python/3.8.14 Linux/4.14.225-175.364.amzn2.aarch64 Botocore/1.27.84",

```

```

"requestParameters": {
  "sessionId": "9cc1ebb2-aac5-b1ca-8247-5d827bd8232f",
  "description": "Calculation started via Jupyter notebook",
  "codeBlock": "HIDDEN_FOR_SECURITY_REASONS",
  "clientRequestToken": "0111cd63-4fd0-4ad8-a738-fd350115fc21"
},
"responseElements": {
  "calculationExecutionId": "82c1ebb4-bd08-e4c3-5631-a662fb2ff2c5",
  "state": "CREATING"
},
"requestID": "1a107461-3f1b-481e-b8a2-7fbd524e2373",
"eventID": "b74dbd00-e839-4bd1-a1da-b75fbc70ab9a",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management",
"tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "clientProvidedHostHeader": "athena.us-east-1.amazonaws.com"
}
}

```

## Superar el límite de tamaño para bloques de código de 68 000

Athena para Spark tiene un límite de tamaño para bloques de código de cálculo conocido de 68 000 caracteres. Cuando ejecuta un cálculo con un bloque de código por encima de este límite, puede recibir el siguiente mensaje de error:

“...” en “CodeBlock” no cumplió con la restricción: la longitud debe ser menor o igual a 68 000

La siguiente imagen muestra este error en el editor de cuadernos de la consola de Athena.



El mismo error puede producirse cuando se utiliza la AWS CLI para ejecutar un cálculo que tiene un bloque de código grande, como en el siguiente ejemplo.

```
aws athena start-calculation-execution \  
  --session-id "{SESSION_ID}" \  
  --description "{SESSION_DESCRIPTION}" \  
  --code-block "{LARGE_CODE_BLOCK}"
```

El comando muestra el siguiente mensaje de error:

`{LARGE_CODE_BLOCK}` at 'codeBlock' failed to satisfy constraint: Member must have length less than or equal to 68000 (`{LARGE_CODE_BLOCK}` en “codeBlock” no cumplió con la restricción: la longitud del miembro debe ser menor o igual a 68 000).

## Solución

Para solucionar este problema, cargue el archivo que contiene su código de cálculo o consulta a Amazon S3. A continuación, utilice boto3 para leer el archivo y ejecutar el código o SQL.

En los siguientes ejemplos se supone que ya ha cargado el archivo que contiene la consulta SQL o el código Python en Amazon S3.

### Ejemplo de SQL

El siguiente ejemplo de código lee el archivo `large_sql_query.sql` de un bucket de Amazon S3 y luego ejecuta la consulta grande que contiene el archivo.

```
s3 = boto3.resource('s3')  
def read_s3_content(bucket_name, key):  
    response = s3.Object(bucket_name, key).get()  
    return response['Body'].read()  
  
# SQL  
sql = read_s3_content('bucket_name', 'large_sql_query.sql')  
df = spark.sql(sql)
```

### Ejemplo de PySpark

El siguiente ejemplo de código lee el archivo `large_py_spark.py` de Amazon S3 y luego ejecuta el bloque de código grande que se encuentra en el archivo.

```
s3 = boto3.resource('s3')  
  
def read_s3_content(bucket_name, key):
```

```
response = s3.Object(bucket_name, key).get()
return response['Body'].read()
```

```
# PySpark
py_spark_code = read_s3_content('bucket_name', 'large_py_spark.py')
exec(py_spark_code)
```

## Solución de problemas de las sesiones

Utilice la información de este tema para solucionar problemas de sesión.

### Sesión en estado incorrecto

Si recibe el mensaje de error `Session in unhealthy state. Please create a new session` (Sesión en estado incorrecto. Cree una nueva sesión), finalice la sesión existente y cree una nueva.

### No se pudo establecer una conexión con el servidor de cuadernos

Al abrir un cuaderno, puede aparecer el siguiente mensaje de error:

```
A connection to the notebook server could not be established.
The notebook will continue trying to reconnect.
Check your network connection or notebook server configuration.
```

### Causa

Cuando Athena abre un cuaderno, Athena crea una sesión y se conecta al cuaderno a través de una URL de cuaderno previamente firmada. La conexión al cuaderno utiliza el protocolo WSS ([WebSocket Secure](#)).

Este error puede producirse por las siguientes razones:

- Un firewall local (por ejemplo, un firewall para toda la empresa) bloquea el tráfico de WSS.
- El software antivirus o el proxy de su equipo local bloquea la conexión WSS.

### Solución

Supongamos que tiene una conexión WSS en la región `us-east-1` como la siguiente:

```
wss://94c2bcdf-66f9-4d17-9da6-7e7338060183.analytics-gateway.us-east-1.amazonaws.com/
```

```
api/kernels/33c78c82-b8d2-4631-bd22-1565dc6ec152/channels?session_id=7f96a3a048ab4917b6376895ea8d7535
```

Para resolver el error, utilice una de las siguientes estrategias.

- Utilice la sintaxis del patrón comodín para permitir listar el tráfico de WSS en el puerto 443 a través de las Regiones de AWS y las Cuentas de AWS.

```
wss://*amazonaws.com
```

- Utilice la sintaxis del patrón comodín para permitir listar el tráfico de WSS en el puerto 443 de una Región de AWS y a través de las Cuentas de AWS de la Región de AWS que especifique. El siguiente ejemplo utiliza `us-east-1`.

```
wss://*analytics-gateway.us-east-1.amazonaws.com
```

## Solución de problemas de las tablas

### No se puede crear un error de ruta al crear una tabla

Mensaje de error: `IllegalArgumentException: Cannot create a path from an empty string.` (`IllegalArgumentException: no se puede crear una ruta a partir de una cadena vacía`).

Causa: este error puede producirse cuando usa Apache Spark en Athena para crear una tabla en una base de datos de AWS Glue y la base de datos tiene una propiedad `LOCATION` vacía.

Solución sugerida: para obtener más información y soluciones, consulte [Excepción de argumento ilegal al crear una tabla](#).

### AccessDeniedException al consultar tablas de AWS Glue

Mensaje de error: `pyspark.sql.utils.AnalysisException: Unable to verify existence of default database: com.amazonaws.services.glue.model.AccessDeniedException: User: arn:aws:sts::aws-account-id:assumed-role/AWSAthenaSparkExecutionRole-unique-identifier/AthenaExecutor-unique-identifier is not authorized to perform: glue:GetDatabase on resource: arn:aws:glue:aws-region:aws-account-id:catalog because no identity-based policy allows the glue:GetDatabase action (Service: AWSGlue; Status Code: 400; Error Code: AccessDeniedException; Request ID: request-id; Proxy: null)` (`pyspark.sql.utils.AnalysisException: no se puede verificar la existencia de la base de datos predeterminada: com.amazonaws.services.glue.model.AccessDeniedException: El`

usuario: arn:aws:sts::aws-account-id:assumed-role/AWSAthenaSparkExecutionRole-unique-identifier/AthenaExecutor-unique-identifier no está autorizado a realizar: glue:GetDatabase en el recurso: arn:aws:glue:aws-region:aws-account-id:catalog porque ninguna política basada en identidades permite la acción glue:GetDatabase [Servicio: AWSGlue; Código de estado: 400; Código de error: AccessDeniedException; ID de solicitud: request-id; Proxy: null]).

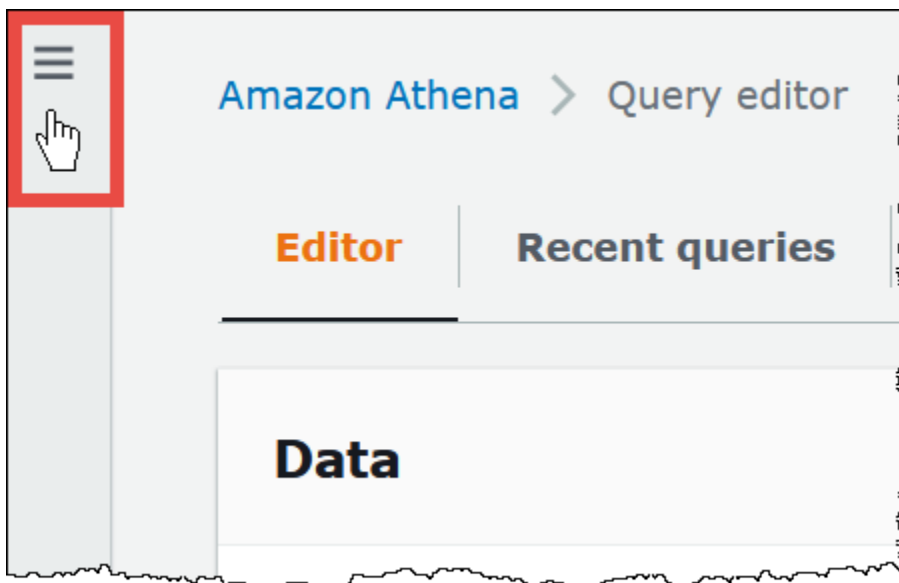
Causa: al rol de ejecución de su grupo de trabajo habilitado para Spark le faltan permisos para acceder a los recursos de AWS Glue.

Solución sugerida: para resolver este problema, conceda a su rol de ejecución acceso a los recursos de AWS Glue y, a continuación, edite la política de su bucket de Amazon S3 para conceder acceso a su rol de ejecución.

En las siguientes secciones se describen los pasos de manera más detallada.

Para conceder a su rol de ejecución acceso a los recursos de AWS Glue

1. Abra la consola de Athena en <https://console.aws.amazon.com/athena/>.
2. Si el panel de navegación de la consola no está visible, elija el menú de expansión de la izquierda.



3. En el panel de navegación de la consola de Athena, elija Grupos de trabajo.
4. En la página Grupos de trabajo, elija el enlace del grupo de trabajo que quiere ver.
5. En la página Overview Details (Detalles de información general) del grupo de trabajo, elija el enlace Role ARN (ARN del rol). El enlace abre el rol de ejecución de Spark en la consola de IAM.

6. En la sección Permissions policies (Políticas de permisos), seleccione el nombre de la política de rol vinculada.
7. Elija Edit policy (Editar política) y, a continuación, elija JSON.
8. Agregue acceso a AWS Glue al rol. Normalmente, se agregan permisos para las acciones `glue:GetDatabase` y `glue:GetTable`. Para obtener más información sobre la configuración de roles de IAM, consulte [Adición y eliminación de permisos de identidad de IAM](#) en la guía del usuario de IAM.
9. Elija Revisar política y, a continuación, elija Guardar cambios.
10. Edite la política de su bucket de Amazon S3 para conceder acceso al rol de ejecución. Tenga en cuenta que debe conceder al rol acceso tanto al bucket como a los objetos del bucket. Para conocer los pasos, consulte [Agregar una política de bucket mediante la consola de Amazon S3](#) en la guía del usuario de Amazon Simple Storage Service.

## Cómo obtener asistencia

Para obtener ayuda de AWS, seleccione Support (Soporte), Support Center (Centro de soporte) en la AWS Management Console. Para facilitar su experiencia, tenga la siguiente información disponible:

- ID de consulta de Athena
- ID de sesión
- ID de cálculo



# Notas de la versión

Describe las características, mejoras y correcciones de errores de Amazon Athena por fecha de lanzamiento.

## Temas

- [Notas de la versión de Athena para 2024](#)
- [Notas de la versión de Athena para 2023](#)
- [Notas de la versión de Athena para 2022](#)
- [Notas de la versión de Athena para 2021](#)
- [Notas de la versión de Athena para 2020](#)
- [Notas de la versión de Athena para 2019](#)
- [Notas de la versión de Athena para 2018](#)
- [Notas de la versión de Athena para 2017](#)

## Notas de la versión de Athena para 2024

### 26 de abril de 2024

Publicado el 26/04/2024

Athena lanza la versión 3.2.0 del controlador JDBC. Para obtener más información sobre esta versión del controlador, consulte [Notas de la versión de JDBC 3.x de Amazon Athena](#). Para descargar el controlador JDBC 3.x, consulte [Descarga del controlador JDBC 3.x](#).

### 24 de abril de 2024

Publicado el 24/04/2024

Athena anuncia las siguientes correcciones y mejoras.

- Parquet: Athena ahora admite lecturas compatibles con versiones anteriores en Parquet para campos primitivos repetidos y sin anotaciones que no estén incluidos en una lista o grupo de asignaciones. Este cambio evita que se devuelvan resultados incorrectos sin previo aviso y mejora los mensajes de error en caso de discrepancias en el esquema.

Para obtener más información, consulte [Support backwards compatible reads for unannotated repeated primitive fields in Parquet](#) en GitHub.com.

- Iceberg OPTIMIZE: se resolvió un problema con las consultas de OPTIMIZE que provocaban la pérdida de datos cuando se utilizaba un filtro de clave que no era de partición en una cláusula WHERE. Para obtener más información, consulte [OPTIMIZE](#).

## 16 de abril de 2024

Publicado el 16/04/2024

Utilice la nueva característica de consultas de acceso directo federadas de Amazon Athena para ejecutar consultas completas directamente en el origen de datos subyacente. Las consultas de acceso directo federadas le ayudan a aprovechar las funciones únicas, el lenguaje de consulta y las capacidades de rendimiento del origen de datos original. Por ejemplo, puede ejecutar consultas de Athena en DynamoDB con el [lenguaje PartiQL](#). Las consultas de acceso directo federadas también son útiles cuando desea ejecutar consultas SELECT que agreguen, unan o invoquen funciones del origen de datos que no están disponibles en Athena. El uso de consultas de acceso directo puede reducir la cantidad de datos que procesa Athena y reducir los tiempos de consulta.

Para obtener más información, consulte [Ejecución de consultas de acceso directo federadas](#). Para obtener la última versión de los conectores que utiliza hoy, consulte [Actualización de un conector de origen de datos](#).

## 10 de abril de 2024

Publicado el 10/04/2024

Athena anuncia las siguientes características y mejoras.

### Controlador ODBC 1.2.3.1000

Controlador ODBC 1.2.3.1000 para Athena.

Problemas resueltos:

- Problema de conexión con el servidor proxy: cuando se utilizaba un servidor proxy sin el certificado raíz, el conector no podía establecer una conexión.

Para obtener más información y descargar el controlador ODBC 1.x, las notas de la versión y la documentación, consulte [Controlador ODBC 1.x de Athena](#).

## Controlador JDBC 2.1.5

Lanzamiento del controlador JDBC 2.1.5 para Athena.

Actualizaciones y mejoras:

- Se actualizó el SDK de AWS para Java a la versión 1.12.687.
- Se actualizaron las bibliotecas Jackson para usar la versión 2.16.0.
- Se actualizaron las bibliotecas Logback para usar la versión 1.3.14.

Para obtener más información y descargar el controlador JDBC 2.x, las notas de la versión y la documentación, consulte [Controlador JDBC 2.x de Athena](#).

## 8 de abril de 2024

Publicado el 08/04/2024

Athena anuncia el controlador ODBC versión 2.0.3.0. Para obtener más información, consulte las notas de la versión [2.0.3.0](#). Para descargar el nuevo controlador ODBC v2, consulte [Descarga del controlador ODBC 2.x](#). Para obtener información sobre la conexión, consulte el [ODBC 2.x de Amazon Athena](#).

## 15 de marzo de 2024

Publicado el 18/03/2024

Amazon Athena anuncia la disponibilidad de Athena SQL en la región Oeste de Canadá (Calgary).

Para obtener una lista completa de los Servicios de AWS disponibles en cada Región de AWS, consulte [Servicios de AWS por región](#).

## 15 de febrero de 2024

Publicado el 15/02/2024

Athena lanza la versión 3.1.0 del controlador JDBC.

La versión 3.1.0 del controlador JDBC de Amazon Athena agrega compatibilidad con la autenticación integrada de Windows y la autenticación basada en formularios de Microsoft Active Directory Federation Services (AD FS). Esta versión 3.1.0 también incluye mejoras generales de rendimiento y correcciones de errores.

Para descargar el controlador JDBC v3, consulte [Descarga del controlador JDBC 3.x](#).

## 31 de enero de 2024

Publicado el 31/01/2024

Athena anuncia las siguientes características y mejoras.

- Hudi upgrade: ahora puede utilizar Athena SQL para consultar tablas Hudi 0.14.0. Para obtener más información acerca de la utilización de Athena SQL para consultar tablas Hudi, consulte [Uso de Athena para consultar conjuntos de datos de Apache Hudi](#).

## Notas de la versión de Athena para 2023

### 14 de diciembre de 2023

Publicado el 14/12/2023

Athena anuncia las siguientes correcciones y mejoras.

Athena lanza la versión 2.1.3 del controlador JDBC. El controlador resuelve los siguientes problemas:

- Se mejoró el registro para evitar conflictos con el registro de aplicaciones de Spring Boot y Gradle.
- Al utilizar el método `executeBatch()` de JDBC para insertar registros, el controlador insertó incorrectamente un solo registro. Como Athena no admite la ejecución por lotes de consultas, el controlador ahora informa de un error cuando utiliza `executeBatch()`. Para evitar esta limitación, puede enviar consultas individuales en un bucle.

Para descargar el nuevo controlador JDBC, las notas de la versión y la documentación, consulte [Controlador JDBC 2.x de Athena](#).

### 9 de diciembre de 2023

Publicado el 09/12/2023

Se lanzó el controlador ODBC 1.2.1.1000 para Athena.

Características y mejoras:

- Se actualizó la compatibilidad con RStudio: el controlador ODBC ahora es compatible con RStudio en macOS.
- Compatibilidad con un único catálogo y esquema: el conector ahora puede devolver un único catálogo y esquema. Para obtener más información, consulte la guía de instalación y configuración descargable.

Problemas resueltos:

- Instrucciones preparadas: cuando se ejecutaban instrucciones preparadas con una matriz de parámetros mediante un esquema por columnas, el conector devolvía un resultado de consulta incorrecto.
- Tamaño de columna: al seleccionar la columna `$file_modified_time` del sistema, el conector devolvía un tamaño de columna incorrecto.
- SQLPrepare: al vincular los parámetros relacionados con las consultas de SQLPrepare en SELECT, el conector devolvía un error.

Para obtener más información y descargar los nuevos controladores, las notas de la versión y la documentación, consulte [Controlador ODBC 1.x de Athena](#).

## 7 de diciembre de 2023

Publicado el 07/12/2023

Athena anuncia la versión 2.0.2.1 del controlador ODBC. Para obtener más información, consulte las notas de la versión [2.0.2.1](#). Para descargar el nuevo controlador ODBC v2, consulte [Descarga del controlador ODBC 2.x](#). Para obtener información sobre la conexión, consulte el [ODBC 2.x de Amazon Athena](#).

## 5 de diciembre de 2023

Publicado el 05/12/2023

Ahora puede crear grupos de trabajo de Athena SQL que utilicen el modo de autenticación de AWS IAM Identity Center. Estos grupos de trabajo admiten la característica de propagación de identidad

de confianza del IAM Identity Center. La propagación de identidad de confianza permite que las identidades se utilicen en los servicios de análisis de AWS, como Amazon Athena y Amazon EMR Studio.

Para obtener más información, consulte [Uso de grupos de trabajo de Athena habilitados para IAM Identity Center](#).

## 28 de noviembre de 2023

Publicado el 28/11/2023

Ahora puede consultar datos en la [clase de almacenamiento Amazon S3 Express One Zone](#) para obtener resultados de consulta rápidos. S3 Express One Zone es una clase de almacenamiento en zona de alto rendimiento y disponibilidad única, diseñada específicamente para ofrecer acceso constante a los datos en milisegundos de un solo dígito para los datos a los que accede con mayor frecuencia y las aplicaciones sensibles a la latencia. Para empezar, mueva sus datos al almacenamiento S3 Express One Zone y catalogue los datos con el [AWS Glue Data Catalog](#) para disfrutar de una experiencia de consulta sin problemas en Athena.

Para obtener más información, consulte [Consulta de datos de S3 Express One Zone](#).

## 27 de noviembre de 2023

Publicado el 27/11/2023

Athena anuncia las siguientes características y mejoras.

- **Vistas del catálogo de datos de Glue:** las vistas del catálogo de datos de Glue proporcionan una única vista común de todos los servicios de AWS, como Amazon Athena y Amazon Redshift. En las vistas del catálogo de datos de Glue, los permisos de acceso los define el usuario que creó la vista y no el usuario que consulta la vista. Estas vistas proporcionan un mayor control de acceso, ayudan a garantizar registros completos, ofrecen una mayor seguridad y pueden impedir el acceso a las tablas subyacentes.

Para obtener más información, consulte [Uso de vistas del AWS Glue Data Catalog](#).

- **Compatibilidad con CloudTrail Lake:** ahora puede usar Amazon Athena para analizar datos en [AWS CloudTrail Lake](#). AWS CloudTrail Lake es un lago de datos administrado para CloudTrail que puede utilizar para agregar, almacenar de forma inmutable y analizar los registros de actividad para investigaciones de auditoría, seguridad y operaciones. Para consultar los registros de

actividad de CloudTrail Lake desde Athena, no es necesario mover datos ni crear canalizaciones de procesamiento de datos independientes. No se requieren operaciones de ETL.

Para empezar, habilite la federación de datos en CloudTrail Lake. Al compartir los metadatos del almacén de datos de eventos de CloudTrail Lake con AWS Glue Data Catalog, CloudTrail crea los recursos necesarios de AWS Glue Data Catalog y registra los datos con AWS Lake Formation. En Lake Formation, puede especificar los usuarios y roles que puede utilizar Athena para consultar el almacén de datos de sus eventos.

Para obtener más información, consulte [Habilitar la federación de consultas de Lake](#) en la Guía del usuario de AWS CloudTrail.

## 17 de noviembre de 2023

Publicado el 17/11/2023

Athena anuncia las siguientes características y mejoras.

### Características

- **Optimizador basado en costes:** Athena anuncia la disponibilidad general de la optimización basada en costos utilizando estadísticas de AWS Glue. Para optimizar sus consultas en Athena SQL, puede solicitar que Athena recopile estadísticas a nivel de tabla o columna para sus tablas en AWS Glue. Si todas las tablas de la consulta tienen estadísticas, Athena las utiliza para examinar planes de ejecución alternativos y seleccionar el que tenga más probabilidades de ser el más rápido.

Para obtener más información, consulte [Uso del optimizador basado en costes](#).

- **Integración con Amazon EMR Studio:** ahora puede usar Athena en un Amazon EMR Studio sin tener que usar la consola Athena directamente. Con la integración de Athena en Amazon EMR, puede llevar a cabo las siguientes tareas:
  - Realizar consultas SQL de Athena
  - Visualización de los resultados de la consulta
  - Visualizar el historial de consultas
  - Visualizar las consultas guardadas
  - Realizar consultas parametrizadas
  - Ver bases de datos, tablas y vistas de un catálogo de datos

Para obtener más información, consulte [Amazon EMR Studio](#) en el tema [Integraciones de los Servicio de AWS con Athena](#).

- Control de acceso anidado: Athena anuncia su compatibilidad con el control de acceso de Lake Formation para datos anidados. En Lake Formation, puede definir y aplicar filtros de datos en columnas anidadas que tengan tipos de datos `struct`. Puede utilizar el filtrado de datos para restringir el acceso de los usuarios a las subestructuras de las columnas anidadas. Para obtener más información acerca de cómo crear un filtro de datos, consulte [Creación de un filtro de datos](#) en la Guía para desarrolladores de AWS Lake Formation.
- Métricas de uso de la capacidad aprovisionada: Athena anuncia nuevas métricas de CloudWatch para las reservas de capacidad. Puede usar las nuevas métricas para realizar un seguimiento del número de DPU que ha aprovisionado y del número de DPU que utilizan sus consultas. Cuando finalicen las consultas, también podrá ver la cantidad de DPU consumidas por la consulta.

Para obtener más información, consulte [Supervisión de las consultas de Athena con métricas de CloudWatch](#).

## Mejoras

- Cambio de mensaje de error: el mensaje de error `Insufficient Lake Formation permissions` ahora dice `Table not found` o `Schema not found`. Este cambio se realizó para evitar que actores malintencionados dedujeran la existencia de recursos de tablas o bases de datos a partir del mensaje de error.

## 16 de noviembre de 2023

Publicado el 16/11/2023

Athena ha lanzado un controlador JDBC nuevo que mejora la experiencia de conexión, consulta y visualización de datos desde aplicaciones de inteligencia empresarial y desarrollo de SQL compatibles. El nuevo controlador es fácil de actualizar. Este controlador puede leer los resultados de las consultas directamente desde Amazon S3, lo que permite que estos resultados se encuentren disponibles con mayor rapidez.

Para obtener más información, consulte [Controlador JDBC 3.x de Athena](#).



## 31 de octubre de 2023

Publicado el 31/10/2023

Amazon Athena anuncia reservas de 1 hora para la capacidad aprovisionada. A partir de hoy, puede reservar y liberar la capacidad aprovisionada después de una hora. Este cambio simplifica la optimización de los costes de las cargas de trabajo cuya demanda cambia con el tiempo.

La capacidad aprovisionada es una característica de Athena que brinda funciones de administración de la carga de trabajo que lo ayudan a priorizar, controlar y escalar sus cargas de trabajo interactivas más importantes. Puede agregar capacidad en cualquier momento para aumentar la cantidad de consultas que ejecuta en simultáneo, controlar qué cargas de trabajo utilizan la capacidad y compartir la capacidad entre las cargas de trabajo.

Para obtener más información, consulte [Administración de la capacidad de procesamiento de consultas](#). Para obtener información sobre los precios, consulte la página de [Precios de Amazon Athena](#).

## 25 de octubre de 2023

Publicado el 26/10/2023

Athena anuncia las siguientes correcciones y mejoras.

Paquete jackson-core: el texto JSON con un valor numérico superior a 1000 caracteres ahora fallará. Esta corrección soluciona el problema de seguridad [sonatype-2022-6438](#).

## 17 de octubre de 2023

Publicado el 17/10/2023

Athena anuncia el controlador ODBC versión 2.0.2.0. Para obtener más información, consulte las notas de la versión [2.0.2.0](#). Para descargar el nuevo controlador ODBC v2, consulte [Descarga del controlador ODBC 2.x](#). Para obtener información sobre la conexión, consulte la [ODBC 2.x de Amazon Athena](#).

## 26 de septiembre de 2023

Publicado el 26/09/2023

Athena anuncia las siguientes características y mejoras.

- Soporte de lectura de Lake Formation para tablas de Delta Lake. Para obtener más información acerca del uso de tablas de Delta Lake con Athena, consulte [Consulta de tablas de Linux Foundation Delta Lake](#).

## 23 de agosto de 2023

Publicado el 23/08/2023

Amazon Athena anuncia la disponibilidad de Athena SQL en la región de Israel (Tel Aviv).

Para obtener una lista completa de los Servicios de AWS disponibles en cada Región de AWS, consulte [Servicios de AWS por región](#).

## 10 de agosto de 2023

Publicado el 10/08/2023

Athena anuncia las siguientes correcciones y mejoras.

### Controlador ODBC versión 2.0.1.1

Athena anuncia el controlador ODBC versión 2.0.1.1. Para obtener más información, consulte las notas de la versión [2.0.1.1](#). Para descargar el nuevo controlador ODBC v2, consulte [Descarga del controlador ODBC 2.x](#). Para obtener información sobre la conexión, consulte el [ODBC 2.x de Amazon Athena](#).

### Controlador JDBC versión 2.1.1

Athena lanza la versión 2.1.1 del controlador JDBC. El controlador resuelve los siguientes problemas:

- Error que se producía al crear una tabla con una instrucción que contenía una expresión regular.
- Problema que provocaba que el parámetro de conexión `ApplicationName` se aplicara de forma incorrecta.

Para descargar el nuevo controlador JDBC, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con JDBC](#).

## 31 de julio de 2023

Publicado el 31/07/2023

Amazon Athena anuncia la disponibilidad de Athena SQL en Regiones de AWS adicionales.

Esta versión amplía la disponibilidad de Athena SQL en las regiones de Asia-Pacífico (Hyderabad), Asia-Pacífico (Melbourne), Europa (España) y Europa (Zúrich).

Para obtener una lista completa de los Servicios de AWS disponibles en cada Región de AWS, consulte [Servicios de AWS por región](#).

## 27 de julio de 2023

Publicado el 27/07/2023

Athena lanza la versión 2023.30.1 del conector de Google BigQuery. Esta versión del conector reduce el tiempo de ejecución de las consultas y agrega soporte para realizar consultas en puntos de conexión privados de BigQuery.

Para obtener información sobre el conector de Google BigQuery, consulte [Conector Google BigQuery de Amazon Athena](#). Para obtener información sobre la actualización de los conectores de orígenes de datos existentes, consulte [Actualización de un conector de origen de datos](#).

## 24 de julio de 2023

Publicado el 24/07/2023

Athena anuncia las siguientes correcciones y mejoras.

- Consultas con uniones: se ha mejorado el rendimiento de determinadas consultas con uniones.
- Combinaciones con comparaciones de tipos: se ha corregido un posible error en la consulta de las instrucciones JOIN que incluían una comparación entre dos tipos diferentes.
- Subconsultas en columnas anidadas: se ha corregido un problema relacionado con los errores de consulta que se producían cuando las subconsultas se correlacionaban en columnas anidadas.
- Vistas de Iceberg: se ha corregido un problema de compatibilidad con la precisión de las columnas de marcas de tiempo en las vistas de Apache Iceberg. Ahora, las vistas de Iceberg que tienen columnas de marca de tiempo se pueden leer independientemente de si las columnas se crearon en la versión 2 o 3 del motor de Athena.

## 20 de julio de 2023

Publicado el 20/07/2023

Athena lanza la versión 2.1.0 del controlador JDBC. El controlador incluye mejoras nuevas y se ha resuelto un problema.

## Mejoras

Se actualizaron las siguientes bibliotecas de analizadores JSON de [Jackson](#):

- jackson-annotations 2.15.2 (anteriormente 2.14.0)
- jackson-core 2.15.2 (anteriormente 2.14.0)
- jackson-databind 2.15.2 (anteriormente 2.14.0)

## Problemas resueltos

- Se ha corregido un problema relacionado con la transmisión de parámetros de matriz cuando se utilizaba la biblioteca [sql2o](#).

Para obtener más información y descargar los nuevos controladores, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con JDBC](#).

## 13 de julio de 2023

Publicado el 19/09/2023

Athena anuncia las siguientes características y mejoras.

- EXPLAIN ANALYZE: se agregó soporte para el tiempo de espera, análisis, planificación y ejecución al resultado de EXPLAIN ANALYZE.
- EXPLAIN: el resultado de EXPLAIN ahora muestra estadísticas cuando la consulta contiene agregaciones.
- Parquet Hive SerDe: se agregó la propiedad `parquet.ignore.statistics` para permitir ignorar las estadísticas de procesamiento al leer los datos de Parquet. Para obtener más información, consulte [Ignorar las estadísticas de Parquet](#).

Para obtener más información sobre EXPLAIN y EXPLAIN ANALYZE, consulte [Uso de EXPLAIN y EXPLAIN ANALYZE en Athena](#). Para obtener más información sobre Parquet Hive SerDe, consulte [El SerDe de Parquet](#).

## 3 de julio de 2023

Publicado el 25/07/2023

El 3 de julio de 2023, Athena comenzó a redactar las cadenas de consulta de los registros de CloudTrail. Ahora, la cadena de consulta tiene el valor `***OMITTED***`. Este cambio se ha realizado para evitar que se divulguen de forma involuntaria nombres de tablas o valores de filtro que puedan incluir información confidencial. Si anteriormente dependía de los registros de CloudTrail para acceder a las cadenas de consulta completas, le recomendamos que utilice la API `Athena::GetQueryExecution` y transfiera el valor `responseElements.queryExecutionId` del registro de CloudTrail. Para obtener más información, consulte la acción [GetQueryExecution](#) en la Referencia de la API de Amazon Athena.

## 30 de junio de 2023

Publicado el 30/06/2023

Ahora, el editor de consultas de Athena admite sugerencias de código de escritura anticipada para una experiencia de creación de consultas más rápida. Ahora puede escribir consultas de SQL con mayor precisión y eficiencia gracias a las siguientes características:

- A medida que escribe, aparecen sugerencias en tiempo real para palabras clave, variables locales, fragmentos y elementos del catálogo.
- Al escribir el nombre de una base de datos o de una tabla seguido de un punto, el editor muestra de forma oportuna una lista de tablas o columnas entre las que puede elegir.
- Al pasar el ratón por encima de una sugerencia de fragmento, aparece una sinopsis que muestra un breve resumen de la sintaxis y del uso del fragmento.
- Para mejorar la legibilidad del código, también se actualizaron las palabras clave y sus reglas de resaltado a fin de adaptarlas a la sintaxis más reciente de Trino y Hive.

Esta característica está habilitada de forma predeterminada. Puede habilitar o deshabilitar la característica mediante la configuración de las preferencias del editor de código.

Para probar las sugerencias de código de escritura anticipada en el editor de consultas de Athena, visite la consola de Athena en <https://console.aws.amazon.com/athena/>.

## 29 de junio de 2023

Publicado el 29/06/2023

- Athena anuncia el controlador ODBC versión 2.0.1.0. Para obtener más información, consulte las notas de la versión [2.0.1.0](#). Para descargar el nuevo controlador ODBC v2, consulte [Descarga del controlador ODBC 2.x](#). Para obtener información sobre la conexión, consulte la [ODBC 2.x de Amazon Athena](#).
- Athena y sus [características](#) ya se encuentran disponibles en la región de Medio Oriente (Emiratos Árabes Unidos). Para obtener una lista completa de los Servicios de AWS disponibles en cada Región de AWS, consulte [Servicios de AWS por región](#).

## 28 de junio de 2023

Publicado el 28/06/2023

Ahora puede utilizar Amazon Athena para consultar objetos restaurados de las [clases de almacenamiento de Amazon S3](#) S3 Glacier Flexible Retrieval (anteriormente Glacier) y S3 Glacier Deep Archive. Esta capacidad se configura por tabla. La característica solo se admite para las tablas de Apache Hive en la versión 3 del motor de Athena.

Para obtener más información, consulte [Consulta de objetos de Amazon S3 Glacier restaurados](#).

## 12 de junio de 2023

Publicado el 12/06/2023

Athena anuncia las siguientes correcciones y mejoras.

- Marcas de tiempo de Parquet Reader: se agregó soporte a fin de leer las marcas de tiempo en forma de `bigint` (milisegundos) para [Parquet Reader](#). Esta actualización proporciona paridad con el soporte de la versión 2 del motor de Athena.
- EXPLAIN ANALYZE: se agregó el tiempo de lectura de la entrada física a las estadísticas de la consulta y al resultado de EXPLAIN ANALYZE. Para obtener más información sobre EXPLAIN ANALYZE, consulte [Uso de EXPLAIN y EXPLAIN ANALYZE en Athena](#).
- INSERT: se mejoró el rendimiento de las consultas en las tablas en las que se escribe con INSERT. Para obtener más información sobre INSERT, consulte [INSERT INTO](#).
- Tablas de Delta Lake: se corrigió un problema con DROP TABLE en las tablas de Delta Lake que impedía que se eliminaran por completo cuando estaban sujetas a modificaciones simultáneas.

## 8 de junio de 2023

Publicado el 08/06/2023

Amazon Athena para Apache Spark anuncia las siguientes características nuevas.

- Soporte para bibliotecas y configuraciones de Java personalizadas: ahora puede utilizar sus propios paquetes y configuraciones personalizadas de Java para sus sesiones de Apache Spark en Athena. Utilice las propiedades de Spark para especificar archivos `.jar`, paquetes u otra configuración personalizada con la consola de Athena, la AWS CLI y la API de Athena. Para obtener más información, consulte [Adición de archivos JAR y configuración personalizada de Spark](#).
- Soporte para tablas de Apache Hudi, Apache Iceberg y Delta Lake: Athena para Spark ahora es compatible con los formatos de tablas de almacenamiento de lagos de datos de código abierto de Apache Iceberg, Apache Hudi y Linux Foundation Delta Lake. Para obtener más información, consulte [Uso de formatos de tabla que no son de Hive en Amazon Athena para Apache Spark](#) y los temas individuales a fin de utilizar las tablas [Apache Iceberg](#), [Apache Hudi](#) y [Linux Foundation Delta Lake](#) de Athena para Spark.
- Soporte de cifrado para Apache Spark: en Athena para Spark, ahora puede habilitar el cifrado de los datos en tránsito entre los nodos de Spark y en los datos en reposo locales almacenados en el disco por Spark. Para habilitar el cifrado de Spark, puede usar la consola, la AWS CLI o la API de Athena. Para obtener más información, consulte [Habilitación del cifrado de Apache Spark](#).

A fin de obtener más información sobre Amazon Athena para Apache Spark, consulte [Uso de Apache Spark en Amazon Athena](#).

## 2 de junio de 2023

Publicado el 02/06/2023

Ahora puede eliminar las reservas de capacidad en Athena y utilizar plantillas de AWS CloudFormation para especificar las reservas de capacidad de Athena.

- Eliminar reservas de capacidad: ahora puede eliminar las reservas de capacidad canceladas en Athena. Se debe cancelar una reserva antes de que esta pueda eliminarse. Al eliminar una reserva de capacidad, la reserva se elimina de su cuenta de inmediato. Ya no se puede hacer referencia a la reserva eliminada, ni siquiera mediante su ARN. Para eliminar una reserva, puede utilizar la consola o la API de Athena. Para obtener más información, consulte [Eliminación de una](#)

[reserva de capacidad](#) en la Guía del usuario de Amazon Athena y [DeleteCapacityReservation](#) en la Referencia de la API de Amazon Athena.

- Utilizar plantillas de AWS CloudFormation para reservas de capacidad: ahora puede utilizar plantillas de AWS CloudFormation para especificar las reservas de capacidad de Athena mediante el recurso `AWS::Athena::CapacityReservation`. Para obtener más información, consulte [AWS::Athena::CapacityReservation](#) en la Guía del usuario de AWS CloudFormation.

Para obtener más información sobre el uso de las reservas de capacidad a fin de aprovisionar su capacidad en Athena, consulte [Administración de la capacidad de procesamiento de consultas](#).

## 25 de mayo de 2023

Publicado el 25/05/2023

Athena ha publicado actualizaciones de conectores de orígenes de datos que mejoran el rendimiento de las consultas federadas. Las optimizaciones de inserción y el filtrado dinámico nuevos permiten realizar más operaciones en la base de datos de origen que en Athena. Estas optimizaciones reducen el tiempo de ejecución de las consultas y la cantidad de datos que se analizan. Estas mejoras requieren la versión 3 del motor de Athena.

Se actualizaron los siguientes conectores:

- [Azure Data Lake Storage](#)
- [Azure Synapse](#)
- [Cloudera Hive](#)
- [Cloudera Impala](#)
- [Db2](#)
- [DynamoDB](#)
- [Google BigQuery](#)
- [Hortonworks](#)
- [MySQL](#)
- [Oracle](#)
- [PostgreSQL](#)
- [Redshift](#)
- [SAP HANA](#)



- [Snowflake](#)
- [SQL Server](#)
- [Teradata](#)

Para obtener información sobre la actualización de los conectores de orígenes de datos, consulte [Actualización de un conector de origen de datos](#).

## 18 de mayo de 2023

Publicado el 18/05/2023

Ahora puede utilizar AWS PrivateLink para las conexiones entrantes de IPv6 a Amazon Athena.

Amazon Athena ha ampliado su compatibilidad con las conexiones entrantes a través de los puntos de conexión del Protocolo de Internet versión 6 (IPv6) para incluir [AWS PrivateLink](#). A partir de hoy, puede conectarse a Athena de forma segura y privada mediante AWS PrivateLink desde su [Amazon Virtual Private Cloud \(Amazon VPC\)](#), además de los puntos de conexión de IPv6 públicos que estaban [disponibles anteriormente](#).

El crecimiento rápido de Internet agota la disponibilidad de las direcciones del Protocolo de Internet de versión 4 (IPv4). El IPv6 multiplica varias veces el número de direcciones disponibles, por lo que ya no es necesario administrar los espacios de direcciones superpuestos en las VPC. Con esta versión, ahora puede combinar los beneficios del direccionamiento de IPv6 con las ventajas de seguridad y rendimiento de AWS PrivateLink.

Para conectarse mediante programación a un servicio de AWS, puede utilizar la [AWS CLI](#) o [AWS SDK](#) para especificar un punto de conexión. Para obtener más información sobre los puntos de conexión de servicio y los puntos de conexión de servicio de Athena, consulte los [puntos de conexión de servicio de AWS](#) y los [puntos de conexión y cuotas de Amazon Athena](#) en la Referencia general de Amazon Web Services.

## 15 de mayo de 2023

Publicado el 15/05/2023

Athena anuncia el lanzamiento de los conectores de Apache Spark DataSourceV2 (DSV2) para DynamoDB, Registros de CloudWatch, Métricas de CloudWatch y CMDB de AWS. Utilice los conectores DSV2 nuevos para consultar estos orígenes de datos mediante Spark. Los conectores

DSV2 utilizan los mismos parámetros que sus conectores federados de Athena correspondientes. Los conectores DSV2 se ejecutan directamente en los trabajos de Spark y no requieren que implemente una función de Lambda para utilizarlos.

Para obtener más información, consulte [Conectores de orígenes de datos de Athena para Apache Spark](#).

## 10 de mayo de 2023

Publicado el 10/05/2023

Se lanzó el controlador ODBC 1.1.20 para Athena.

Características y mejoras:

- Soporte para la anulación de puntos de conexión de Lake Formation.
- El complemento de autenticación de ADFS tiene un parámetro nuevo para configurar el valor de relación de confianza (LoginToRP).
- Actualizaciones de la biblioteca de AWS.

Correcciones de errores:

- Error de desasignación de la instrucción preparada cuando el método `SQLPrepare()` no se podía enviar.
- Error al vincular los parámetros de una instrucción preparada al convertir un tipo C en un tipo de SQL.
- Error de devolución de los datos cuando las consultas `EXPLAIN` y `EXPLAIN ANALYZE` utilizaban `SQLPrepare()` y `SQLExecute()`.

Para obtener más información y descargar los nuevos controladores, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con ODBC](#).

## 8 de mayo de 2023

Publicado el 08/05/2023

Athena anuncia las siguientes correcciones y mejoras.

- Integración de Hudi actualizada: Athena ha actualizado su integración con Apache Hudi. Ahora puede utilizar Athena para consultar las tablas de Hudi 0.12.2, y también se admite la lista de metadatos de Hudi para las tablas de Hudi. Para obtener más información, consulte [Uso de Athena para consultar conjuntos de datos de Apache Hudi](#) y [Lista de metadatos de Hudi](#).
- Corrección de la conversión de marcas de tiempo: se ha corregido la gestión de las conversiones de marcas de tiempo a un tipo de datos de menor precisión. Anteriormente, la versión 3 del motor de Athena redondeaba de forma incorrecta el valor al tipo objetivo en lugar de truncarlo durante la conversión.

En los siguientes ejemplos se ilustra la gestión incorrecta antes de la corrección.

Ejemplo 1: conversión de una marca de tiempo en microsegundos a milisegundos

Datos de ejemplo

```
A, 2020-06-10 15:55:23.383
B, 2020-06-10 15:55:23.382
C, 2020-06-10 15:55:23.383345
D, 2020-06-10 15:55:23.383945
E, 2020-06-10 15:55:23.383345734
F, 2020-06-10 15:55:23.383945278
```

La siguiente consulta intenta recuperar las marcas de tiempo que coinciden con un valor específico.

```
SELECT *
FROM table
WHERE timestamps.col = timestamp'2020-06-10 15:55:23.383'
```

La consulta arrojó los siguientes resultados.

```
A, 2020-06-10 15:55:23.383
C, 2020-06-10 15:55:23.383
E, 2020-06-10 15:55:23.383
```

Antes de la corrección, Athena no incluía los valores `2020-06-10 15:55:23.383945` o `2020-06-10 15:55:23.383945278` porque se redondeaban a `2020-06-10 15:55:23.384`.

Ejemplo 2: conversión de una marca de tiempo a una fecha

La siguiente consulta arrojó un resultado erróneo.

```
SELECT date(timestamp '2020-12-31 23:59:59.999')
```

Resultado

2021-01-01

Antes de la corrección, Athena redondeaba el valor para arriba, por lo tanto, adelantaba el día. Estos valores ahora se truncan en lugar de redondearse para arriba.

## 28 de abril de 2023

Publicado el 28/04/2023

Ahora puede utilizar las reservas de capacidad en Amazon Athena para ejecutar consultas SQL en una capacidad de procesamiento totalmente administrada.

La capacidad aprovisionada brinda funciones de administración de la carga de trabajo que lo ayudan a priorizar, controlar y escalar sus cargas de trabajo interactivas más importantes. Puede agregar capacidad en cualquier momento para aumentar la cantidad de consultas que ejecuta en simultáneo, controlar qué cargas de trabajo utilizan la capacidad y compartir la capacidad entre las cargas de trabajo.

Para obtener más información, consulte [Administración de la capacidad de procesamiento de consultas](#). Para obtener información sobre los precios, consulte la página de [Precios de Amazon Athena](#).

## 17 de abril de 2023

Publicado el 17/04/2023

Athena lanza la versión 2.0.36 del controlador JDBC. El controlador incluye características nuevas y se ha resuelto un problema.

### Nuevas características

- Ahora puede utilizar identificadores de relación de confianza personalizables con la autenticación de AD FS.

- Ahora puede agregar el nombre de la aplicación que utiliza el conector a la cadena del agente de usuario.

## Problemas resueltos

- Se ha corregido un error que se producía cuando se utilizaba `getSchema()` para recuperar un esquema inexistente.

Para obtener más información y descargar los nuevos controladores, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con JDBC](#).

## 14 de abril de 2023

Publicado el 20/06/2023

Athena anuncia las siguientes correcciones y mejoras.

- Al convertir una cadena en una marca de tiempo, se requiere un espacio entre el día y la hora o la zona horaria. Para obtener más información, consulte [Se requiere espacio entre los valores de fecha y hora al convertir una cadena en una marca de tiempo](#).
- Se ha eliminado un cambio sustancial en la forma en que se gestionaba la precisión de la marca de tiempo. Para mantener la coherencia entre la versión 2 y 3 del motor de Athena, la precisión de la marca de tiempo se ve ahora de forma predeterminada en milisegundos y no microsegundos.
- Ahora, Athena impone de forma coherente el acceso al bucket de resultados de consultas cuando ejecuta consultas. Asegúrese de que todas las entidades principales de IAM que ejecutan la acción [StartQueryExecution](#) tengan el permiso [S3:GetBucketLocation](#) en el bucket de resultados de consultas.

## 4 de abril de 2023

Publicado el 04/04/2023

Ahora puede utilizar Amazon Athena para crear y consultar vistas en los orígenes de datos federados. Utilice una vista federada única para consultar varias tablas o subconjuntos de datos externos. Esto simplifica el SQL necesario y le brinda la flexibilidad de ocultar los orígenes de datos de los usuarios finales que deben utilizar SQL para consultar los datos.

Para obtener más información, consulte [Uso de vistas](#) y [Ejecución de consultas federadas](#).

## 30 de marzo de 2023

Publicado el 30/03/2023

Amazon Athena anuncia la disponibilidad de Amazon Athena para Apache Spark en Regiones de AWS adicionales.

Esta versión amplía la disponibilidad de Amazon Athena para Apache Spark en las regiones de Asia-Pacífico (Bombay), Asia-Pacífico (Singapur), Asia-Pacífico (Sídney) y Europa (Fráncfort).

A fin de obtener más información sobre Amazon Athena para Apache Spark, consulte [Uso de Apache Spark en Amazon Athena](#).

## 28 de marzo de 2023

Publicado el 28/03/2023

Athena anuncia las siguientes correcciones y mejoras.

- En las respuestas a las acciones de la API de Athena `GetQueryExecution` y `BatchGetQueryExecution`, el campo `subStatementType` nuevo muestra el tipo de consulta que se ejecutó (por ejemplo, `SELECT`, `INSERT`, `UNLOAD`, `CREATE_TABLE` o `CREATE_TABLE_AS_SELECT`).
- Se ha corregido un error que provocaba que los archivos de manifiesto no se cifraran de forma correcta para las operaciones de escritura de Apache Hive.
- La versión 3 del motor de Athena ahora gestiona los valores `NaN` y `Infinity` de forma correcta en la función `approx_percentile`. La función `approx_percentile` devuelve el percentil aproximado de un conjunto de datos en el porcentaje indicado.

La versión 2 del motor de Athena trata de forma incorrecta `NaN` como un valor superior a `Infinity`. La versión 3 del motor de Athena ahora gestiona `NaN` y `Infinity` de acuerdo con el tratamiento de estos valores en otras funciones analíticas y estadísticas. En los siguientes puntos se describe el comportamiento nuevo con mayor detalle.

- Si `NaN` se encuentra presente en el conjunto de datos, Athena devuelve `NaN`.
- Si `NaN` no se encuentra presente, pero `Infinity` sí lo está, Athena trata a `Infinity` como un número muy grande.
- Si hay varios valores `Infinity`, Athena los trata como un mismo número muy grande. Si es necesario, Athena produce `Infinity`.

- Si un único conjunto de datos tiene ambos (Infinity y -Double.MAX\_VALUE) y el resultado percentil es -Double.MAX\_VALUE, Athena devuelve -Infinity.
- Si un único conjunto de datos tiene Infinity y Double.MAX\_VALUE, y el resultado percentil es Double.MAX\_VALUE, Athena devuelve Infinity.
- Para excluir Infinity y NaN de un cálculo, utilice la función `is_finite()`, como en el siguiente ejemplo.

```
approx_percentile(x, 0.5) FILTER (WHERE is_finite(x))
```

## 27 de marzo de 2023

Publicado el 27/03/2023

Ahora puede especificar un nivel mínimo de cifrado para los grupos de trabajo de Athena SQL en Amazon Athena. Esta característica garantiza que los resultados de todas las consultas del grupo de trabajo de Athena SQL se encuentren cifrados al nivel de cifrado que especifique o supere. Puede elegir entre varios niveles de seguridad de cifrado para proteger los datos. Para configurar el nivel mínimo de cifrado que desee, puede utilizar la consola, la AWS CLI, la API o el SDK de Athena.

La característica de cifrado mínimo no se encuentra disponible para los grupos de trabajo habilitados para Apache Spark. Para obtener más información, consulte [Configuración del cifrado mínimo para un grupo de trabajo](#).

## 17 de marzo de 2023

Publicado el 17/03/2023

Athena anuncia las siguientes correcciones y mejoras.

- Se ha corregido un problema con el conector DynamoDB de Amazon Athena que provocaba que las consultas fallaran con el mensaje de error `KeyConditionExpressions solo debe contener una condición por clave`.

Este problema se produce porque la versión 3 del motor de Athena reconoce la oportunidad de introducir más tipos de predicados que la versión 2 del motor de Athena. En la versión 3 del motor de Athena, cláusulas como `some_column LIKE 'someprefix%'` se introducen como predicados de filtro que aplican un límite inferior y superior a una columna determinada. La versión 2 del motor de Athena no introducía estos predicados. En la versión 3 del motor de Athena, cuando

`some_column` es una columna de clave de clasificación, el motor inserta el predicado de filtro en el conector DynamoDB. Luego, el predicado de filtro se inserta más en el servicio de DynamoDB. Debido a que DynamoDB no admite más de una condición de filtro en una clave de clasificación, DynamoDB devuelve el error.

Para corregir este problema, actualice su conector DynamoDB de Amazon Athena a la versión 2023.11.1. Para obtener instrucciones sobre cómo actualizar el conector, consulte [Actualización de un conector de origen de datos](#).

## 8 de marzo de 2023

Publicado el 08/03/2023

Athena anuncia las siguientes correcciones y mejoras.

- Se ha corregido un problema con las consultas federadas que provocaba que los valores de los predicados de marca de tiempo se enviaran en microsegundos en lugar de milisegundos.

## 15 de febrero de 2023

Publicado el 15/02/2023

Athena anuncia las siguientes correcciones y mejoras.

- Ahora puede utilizar el [cifrado del cliente](#) a fin de cifrar los datos en Amazon S3 para las operaciones de escritura de Iceberg.
- Se ha corregido un problema que afectaba al [cifrado del servidor](#) en Amazon S3 para las operaciones de escritura de Iceberg.

## 31 de enero de 2023

Publicado el 31/01/2023

Ahora puede utilizar Amazon Athena para consultar datos en Google Cloud Storage. Al igual que Amazon S3, Google Cloud Storage es un servicio administrado que almacena los datos en buckets. Utilice el conector de Athena para Google Cloud Storage para ejecutar consultas federadas interactivas en datos externos.



Para obtener más información, consulte [Conector de Google Cloud Storage para Amazon Athena](#).

## 20 de enero de 2023

Publicado el 20/01/2023

Ahora puede ver documentación ampliada sobre la compatibilidad de compresión de Athena. Se añadieron temas individuales para [Compresión de tablas de Hive](#), [Compresión de tablas de Iceberg](#) y [Niveles de compresión ZSTD](#).

Para obtener más información, consulte [Compatibilidad con la compresión de Athena](#).

## 3 de enero de 2023

Publicado el 03/01/2023

Athena anuncia las siguientes actualizaciones:

- Comandos adicionales para los metaalmacenes de Hive: puede utilizar Athena para conectarse a su almacén autoadministrado de Apache Hive como catálogo de metadatos y consultar los datos almacenados en Amazon S3. Con esta versión, puede usar CREATE TABLE AS (CTAS), INSERT INTO y 12 comandos adicionales de lenguaje de definición de datos (DDL) para interactuar con el metaalmacén de Apache Hive. Puede administrar sus esquemas de metaalmacén de Hive directamente desde Athena mediante este conjunto ampliado de capacidades SQL.

Para obtener más información, consulte [Uso del conector de datos de Athena para metaalmacén externo de Hive](#).

- Controlador JDBC versión 2.0.35: Athena lanza el controlador JDBC versión 2.0.35. El controlador JDBC 2.0.35 contiene las siguientes actualizaciones:
  - El controlador ahora usa las siguientes bibliotecas para el analizador JSON de Jackson.
    - jackson-annotations 2.14.0 (anteriormente 2.13.2)
    - jackson-core 2.14.0 (anteriormente 2.13.2)
    - jackson-databind 2.14.0 (anteriormente 2.13.2.2)
  - Se ha interrumpido el soporte para la versión 4.1 de JDBC.

Para obtener más información y descargar el nuevo controlador, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con JDBC](#).

# Notas de la versión de Athena para 2022

## 14 de diciembre de 2022

Publicado el 14/12/2022

Ahora puede utilizar el conector de Amazon Athena para que Kafka ejecute consultas SQL en datos de transmisión. Por ejemplo, puede ejecutar consultas analíticas sobre datos de transmisión en tiempo real en Amazon Managed Streaming para Apache Kafka (Amazon MSK) y unirlos a los datos históricos de su lago de datos de Amazon S3.

El conector de Amazon Athena para Kafka admite consultas en varios motores de transmisión. Puede utilizar Athena para ejecutar consultas SQL en clústeres aprovisionados y sin servidor de Amazon MSK, en implementaciones de Kafka autoadministrado y en datos de transmisión en Confluent Cloud.

Para obtener más información, consulte [Conector para MSK de Amazon Athena](#).

## 2 de diciembre de 2022

Publicado el 02/12/2022

Athena lanza la versión 2.0.34 del controlador JDBC. El controlador JDBC 2.0.34 incluye las siguientes características y problemas resueltos:

- Reutilización de resultados de consultas: ahora puede reutilizar los resultados de consultas ejecutadas anteriormente hasta un límite de tiempo especificado, en lugar de que Athena vuelva a calcular los resultados cada vez que se ejecuta la consulta. Para obtener más información, consulte la guía de instalación y configuración, disponible en la página de descargas de JDBC y [Reutilización de los resultados de las consultas](#).
- Compatibilidad con Ec2InstanceMetadata: el controlador JDBC ahora es compatible con el [método de autenticación Ec2InstanceMetadata](#) mediante [perfiles de instancia](#) de IAM.
- Corrección de excepciones basadas en caracteres: se ha corregido una excepción que se producía con consultas que contenían determinados caracteres de idioma.
- Corrección de vulnerabilidad: se corrigió una vulnerabilidad relacionada con las dependencias de AWS empaquetadas con el conector.

Para obtener más información y descargar los nuevos controladores, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con JDBC](#).

## 30 de noviembre de 2022

Publicado el 30/11/2022

Ahora puede crear y ejecutar aplicaciones de Apache Spark y cuadernos compatibles con Jupyter de forma interactiva en Athena. Ejecute análisis de datos en Athena con Spark sin tener que planificar, configurar ni administrar los recursos. Envíe el código Spark para su procesamiento y reciba los resultados directamente. Utilice la experiencia simplificada de cuadernos de la consola de Amazon Athena para desarrollar aplicaciones de Apache Spark mediante Python o [API de cuadernos de Athena](#).

Apache Spark en Amazon Athena no requiere servidor y proporciona un escalado automático y bajo demanda que ofrece computación instantánea para cumplir con los cambios en los volúmenes de datos y los requisitos de procesamiento.

Para obtener más información, consulte [Uso de Apache Spark en Amazon Athena](#).

## 18 de noviembre de 2022

Publicado el 18/11/2022

Ahora puede utilizar el conector de Amazon Athena para IBM Db2 para realizar consultas de Db2 desde Athena. Por ejemplo, puede ejecutar consultas analíticas a través de un almacenamiento de datos en Db2 y un lago de datos en Amazon S3.

El conector para Db2 de Amazon Athena expone varias opciones de configuración a través de variables de entorno de Lambda. Para obtener información sobre las opciones de configuración, los parámetros, las cadenas de conexión, la implementación y las limitaciones, consulte [Conector para IBM Db2 de Amazon Athena](#).

## 17 de noviembre de 2022

Publicado el 17/11/2022

La compatibilidad de Apache Iceberg en la versión 3 del motor de Athena ofrece ahora las siguientes características mejoradas de transacciones ACID:

- Compatibilidad con ORC y Avro: cree tablas de Iceberg con los formatos de archivo basados en filas y columnas de [Apache Avro](#) y [Apache ORC](#). La compatibilidad con estos formatos se suma a la compatibilidad existente con Parquet.
- MERGE INTO: utilice el comando MERGE INTO para combinar datos a escala de manera eficiente. MERGE INTO combina las operaciones INSERT, UPDATE y DELETE en una sola transacción. Esto reduce la sobrecarga de procesamiento en su canalización de datos y requiere menos SQL para escribir. Para obtener más información, consulte [Actualización de los datos de las tablas de Iceberg](#) y [MERGE INTO](#).
- Compatibilidad con CTAS y VIEW: utilice CREATE TABLE AS SELECT (CTAS) y las instrucciones CREATE VIEW con tablas de Iceberg. Para obtener más información, consulte [CREATE TABLE AS](#) y [CREATE VIEW](#).
- Compatibilidad con VACUUM: puede utilizar la instrucción VACUUM para optimizar su lago de datos al eliminar las instantáneas y los datos que ya no sean necesarios. Puede utilizar esta característica para mejorar el rendimiento de la lectura y cumplir con los requisitos reglamentarios, como el [RGPD](#). Para obtener más información, consulte [Optimización de las tablas de Iceberg](#) y [VACUUM](#).

Estas nuevas características requieren la versión 3 del motor de Athena y están disponibles en todas las regiones en las que se admite Athena. Puede utilizarlos con la [consola de Athena](#), los [controladores](#) o la [API](#).

Para obtener información acerca del uso de Iceberg en Athena, consulte [Uso de tablas de Apache Iceberg](#).

## 14 de noviembre de 2022

Publicado el 14/11/2022

Amazon Athena ahora admite puntos de conexión IPv6 para las conexiones entrantes que puede utilizar para invocar funciones de Athena a través de IPv6. Puede utilizar esta característica para cumplir con los requisitos de cumplimiento de IPv6. También elimina la necesidad de equipos de red adicionales para gestionar la traducción de direcciones entre IPv4 e IPv6.

Para utilizar esta característica, configure sus aplicaciones para que utilicen los nuevos puntos de conexión de doble pila de Athena, que admiten IPv4 e IPv6. Los puntos de conexión de doble pila utilizan el formato `athena.region.api.aws`. Por ejemplo, el punto de conexión de doble pila en la región Este de EE. UU. (Norte de Virginia) es `athena.us-east-1.api.aws`.

Cuando realiza una solicitud a un punto de conexión de doble pila de Athena, el punto de conexión resuelve a una dirección IPv6 o IPv4, según el protocolo que utilicen la red y el cliente. Para conectarse mediante programación a un servicio de AWS, puede utilizar la [AWS CLI](#) o [AWS SDK](#) para especificar un punto de conexión.

Para obtener más información sobre los puntos de conexión de servicios, consulte [Puntos de conexión de servicios de AWS](#). Para obtener más información sobre los puntos de conexión del servicio de Athena, consulte [Puntos de conexión y cuotas de Amazon Athena](#) en la documentación de AWS.

Puede utilizar los nuevos puntos de conexión de doble pila de Athena para las conexiones entrantes sin costo adicional. Los puntos de conexión de doble pila suelen estar disponibles en todas las Regiones de AWS.

## 11 de noviembre de 2022

Publicado el 11/11/2022

Athena anuncia las siguientes correcciones y mejoras.

- Control de acceso detallado ampliado de Lake Formation: ahora puede utilizar políticas de control de acceso detallado de [AWS Lake Formation](#) en las consultas de Athena para los datos almacenados en cualquier formato de archivo o tabla compatible. Puede utilizar un control de acceso detallado en Lake Formation para restringir el acceso a los datos de los resultados de las consultas mediante filtros de datos para lograr una seguridad de nivel de columna, fila y celda. Los formatos de tabla admitidos en Athena incluyen Apache Iceberg, Apache Hudi y Apache Hive. El control de acceso detallado ampliado está disponible en todas las regiones compatibles con Athena. La compatibilidad ampliada con formatos de tablas y archivos requiere la [Versión 3 del motor Athena](#), que [ofrece nuevas características y un mejor rendimiento de las consultas](#), pero no cambia la forma en que se configuran políticas de control de acceso detallado en Lake Formation.

El uso de este control de acceso detallado ampliado en Athena tiene las siguientes consideraciones:

- EXPLAIN: la información de filtrado de filas o celdas definida en Lake Formation y la información de estadísticas de consultas no se muestra en la salida de EXPLAIN ni EXPLAIN ANALYZE. Para obtener información acerca de EXPLAIN para Athena, consulte [Uso de EXPLAIN y EXPLAIN ANALYZE en Athena](#).
- Metaalmacenes de Hive externos: las columnas ocultas de Apache Hive no se pueden utilizar para filtrar el control de acceso con precisión, y el control de acceso detallado no

admite las tablas ocultas del sistema de Apache Hive. Para obtener más información, consulte [Consideraciones y limitaciones](#) en el tema [Uso del conector de datos de Athena para metaalmacén externo de Hive](#).

- Estadísticas de consultas: la información sobre el recuento de filas y el tamaño de los datos de entrada y salida de nivel de etapa no se muestra en las estadísticas de consulta de Athena cuando una consulta tiene filtros de nivel de fila definidos en Lake Formation. Para obtener información sobre cómo ver las estadísticas de las consultas de Athena, consulte [Visualización de estadísticas y detalles de ejecución de consultas completadas](#) y [GetQueryRuntimeStatistics](#).
- Grupos de trabajo: los usuarios del mismo grupo de trabajo de Athena pueden ver los datos que el control de acceso detallado de Lake Formation ha configurado para que el grupo de trabajo pueda acceder a ellos. Para obtener información sobre el uso de Athena para consultar datos registrados en Lake Formation, consulte [Uso de Athena para consultar datos registrados en AWS Lake Formation](#).

Para obtener información sobre el uso del control de acceso detallado en Lake Formation, consulte [Manage fine-grained access control using AWS Lake Formation](#) en el Blog de macrodatos de AWS.

- Consulta federada de Athena: ahora la consulta federada de Athena conserva el formato original de los nombres de campo de los objetos `struct`. Anteriormente, los nombres de los campos `struct` se convertían automáticamente en minúsculas.

## 8 de noviembre de 2022

Publicado el 08/11/2022

Ahora puede utilizar la característica de almacenamiento en caché para reutilizar los resultados de las consultas a fin de acelerar las consultas repetidas en Athena. Una consulta repetida es una consulta SQL idéntica a una enviada recientemente que produce los mismos resultados. Cuando sea necesario ejecutar varias consultas idénticas, el almacenamiento en caché para reutilizar los resultados puede disminuir el tiempo necesario para producir resultados. El almacenamiento en caché para reutilizar los resultados también reduce los costos al reducir la cantidad de bytes escaneados.

Para obtener más información, consulte [Reutilización de los resultados de las consultas](#).

## 13 de octubre de 2022

Publicado el 13/10/2022

Athena anuncia la versión 3 del motor Athena.

Athena ha actualizado su motor de consultas SQL para incluir las características más recientes del proyecto de código abierto [Trino](#). Además de admitir todas las características de la versión 2 del motor Athena, la versión 3 del motor incluye más de 50 funciones SQL nuevas, 30 características nuevas y más de 90 mejoras en el rendimiento de las consultas. Con el lanzamiento de hoy, Athena también presenta un enfoque de integración continua para la administración del software de código abierto que mejora la aceptación de los proyectos de Trino y [Presto](#) para que pueda acceder más rápido a las mejoras de la comunidad, integradas y ajustadas en el motor Athena.

Para obtener más información, consulte [Versión 3 del motor Athena](#).

## 10 de octubre de 2022

Publicado el 10/10/2022

Athena lanza la versión 2.0.33 del controlador JDBC. El controlador JDBC 2.0.33 incluye los siguientes cambios:

- Se agregaron las propiedades de la nueva versión del controlador, la versión de JDBC y el nombre del complemento a la cadena de agente de usuario de la clase de proveedor de credenciales.
- Se corrigieron los mensajes de error y se agregó la información necesaria.
- Ahora se cancela la asignación de las instrucciones preparadas si la conexión se cierra o si se produce un error en la ejecución de la instrucción que prepara Athena.

Para obtener más información y descargar los nuevos controladores, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con JDBC](#).

## 23 de septiembre de 2022

Publicado el 26/09/2022

El conector de Amazon Athena para Neptune ahora admite la coincidencia sin distinción entre mayúsculas y minúsculas en los nombres de columnas y tablas.

- El conector de orígenes de datos de Neptune puede resolver los nombres de las columnas en las tablas de Neptune que hacen distinción entre mayúsculas y minúscula, incluso si todos los nombres de las columnas aparecen en minúsculas en la tabla de AWS Glue. Para habilitar este

comportamiento, establezca la variable de entorno `enable_caseinsensitivematch` como `true` en la función de Lambda del conector para Neptune.

- Dado que AWS Glue solo admite nombres de tablas en minúsculas, al crear una tabla de AWS Glue para Neptune, especifique el parámetro de tabla de AWS Glue `"gl_label" = table_name`.

Para obtener más información sobre el conector para Neptune, consulte [Conector para Neptune de Amazon Athena](#).

## 13 de septiembre de 2022

Publicado el 13/09/2022

Athena anuncia las siguientes correcciones y mejoras.

- Almacén de metadatos de Hive externo: Athena ahora devuelve el valor NULL en lugar de una excepción cuando una cláusula WHERE incluye una partición que no existe en un [almacén de metadatos de Hive](#) (EHMS). El nuevo comportamiento coincide con el de AWS Glue Data Catalog.
- Consultas parametrizadas: ahora se pueden enviar los valores de [consultas parametrizadas](#) al tipo de datos DOUBLE.
- Apache Iceberg: las operaciones de escritura en [tablas de Iceberg](#) ahora se llevan a cabo correctamente cuando el [bloqueo de objetos](#) está activado en un bucket de Amazon S3.

## 31 de agosto de 2022

Publicado el 31/08/2022

Amazon Athena anuncia la disponibilidad de Athena y sus [características](#) en la región Asia-Pacífico (Yakarta).

Esta versión amplía la disponibilidad de Athena en la región Asia-Pacífico e incluye Asia-Pacífico (Hong Kong), Asia-Pacífico (Yakarta), Asia-Pacífico (Bombay), Asia-Pacífico (Osaka), Asia-Pacífico (Seúl), Asia-Pacífico (Singapur), Asia-Pacífico (Sídney) y Asia-Pacífico (Tokio). Para obtener una lista completa de Servicios de AWS disponibles en estas y otras regiones, consulte la [Lista de servicios por Región de AWS](#).

## 23 de agosto de 2022

Publicado el 23/08/2022



La versión [v2022.32.1](#) del SDK de Athena Query Federation incluye los siguientes cambios:

- Se ha agregado compatibilidad con el conector de orígenes de datos Oracle de Amazon Athena para conexiones basadas en SSL a instancias de Amazon RDS. La compatibilidad se limita al protocolo de seguridad de la capa de transporte (TLS) y a la autenticación del servidor por parte del cliente. Dado que la autenticación mutua no se admite en Amazon RDS, la actualización no incluye la compatibilidad con la autenticación mutua.

Para obtener más información, consulte [Conector Oracle de Amazon Athena](#).

## 3 de agosto de 2022

Publicado el 03/08/2022

Athena lanza el controlador JDBC versión 2.0.32. El controlador JDBC 2.0.32 incluye los siguientes cambios:

- La cadena `User-Agent` enviada al SDK de Athena se ha ampliado para incluir la versión del controlador, la versión de la especificación JDBC y el nombre del complemento de autenticación.
- Se solucionó una `NullPointerException` arrojada cuando no se proporcionó ningún valor para el parámetro `CheckNonProxyHost`.
- Se solucionó un problema con el análisis `login_url` en el complemento de autenticación `BrowserSaml`.
- Se solucionó un problema de host proxy que se produjo cuando el parámetro `UseProxyforIdp` se estableció en `true`.

Para obtener más información y descargar los nuevos controladores, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con JDBC](#).

## 1 de agosto de 2022

Publicado el 01/08/2022

Athena anuncia mejoras en el SDK de Athena Query Federation y en los conectores de origen de datos prediseñados de Athena. Las mejoras incluyen lo siguiente:

- **Análisis de estructuras:** se ha corregido el problema de análisis `GlueFieldLexer` en el SDK de Athena Query Federation que impedía que ciertas estructuras complicadas mostraran todos sus datos. Este problema afectó a los conectores creados en el SDK de Athena Query Federation.
- **Tablas de AWS Glue:** se ha agregado compatibilidad con los tipos de columna `set` y `decimal` en las tablas de AWS Glue.
- **Conector de DynamoDB:** se agregó la posibilidad de ignorar las mayúsculas en los nombres de atributos de DynamoDB. Para obtener más información, consulte `disable_projection_and_casing` en la sección [Parámetros](#) de la página [Conector para DynamoDB de Amazon Athena](#).

Para obtener más información, consulte [Versión v2022.30.2 de Athena Query Federation](#) en GitHub.

## 21 de julio de 2022

Publicado el 21/07/2022

Ahora puede analizar y depurar sus consultas mediante métricas de rendimiento y herramientas de análisis de consultas visuales e interactivas en la consola de Athena. Los datos de rendimiento de la consulta y los detalles de ejecución pueden ayudarlo a identificar los cuellos de botella en las consultas, inspeccionar los operadores y las estadísticas de cada etapa de una consulta, rastrear el volumen de datos que fluyen entre las etapas y validar el impacto de los predicados de consulta. Ahora puede hacer lo siguiente:

- Acceda al plan de ejecución distribuido y lógico de su consulta con un solo clic.
- Explore las operaciones en cada etapa antes de que se ejecute la etapa.
- Visualice el rendimiento de las consultas completadas con métricas del tiempo empleado en las etapas de colocación en cola, planificación y ejecución.
- Obtenga información sobre el número de filas y la cantidad de datos de origen procesados y generados por la consulta.
- Consulte detalles pormenorizados de la ejecución de sus consultas presentadas en contexto y en formato de gráfico interactivo.
- Utilice detalles de ejecución precisos a nivel de etapa para comprender el flujo de datos a través de la consulta.
- Analice los datos de rendimiento de las consultas de manera programada mediante API nuevas para [obtener estadísticas de tiempo de ejecución](#), una característica que también se lanzó hoy.

Para obtener información sobre cómo utilizar estas funciones en sus consultas, vea el videotutorial [Optimize Amazon Athena Queries with New Query Analysis Tools](#) (Optimice las consultas de Amazon Athena con nuevas herramientas de análisis de consultas) en el canal de YouTube de AWS.

Para obtener la documentación, consulte [Ver planes de ejecución para consultas SQL y Visualización de estadísticas y detalles de ejecución de consultas completadas](#).

## 11 de julio de 2022

Publicado el 11/07/2022

Ahora puede ejecutar consultas parametrizadas directamente desde la consola o API de Athena sin preparar instrucciones SQL con anterioridad.

Ahora, cuando ejecute consultas en la consola de Athena que tienen parámetros en forma de signos de interrogación, la interfaz de usuario le pedirá que ingrese valores para los parámetros directamente. Esto elimina la necesidad de modificar los valores literales en el editor de consultas cada vez que quiera ejecutar la consulta.

Si usa la API mejorada [query execution](#), ahora puede proporcionar los parámetros de ejecución y sus valores en una sola llamada.

Para obtener más información, consulte [Uso de consultas parametrizadas](#) en esta guía del usuario y la publicación del Blog de macrodatos de AWS [Utilizar consultas parametrizadas de Amazon Athena para proporcionar datos como servicio](#).

## 8 de julio de 2022

Publicado el 08/07/2022

Athena anuncia las siguientes correcciones y mejoras.

- Se ha corregido un problema con la gestión de conversión de columnas DATE para puntos de conexión de SageMaker (UDF) que provocaba errores en las consultas.

## 6 de junio de 2022

Publicado el 06/06/2022

Athena lanza el controlador JDBC versión 2.0.31. El controlador JDBC 2.0.31 incluye los siguientes cambios:

- Problema de dependencia log4j: se ha resuelto el mensaje de error No se puede encontrar clase de controlador causado por una dependencia log4j.

Para obtener más información y descargar los nuevos controladores, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con JDBC](#).

## 25 de mayo de 2022

Publicado el 25/05/2022

Athena anuncia las siguientes correcciones y mejoras.

- Soporte Iceberg
  - Se introdujo compatibilidad para consultas entre regiones. Ahora puede consultar las tablas de Iceberg en una Región de AWS diferente de la Región de AWS que está usando. No se admiten las consultas entre regiones en las regiones de China.
  - Se introdujo compatibilidad con la configuración de cifrado del lado del servidor. Ahora puede utilizar [SSE-S3/SSE-KMS](#) para cifrar los datos de las operaciones de escritura de Iceberg en Amazon S3.

Para obtener información acerca del uso de Apache Iceberg en Athena, consulte [Uso de tablas de Apache Iceberg](#).

- Versión del controlador JDBC 2.0.30

El controlador JDBC 2.0.30 para Athena presenta las siguientes mejoras:

- Soluciona un problema de carrera de datos que afectaba a las instrucciones preparadas parametrizadas.
- Soluciona un problema de inicio de aplicaciones que se produjo en entornos de compilación de Gradle.

Para descargar el controlador JDBC 2.0.30, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con JDBC](#).

## 6 de mayo de 2022

Publicado el 06/05/2022

Se publicaron los controladores JDBC 2.0.29 y ODBC 1.1.17 para Athena.

Estos controladores incluyen los siguientes cambios:

- Se ha actualizado el proceso de lanzamiento del navegador del complemento SAML.

Para obtener más información sobre estos cambios y, a fin de descargar los nuevos controladores, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con JDBC](#) y [Conexión a Amazon Athena con ODBC](#).

## 22 de abril de 2022

Publicado el 22/04/2022

Athena anuncia las siguientes correcciones y mejoras.

- Se ha solucionado un problema en la sección de [índices de partición y característica de filtrado](#) con la caché de particiones que se produjo cuando se cumplieron las siguientes condiciones:
  - La clave `partition_filtering.enabled` se estableció como `true` en las propiedades AWS Glue de tabla de una tabla.
  - La misma tabla se utilizó varias veces con distintos valores de filtro de partición.

## 21 de abril de 2022

Publicado el 21/04/2022

Ahora puede utilizar Amazon Athena para ejecutar consultas federadas en orígenes de datos nuevos, incluidos Google BigQuery, Azure Synapse y Snowflake. Los nuevos conectores de orígenes de datos incluyen lo siguiente:

- [Gen2 de Azure Data Lake Storage \(ADLS\)](#)
- [Azure Synapse](#)
- [Cloudera Hive](#)
- [Cloudera Impala](#)
- [Google BigQuery](#)
- [Hortonworks](#)
- [Microsoft SQL Server](#)
- [Oracle](#)

- [SAP HANA \(Edición Express\)](#)
- [Snowflake](#)
- [Teradata](#)

Para obtener una lista completa de los orígenes de datos admitidos por Athena, consulte [Conectores de orígenes de datos disponibles](#).

Para facilitar la navegación por los orígenes disponibles y conectarse a los datos, ahora puede buscar, clasificar y filtrar los conectores disponibles desde una pantalla actualizada de Orígenes de datos en la consola de Athena.

Para obtener más información sobre las consultas de orígenes federados, consulte [Uso de consulta federada de Amazon Athena](#) y [Ejecución de consultas federadas](#).

## 13 de abril de 2022

Publicado el 13/04/2022

Athena lanza el controlador JDBC versión 2.0.28. El controlador JDBC 2.0.28 incluye los siguientes cambios:

- Soporte JWT: ahora es compatible con tokens web de JSON (JWT) para la autenticación. Para obtener información sobre la utilización de JWT con el controlador JDBC, consulte la guía de instalación y configuración, que se puede descargar desde la [página del controlador JDBC](#).
- Bibliotecas Log4j actualizadas: el controlador JDBC utiliza ahora las siguientes bibliotecas Log4j:
  - Log4j-api 2.17.1 (antes 2.17.0)
  - Log4j-core 2.17.1 (antes 2.17.0)
  - Log4J-jcl 2.17.2
- Otras mejoras: el nuevo controlador también incluye las siguientes mejoras y correcciones de errores:
  - La característica de instrucciones preparadas por Athena ya está disponible a través de JDBC. Para obtener más información sobre las instrucciones preparadas, consulte [Uso de consultas parametrizadas](#).
  - La federación SAML de Athena JDBC ahora funciona para las regiones de China.
  - Mejoras adicionales menores.

Para obtener más información y descargar los nuevos controladores, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con JDBC](#).

## 30 de marzo de 2022

Publicado el 30/03/2022

Athena anuncia las siguientes correcciones y mejoras.

- Consultas entre regiones: ahora puede utilizar Athena para consultar datos ubicados en un bucket de Amazon S3 en Regiones de AWS incluidos Asia-Pacífico (Hong Kong), Medio Oriente (Baréin), África (Ciudad del Cabo) y Europa (Milán). No se admiten las consultas entre regiones en las regiones de China.
- Para obtener una lista de Regiones de AWS en la que Athena está disponible, consulte [Cuotas y puntos de conexión de Amazon Athena](#).
- Para obtener más información acerca de cómo habilitar una Región de AWS que está deshabilitada de forma predeterminada, consulte [Activación de una región](#).
- Para obtener información sobre las consultas en todas las regiones, consulte [Consultas entre regiones](#).

## 18 de marzo de 2022

Publicado el 18/03/2022

Athena anuncia las siguientes correcciones y mejoras.

- Filtrado dinámico: se mejoró el [filtrado dinámico](#) para columnas enteras aplicando eficazmente el filtro a cada registro de una tabla correspondiente.
- Iceberg: se ha corregido un error que provocaba errores al escribir archivos Iceberg Parquet de más de 2 GB.
- Salida sin comprimir: las declaraciones [CREATE TABLE](#) ahora admiten la escritura de archivos sin comprimir. Para escribir archivos sin comprimir, utilice la siguiente sintaxis:
  - CREATE TABLE (archivo de texto o JSON): en TBLPROPERTIES, especifique `write.compression = NONE`.
  - CREATE TABLE (Parquet): en TBLPROPERTIES, especifique `parquet.compression = UNCOMPRESSED`.

- CREATE TABLE (ORC): en TBLPROPERTIES, especifique `orc.compress = NONE`.
- Compresión: se corrigió un problema con las inserciones de tablas de archivos de texto que creaban archivos comprimidos en un formato, pero utilizaban otra extensión de archivo de formato de compresión cuando se utilizaban métodos de compresión no predeterminados.
- Avro: se corrigieron los problemas que se produjeron al leer decimales del tipo fijo de archivos Avro.

## 2 de marzo de 2022

Publicado el 02/03/2022

Athena anuncia las siguientes características y mejoras.

- Ahora puede conceder al propietario del bucket de Simple Storage Service (Amazon S3) acceso de control total sobre los resultados de las consultas cuando [estén habilitadas las ACL](#) para el bucket de resultados de las consultas. Para obtener más información, consulte [Especificación de una ubicación de resultados de consulta](#).
- Ahora puede actualizar las consultas con nombre existentes. Para obtener más información, consulte [Uso de consultas guardadas](#).

## 23 de febrero de 2022

Publicado el 23/02/2022

Athena anuncia las siguientes correcciones y mejoras de rendimiento.

- Mejoras en la gestión de la memoria para mejorar el rendimiento y reducir los errores de memoria.
- Athena ahora lee las columnas de marca temporal ORC con información de zona horaria almacenada en pies de página del conjunto seccionado (stripe) y escribe archivos ORC con zona horaria (UTC) en pies de página. Esto solo afecta al comportamiento de las lecturas de marca temporal ORC si el archivo ORC que se va a leer se ha creado en un entorno de zona horaria distinto a UTC.
- Se corrigieron estimaciones incorrectas del tamaño de la tabla de enlaces simbólicos que generaban planes de consulta inferiores a los óptimos.
- Ahora se pueden consultar vistas explosionadas laterales en la consola de Athena desde orígenes de datos de metaalmacén de Hive.



- Mensajes de error de lectura de Simple Storage Service (Amazon S3) mejorados para incluir información más detallada de [Código de error de Simple Storage Service \(Amazon S3\)](#).
- Se corrigió un error que provocaba que los archivos de salida en formato ORC se volvieran incompatibles con Apache Hive 3.1.
- Se corrigió un error que provocaba que los nombres de tablas con comillas fallaran en determinadas consultas DML y DDL.

## 15 de febrero de 2022

Publicado el 15/02/2022

Amazon Athena ha aumentado la cuota de consulta DML activa en todas las regiones de AWS. Las consultas activas incluyen tanto las consultas en ejecución como en cola. Con este cambio, ahora puede haber más consultas DML en estado activo que antes.

Para obtener más información acerca de Service Quotas de Athena, consulte [Service Quotas](#). Para obtener información sobre las cuotas de consulta de la región en la que utiliza Athena, consulte los [puntos de conexión y cuotas de Amazon Athena](#) en la Referencia general de AWS.

Para supervisar el uso de cuotas, puede utilizar las métricas de uso de CloudWatch. Athena publica la métrica `ActiveQueryCount` en el espacio de nombres `AWS/Usage`. Para obtener más información, consulte [Supervisión de las métricas de uso de Athena](#).

Después de revisar su uso, puede utilizar la consola de [Service Quotas](#) para solicitar un aumento de cuota. Si anteriormente solicitó un aumento de cuota para la cuenta, la cuota solicitada sigue siendo aplicable si supera la nueva cuota de consulta DML activa predeterminada. En caso contrario, todas las cuentas utilizan el nuevo valor predeterminado.

## 14 de febrero de 2022

Publicado el 14/02/2022

En esta versión se agrega el subcampo `ErrorType` al objeto de respuesta [AthenaError](#) de la acción de la API [GetQueryExecution](#) de Athena.

Si bien el campo `ErrorCode` existente indica el origen general de una consulta fallida (sistema, usuario u otro), el nuevo campo `ErrorType` proporciona información más detallada sobre el error que se ha producido. Combine la información de ambos campos para obtener información sobre las causas del error de la consulta.

Para obtener más información, consulte [Catálogo de errores de Athena](#).

## 9 de febrero de 2022

Publicado el 09/02/2022

La consola antigua de Athena ya no está disponible. La nueva consola de Athena admite todas las características de la consola anterior, pero con una interfaz moderna y fácil de usar e incluye nuevas características que mejoran la experiencia de desarrollar consultas, analizar datos y administrar el uso. Para utilizar la nueva consola Athena, vaya a <https://console.aws.amazon.com/athena/>.

## 8 de febrero de 2022

Publicado el 08/02/2022

Propietario esperado del bucket: como medida de seguridad adicional, ahora puede especificar opcionalmente el ID de la Cuenta de AWS que espera que sea la propietaria del bucket de la ubicación de salida de los resultados de la consulta en Athena. Si el ID de la cuenta del propietario del bucket de los resultados de la consulta no coincide con el ID de la cuenta especificado, los intentos de generar el bucket fallarán con un error de permisos de Amazon S3. Puede establecer esta configuración a nivel de cliente o de grupo de trabajo.

Para obtener más información, consulte [Especificación de una ubicación de resultados de consulta](#).

## 28 de enero de 2022

Publicado el 28/01/2022

Athena anuncia las siguientes mejoras en las características del motor.

- Apache Hudi: las consultas de instantáneas en las tablas Merge on Read (fusionar al leer, MoR) de Hudi ahora pueden leer columnas de marca de tiempo que tienen el tipo de datos INT64.
- Consultas UNION: mejora del rendimiento y reducción del análisis de datos de determinadas consultas UNION que analizan la misma tabla varias veces.
- Consultas disyuntivas: mejora del rendimiento de las consultas que solo tienen valores disyuntivos en cada columna de partición en el filtro.
- Mejoras en la proyección de particiones
  - Ahora se permiten varios valores disyuntivos en la condición de filtro en las columnas del tipo `injected`. Para obtener más información, consulte [Tipo inyectado](#).

- Mejora del rendimiento de columnas de tipos basados en cadenas como CHAR o VARCHAR que solo tienen valores disyuntivos en el filtro.

## 13 de enero de 2022

Publicado el 13/01/2022

Se publicaron los controladores JDBC 2.0.27 y ODBC 1.1.15 para Athena.

El controlador JDBC 2.0.27 incluye los siguientes cambios:

- El controlador se actualizó para recuperar catálogos externos.
- El número ampliado de versión del controlador ahora se incluye en la cadena user-agent como parte de la llamada a la API de Athena.

El controlador ODBC 1.1.15 incluye los siguientes cambios:

- Corrige un problema con las segundas llamadas a `SQLParamData()`.

Para obtener más información sobre estos cambios y, a fin de descargar los nuevos controladores, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con JDBC](#) y [Conexión a Amazon Athena con ODBC](#).

## Notas de la versión de Athena para 2021

### 26 de noviembre de 2021

Publicado el 26/11/2021

Athena anuncia la versión preliminar pública de las transacciones ACID de Athena, que agrega operaciones de escritura, eliminación, actualización y viaje en el tiempo al lenguaje de manipulación de datos SQL (DML) de Athena. Las transacciones ACID de Athena permiten que varios usuarios simultáneos hagan modificaciones fiables a nivel de fila en los datos de Amazon S3. Al estar basadas en el formato de tabla de [Apache Iceberg](#), las transacciones ACID de Athena son compatibles con otros servicios y motores, como [Amazon EMR](#) y [Apache Spark](#), que también admiten los formatos de tabla de Iceberg.

Las transacciones ACID de Athena y la conocida sintaxis SQL simplifican las actualizaciones de los datos empresariales y normativos. Por ejemplo, para responder a una solicitud de borrado de datos, puede llevar a cabo una operación DELETE de SQL. Para hacer correcciones manuales de registros, puede utilizar una sola instrucción UPDATE. Para recuperar los datos eliminados recientemente, puede emitir consultas de viaje en el tiempo mediante una instrucción SELECT. Las transacciones de Athena están disponibles a través de la consola de Athena, las operaciones API y los controladores ODBC y JDBC.

Para obtener más información, consulte [Uso de las transacciones ACID de Athena](#).

## 24 de noviembre de 2021

Publicado el 24/11/2021

Athena anuncia la compatibilidad con la lectura y escritura de datos ORC, Parquet y archivos de texto comprimidos en [ZStandard](#). Athena utiliza el nivel 3 de compresión Zstandard al escribir datos comprimidos en ZStandard.

Para obtener información sobre la compresión de datos en Athena, consulte [Compatibilidad con la compresión de Athena](#).

## 22 de noviembre de 2021

Publicado el 22/11/2021

A partir de ahora, puede administrar los flujos de trabajo de AWS Step Functions desde la consola de Amazon Athena, lo que facilita la creación de canalizaciones de procesamiento de datos escalables, la ejecución de consultas basadas en la lógica empresarial personalizada, la automatización de tareas administrativas y de alertas, etc.

Step Functions ahora está integrado en la consola actualizada de Athena y puede utilizarla para ver un diagrama de flujo de trabajo interactivo de las máquinas de estado que invocan Athena. Para comenzar a trabajar, seleccione Flujos de trabajo del panel de navegación izquierdo. Si tiene máquinas de estado existentes con consultas de Athena, seleccione una máquina de estado para ver un diagrama interactivo del flujo de trabajo. Si es la primera vez que utiliza Step Functions, puede comenzar con el lanzamiento de un proyecto de ejemplo desde la consola de Athena y su personalización para que se adapte a sus casos de uso.

Para obtener más información, consulte [Creación y orquestación de canalizaciones de ETL con Amazon Athena y AWS Step Functions](#), o bien consulte la [Documentación de Step Functions](#).

## 18 de noviembre de 2021

Publicado el 18/11/2021

Athena anuncia nuevas características y mejoras.

- Compatibilidad con vertido en disco para consultas de agregación que contienen DISTINCT, ORDER BY o ambas, como en el siguiente ejemplo:

```
SELECT array_agg(orderstatus ORDER BY orderstatus)
FROM orders
GROUP BY orderpriority, custkey
```

- Se solucionaron los problemas de gestión de la memoria en las consultas que utilizan DISTINCT. Para evitar mensajes de error como Consultar recursos agotados en este factor de escala cuando utilice consultas DISTINCT, elija las columnas que tengan una cardinalidad baja para DISTINCT o reduzca el tamaño de los datos de la consulta.
- En las consultas SELECT COUNT(\*) que no especifican una columna específica, se mejoró el rendimiento y el uso de la memoria al mantener solo el recuento sin almacenamiento en búfer de filas.
- Se presentaron las siguientes funciones de cadena.
  - `translate(source, from, to)`: devuelve la cadena `source` y los caracteres encontrados en la cadena `from` reemplazados por los caracteres correspondientes en la cadena `to`. Si la cadena `from` contiene duplicados, solo se utiliza la primera. Si el carácter `source` no existe en la cadena `from`, el carácter `source` se copia sin traducción. Si el índice del carácter coincidente en la cadena `from` es mayor que la longitud de la cadena `to`, el carácter se omite de la cadena resultante.
  - `concat_ws(string0, array(varchar))`: devuelve la concatenación de elementos de la matriz mediante `string0` como separador. Si `string0` es NULL, el valor devuelto es NULL. Se omiten los valores NULL de la matriz.
- Se corrigió un error por el que no se podían hacer consultas al intentar acceder a un subcampo que faltaba en una `struct`. Las consultas devuelven ahora un valor NULL para el subcampo que falte.
- Se corrigió un problema de hash incoherente para el tipo de datos decimales.
- Se corrigió un problema que provocaba que los recursos se agotaran cuando había demasiadas columnas en una partición.

## 17 de noviembre de 2021

Publicado el 17/11/2021

[Amazon Athena](#) admite ahora la indexación de particiones para acelerar las consultas en tablas particionadas en [AWS Glue Data Catalog](#).

Al consultar las tablas particionadas, Athena recupera y filtra las particiones de tabla disponibles en el subconjunto pertinente para la consulta. A medida que se agregan nuevos datos y particiones, se necesita más tiempo para procesar las particiones y el tiempo de ejecución de consulta puede aumentar. Para optimizar el procesamiento de particiones y mejorar el rendimiento de las consultas en las tablas de muchas particiones, Athena ahora admite los [índices de particiones de AWS Glue](#).

Para obtener más información, consulte [Indexación y filtrado de particiones de AWS Glue](#).

## 16 de noviembre de 2021

Publicado el 16/11/2021

La nueva y mejorada consola de [Amazon Athena](#) ya está disponible de manera general en regiones comerciales y de GovCloud de AWS donde [Athena está disponible](#). La nueva consola de Athena admite todas las características de la consola anterior, pero con una interfaz moderna y fácil de usar e incluye nuevas características que mejoran la experiencia de desarrollar consultas, analizar datos y administrar el uso. Ahora puede hacer lo siguiente:

- Ir a varias pestañas de consulta, reorganizarlas o cerrarlas desde una barra de pestañas de consulta rediseñada.
- Leer y editar consultas con más facilidad gracias a la mejora del formato de texto y SQL.
- Copiar los resultados de las consultas en el portapapeles, además de descargar el conjunto de resultados completo.
- Ordenar el historial de consultas, las consultas guardadas y los grupos de trabajo y elegir qué columnas mostrar u ocultar.
- Utilizar una interfaz simplificada para configurar orígenes de datos y grupos de trabajo con menos clics.
- Establecer preferencias para mostrar los resultados de las consultas, el historial de consultas, el ajuste de líneas y más.
- Aumentar la productividad con métodos abreviados de teclado nuevos y mejorados y documentación de productos incrustada.

Con el anuncio de hoy, la [consola rediseñada](#) ahora es la opción predeterminada. Para contarnos su experiencia, elija Comentarios en la esquina inferior izquierda de la consola.

Si lo desea, puede utilizar la consola anterior. Para ello, inicie sesión en su Cuenta de AWS, elija Amazon Athena y anule la selección New Athena experience (Nueva experiencia de Athena) del panel de navegación de la izquierda.

## 12 de noviembre de 2021

Publicado el 12/11/2021

Ahora puede utilizar Amazon Athena para ejecutar consultas federadas en orígenes de datos ubicados en una cuenta de AWS aparte de la suya. Hasta el día de hoy, la consulta de estos datos requería que el origen de datos y su conector usaran la misma Cuenta de AWS que el usuario que consultó los datos.

Como administrador de datos, puede habilitar las consultas federadas entre cuentas. Para ello, comparta el conector de datos con la cuenta de un analista de datos. Como analista de datos, puede agregar un conector de datos que un administrador de datos compartió con usted en su cuenta. Los cambios de configuración del conector de la cuenta de origen se aplican automáticamente al conector compartido.

Para obtener información sobre cómo habilitar las consultas federadas entre cuentas, consulte [Habilitación de las consultas federadas entre cuentas](#). Para obtener más información sobre las consultas de orígenes federados, consulte [Uso de consulta federada de Amazon Athena](#) y [Ejecución de consultas federadas](#).

## 2 de noviembre de 2021

Publicado el 02/11/2021

Ahora puede utilizar la instrucción EXPLAIN ANALYZE en Athena para ver el plan de ejecución distribuido y el costo de cada operación de las consultas SQL.

Para obtener más información, consulte [Uso de EXPLAIN y EXPLAIN ANALYZE en Athena](#).

## 29 de octubre de 2021

Publicado el 29/10/2021

Athena publica los controladores JDBC 2.0.25 y ODBC 1.1.13 y anuncia características y mejoras.

## Controladores JDBC y ODBC

Se lanzaron los controladores JDBC 2.0.25 y ODBC 1.1.13 para Athena. Ambos controladores ofrecen compatibilidad con la autenticación multifactor SAML del navegador que se puede configurar para que funcione con cualquier proveedor SAML 2.0.

El controlador JDBC 2.0.25 incluye los siguientes cambios:

- Compatibilidad con la autenticación SAML de navegador. El controlador incluye un complemento SAML de navegador que se puede configurar para que funcione con cualquier proveedor SAML 2.0.
- Compatibilidad con llamadas a la API AWS Glue. Puede utilizar el parámetro `GlueEndpointOverride` para anular el punto de conexión de AWS Glue.
- Se cambió la ruta de la clase `com.simba.athena.amazonaws` por `com.amazonaws`.

El controlador ODBC 1.1.13 incluye los siguientes cambios:

- Compatibilidad con la autenticación SAML de navegador. El controlador incluye un complemento SAML de navegador que se puede configurar para que funcione con cualquier proveedor SAML 2.0. Para ver un ejemplo de cómo utilizar el complemento SAML del navegador con el controlador ODBC, consulte [Configuración del inicio de sesión único con ODBC, SAML 2.0 y el proveedor de identidades Okta](#).
- Ahora puede configurar la duración de la sesión del rol al utilizar ADFS, Azure AD o Azure AD de navegador para la autenticación.

Para obtener más información sobre estos y otros cambios y, a fin de descargar los nuevos controladores, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con JDBC](#) y [Conexión a Amazon Athena con ODBC](#).

## Características y mejoras

Athena anuncia las siguientes características y mejoras.

- Se presentó una nueva regla de optimización para evitar análisis de tablas duplicados en determinados casos.



## 4 de octubre de 2021

Publicado el 04/10/2021

Athena anuncia las siguientes características y mejoras.

- **SQL OFFSET:** la cláusula `OFFSET` de SQL ahora es compatible en las instrucciones `SELECT`. Para obtener más información, consulte [SELECT](#).
- **Métricas de uso de CloudWatch:** Athena publica ahora la métrica `ActiveQueryCount` en el espacio de nombres `AWS/Usage`. Para obtener más información, consulte [Supervisión de las métricas de uso de Athena](#).
- **Planificación de consultas:** se corrigió un error que, en raras ocasiones, podía provocar tiempos de espera de planificación de consultas.

## 16 de septiembre de 2021

Publicado el 16/09/2021

Athena anuncia las nuevas características y mejoras que se indican a continuación.

### Características

- Se agregó compatibilidad para especificar la compresión de archivos de texto y JSON en CTAS mediante la propiedad de tablas `write_compression`. También puede especificar la propiedad `write_compression` en CTAS para los formatos Parquet y ORC. Para obtener más información, consulte [Propiedades de la tabla CTAS](#).
- El formato de compresión BZIP2 ahora es compatible para escribir archivos de texto y archivos JSON. Para obtener más información sobre los formatos de compresión en Athena, consulte [Compatibilidad con la compresión de Athena](#).

### Mejoras

- Se corrigió un error por el que la información de identidad no se podía enviar a la función de Lambda de UDF.
- Se corrigió un problema de inserción de predicados con condiciones de filtro disyuntivas.
- Se corrigió un problema de hash de los tipos decimales.
- Se corrigió un problema de recopilación de estadísticas innecesarias.

- Se eliminó un mensaje de error incoherente.
- Se mejoró el rendimiento de las uniones de difusión mediante la aplicación de la eliminación de particiones dinámica en el nodo de trabajo.
- Para consultas federadas:
  - Se alteró la configuración para reducir la aparición de errores CONSTRAINT\_VIOLATION en las consultas federadas.

## 15 de septiembre de 2021

Publicado el 15/09/2021

Ahora puede utilizar una consola de Amazon Athena rediseñada (versión preliminar). Se publicó un nuevo controlador JDBC de Athena.

### Vista previa de la consola de Athena

A partir de ahora, puede utilizar una consola rediseñada de [Amazon Athena](#) (versión preliminar) desde cualquier Región de AWS donde Athena está disponible. La nueva consola admite todas las características de la consola existente, pero desde una interfaz moderna y fácil de utilizar.

Para cambiar a la nueva [consola](#), inicie sesión en su Cuenta de AWS y elija Amazon Athena. Desde la barra de navegación de la consola de AWS, elija Cambiar a la nueva consola. Para volver a la consola predeterminada, anule la selección Nueva experiencia Athena desde el panel de navegación de la izquierda.

Comience a trabajar con la nueva [consola](#) hoy. Elija Comentarios en la esquina inferior izquierda para contarnos su experiencia.

### Controlador JDBC de Athena 2.0.24

Athena anuncia la disponibilidad del controlador JDBC versión 2.0.24 para Athena. Esta versión actualiza la compatibilidad de proxy de todos los proveedores de credenciales. El controlador ahora admite la autenticación proxy de todos los hosts que no son compatibles con la propiedad de conexión NonProxyHosts.

Para mayor comodidad, esta versión incluye descargas del controlador JDBC con y sin el SDK AWS. Esta versión del controlador JDBC le permite tener el SDK de AWS y el controlador JDBC de Athena integrados en el proyecto.

Para obtener más información y descargar el nuevo controlador, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con JDBC](#).

## 31 de agosto de 2021

Publicado el 31/08/2021

Athena anuncia las siguientes mejoras de características y correcciones de errores.

- Mejoras de la federación de Athena: Athena agregó soporte para tipos de mapas y mejor soporte para tipos complejos como parte del [SDK de Athena Query Federation](#). Esta versión incluye también algunas mejoras de memoria y optimizaciones de rendimiento.
- Nuevas categorías de error: se presentaron las categorías de error USER y SYSTEM en los mensajes de error. Estas categorías lo ayudan a distinguir errores que puede corregir usted mismo (USER) y errores que pueden requerir asistencia del soporte técnico de Athena (SYSTEM).
- Mensajería de error de consultas federadas: actualización de categorizaciones de USER\_ERROR para errores relacionados con consultas federadas.
- JOIN: se corrigieron errores relacionados con el vertido en disco y problemas de memoria para mejorar el rendimiento y reducir los errores de memoria en operaciones JOIN.

## 12 de agosto de 2021

Publicado el 12/08/2021

Se publicó el controlador ODBC 1.1.12 para Athena. Esta versión corrige problemas relacionados con `SQLPrepare()`, `SQLGetInfo()` y `EndpointOverride`.

Para descargar el nuevo controlador, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con ODBC](#).

## 6 de agosto de 2021

Publicado el 06/08/2021

Amazon Athena anuncia la disponibilidad de Athena y sus [características](#) en la región Asia-Pacífico (Osaka).

Esta versión amplía la disponibilidad de Athena en Asia-Pacífico e incluye Asia-Pacífico (Hong Kong), Asia-Pacífico (Bombay), Asia-Pacífico (Osaka), Asia-Pacífico (Seúl), Asia-Pacífico (Singapur),

Asia-Pacífico (Singapur), Asia-Pacífico (Sídney) y Asia-Pacífico (Tokio). Para obtener una lista completa de los Servicios de AWS disponibles en estas y otras regiones, consulte la [Lista de servicios por Región de AWS](#).

## 5 de agosto de 2021

Publicado el 05/08/2021

Puede utilizar la instrucción UNLOAD para escribir la salida de una consulta SELECT a los formatos PARQUET, ORC, AVRO y JSON.

Para obtener más información, consulte [UNLOAD](#).

## 30 de julio de 2021

Publicado el 30/07/2021

Athena anuncia las siguientes mejoras de características y correcciones de errores.

- Filtrado dinámico y poda de particiones: las mejoras aumentan el rendimiento y reducen la cantidad de datos analizados en determinadas consultas, como en el siguiente ejemplo.

En este ejemplo, se supone que Table\_B es una tabla no particionada que tiene tamaños de archivo que suman menos de 20 MB. Para consultas como esta, se leen menos datos de Table\_A y la consulta se completa más rápidamente.

```
SELECT *
FROM Table_A
JOIN Table_B ON Table_A.date = Table_B.date
WHERE Table_B.column_A = "value"
```

- ORDER BY with LIMIT, DISTINCT with LIMIT: mejoras en el rendimiento de consultas que utilizan ORDER BY o DISTINCT seguido de una cláusula LIMIT.
- Archivos S3 Glacier Deep Archive: cuando Athena consulta una tabla que contiene una combinación de [archivos S3 Glacier Deep Archive](#) y archivos que no son S3 Glacier, Athena ahora omite los archivos de S3 Glacier Deep Archive por usted. Anteriormente, era necesario mover manualmente estos archivos desde la ubicación de la consulta, o la consulta generaba un error. Si desea utilizar Athena para consultar objetos en el almacenamiento de S3 Glacier Deep Archive, debe restaurarlos. Para obtener más información, consulte [Restaurar un objeto archivado](#) en la Guía del usuario de Amazon S3.

- Se corrigió un error en el que archivos vacíos creados por la [propiedad de tabla](#) CTAS bucketed\_by no se cifraron correctamente.

## 21 de julio de 2021

Publicado el 21/07/2021

Con la publicación de julio de 2021 del [Escritorio Microsoft Power BI](#), es posible generar informes y paneles con un conector de origen de datos nativo para Amazon Athena. El conector de Amazon Athena está disponible como conector estándar en Power BI, admite [DirectQuery](#), y permite el análisis de conjuntos de datos grandes y la actualización de contenido a través de la [Puerta de enlace de Power BI](#).

Dado que el conector utiliza el nombre de origen de datos (DSN) ODBC existente para conectarse y ejecutar consultas en Athena, requiere el controlador ODBC de Athena. Para descargar el controlador ODBC más reciente, consulte [Conexión a Amazon Athena con ODBC](#).

Para obtener más información, consulte [Uso del conector Power BI de Amazon Athena](#).

## 16 de julio de 2021

Publicado el 16/07/2021

Amazon Athena actualizó su integración con Apache Hudi. Hudi es un marco de administración de datos de código abierto que se utiliza para simplificar el procesamiento incremental de datos en lagos de datos de Amazon S3. La integración actualizada le permite utilizar Athena para consultar tablas Hudi 0.8.0 administradas a través de Amazon EMR, Apache Spark, Apache Hive u otros servicios compatibles. Además, Athena ahora admite dos características adicionales: consultas de instantáneas en tablas Fusionar al leer (MoR, Merge-on-Read) y soporte de lectura en tablas de arranque.

Apache Hudi proporciona procesamiento de datos a nivel de registro que puede ayudarlo a simplificar el desarrollo de canalizaciones de captura de datos modificados (CDC), cumplir con las actualizaciones y eliminaciones impulsadas por el RGPD y administrar mejor los datos de transmisión desde sensores o dispositivos que requieren inserción de datos y actualizaciones de eventos. La versión 0.8.0 facilita la migración de tablas de Parquet grandes a Hudi sin copiar datos, de modo que pueda consultarlas y analizarlas a través de Athena. Puede utilizar la nueva compatibilidad de Athena para consultas de instantáneas para tener vistas casi en tiempo real de las actualizaciones de las tablas de transmisión.

Para obtener más información sobre el uso de Hudi con Athena, consulte [Uso de Athena para consultar conjuntos de datos de Apache Hudi](#).

## 8 de julio de 2021

Publicado el 08/07/2021

Se publicó el controlador ODBC 1.1.11 para Athena. El controlador ODBC ahora puede autenticar la conexión mediante un JSON Web Token (JWT). En Linux, el valor predeterminado de la propiedad de grupo de trabajo se estableció en Principal.

Para obtener más información y descargar el nuevo controlador, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con ODBC](#).

## 1 de julio de 2021

Publicado el 01/07/2021

El 1 de julio de 2021, finalizó la gestión especial de grupos de trabajo de vista previa. Aunque los grupos de trabajo AmazonAthenaPreviewFunctionality conservan su nombre, ya no tienen un estado especial. Puede continuar usando los grupos de trabajo AmazonAthenaPreviewFunctionality para ver, modificar, organizar y ejecutar consultas. Sin embargo, las consultas que utilizan características que anteriormente estaban en vista previa ahora están sujetas a los términos y condiciones de facturación estándar de Athena. Para obtener información sobre la facturación, consulte [Precios de Amazon Athena](#).

## 23 de junio de 2021

Publicado el 23/06/2021

Se publicaron los controladores JDBC 2.0.23 y ODBC 1.1.10 para Athena. Ambos controladores ofrecen un mejor rendimiento de lectura, instrucciones [EXPLAIN](#) de soporte y [consultas parametrizadas](#).

Las instrucciones EXPLAIN muestran el plan de ejecución lógico o distribuido de una consulta SQL. Las consultas parametrizadas permiten que la misma consulta se utilice varias veces con diferentes valores suministrados en tiempo de ejecución.

La versión de JDBC también agrega compatibilidad con Active Directory Federation Services 2019 y una opción de anulación de puntos de conexión personalizada para AWS STS. La versión ODBC corrige un problema con las credenciales de perfil de IAM.

Para obtener más información y descargar el nuevo controlador, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con JDBC](#) y [Conexión a Amazon Athena con ODBC](#).

## 12 de mayo de 2021

Publicado el 12/05/2021

Ahora puede utilizar Amazon Athena para registrar un catálogo AWS Glue desde una cuenta que no sea la suya. Una vez configurados los permisos de IAM necesarios para AWS Glue, puede utilizar Athena para ejecutar consultas entre cuentas.

Para obtener más información, consulte [Registro de un AWS Glue Data Catalog de otra cuenta y Acceso entre cuentas a los catálogos de datos de AWS Glue](#).

## 10 de mayo de 2021

Publicado el 10/05/2021

Se publicó la versión 1.1.9.1001 del controlador ODBC para Athena. Esta versión corrige un problema con el tipo de autenticación BrowserAzureAD cuando se utiliza Azure Active Directory (AD).

Para descargar los nuevos controladores, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con ODBC](#).

## 5 de mayo de 2021

Publicado el 05/05/2021

Ahora puede utilizar el conector de Amazon Athena Vertica en consultas federadas para consultar orígenes de datos Vertica desde Athena. Por ejemplo, puede ejecutar consultas analíticas a través de un almacenamiento de datos en Vertica y un lago de datos en Amazon S3.

Para implementar el conector Athena Vertica, visite la página [AthenaVerticaConnector](#) en AWS Serverless Application Repository.

El conector de Amazon Athena Vertica expone varias opciones de configuración a través de variables de entorno Lambda. Para obtener información sobre las opciones de configuración, los

parámetros, las cadenas de conexión, la implementación y las limitaciones, consulte [Conector para Vertica de Amazon Athena](#).

Para obtener información detallada acerca del uso de conector Vertica, consulte [Consulta de un origen de datos Vertica en Amazon Athena mediante el SDK de consulta federada de Athena](#) en el Blog de macrodatos de AWS.

## 30 de abril de 2021

Publicado el 30/04/2021

Se publicaron los controladores JDBC 2.0.21 y ODBC 1.1.9 para Athena. Ambas versiones admiten la autenticación SAML con Azure Active Directory (AD) y la autenticación SAML con PingFederate. La versión JDBC admite también consultas parametrizadas. Para obtener información acerca de las consultas parametrizadas en Athena, consulte [Uso de consultas parametrizadas](#).

Para descargar los nuevos controladores, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con JDBC](#) y [Conexión a Amazon Athena con ODBC](#).

## 29 de abril de 2021

Publicado el 29/04/2021

Amazon Athena anuncia disponibilidad de la versión 2 del motor Athena en las regiones China (Pekín) y China (Ningxia).

Para obtener información acerca de la versión 2 del motor Athena, consulte [Versión 2 del motor Athena](#).

## 26 de abril de 2021

Publicado el 26/04/2021

Las funciones de valor de ventana en la versión 2 del motor Athena ahora admiten IGNORE NULLS y RESPECT NULLS.

Para obtener más información, consulte [Funciones de valor](#) en la documentación de Presto.

## 21 de abril de 2021

Publicado el 21/04/2021



Amazon Athena anuncia disponibilidad de la versión 2 del motor Athena en las regiones Europa (Milán) y África (Ciudad del Cabo).

Para obtener información acerca de la versión 2 del motor Athena, consulte [Versión 2 del motor Athena](#).

## 5 de abril de 2021

Publicado el 05/04/2021

### Instrucción EXPLAIN

Ahora puede utilizar la instrucción EXPLAIN en Athena para ver el plan de ejecución de las consultas SQL.

Para obtener más información, consulte [Uso de EXPLAIN y EXPLAIN ANALYZE en Athena y Descripción de los resultados de la instrucción EXPLAIN de Athena](#).

### Modelos de SageMaker Machine Learning en consultas SQL

La inferencia del modelo de machine learning con Amazon SageMaker ahora está disponible de forma general para Amazon Athena. Utilice modelos de machine learning en consultas SQL para simplificar tareas complejas, como la detección de anomalías, el análisis de cohortes de clientes y las predicciones de serie temporal al invocar una función en una consulta SQL.

Para obtener más información, consulte [Uso de Machine Learning \(ML\) con Amazon Athena](#).

### Funciones definidas por el usuario (UDF)

Las funciones definidas por el usuario (UDF) ya están disponibles en general para Athena. Utilice UDF para aprovechar las funciones personalizadas que procesan registros o grupos de registros en una sola consulta SQL.

Para obtener más información, consulte [Consulta con funciones definidas por el usuario](#).

## 30 de marzo de 2021

Publicado el 30/03/2021

Amazon Athena anuncia disponibilidad de la versión 2 del motor Athena en las regiones Asia-Pacífico (Hong Kong) y Medio Oriente (Baréin).

Para obtener información acerca de la versión 2 del motor Athena, consulte [Versión 2 del motor Athena](#).

## 25 de marzo de 2021

Publicado el 25/03/2021

Amazon Athena anuncia la disponibilidad de la versión 2 del motor Athena en la región de Europa (Estocolmo).

Para obtener información acerca de la versión 2 del motor Athena, consulte [Versión 2 del motor Athena](#).

## 5 de marzo de 2021

Publicado el 05/03/2021

Amazon Athena anuncia disponibilidad de la versión 2 del motor Athena en las regiones Canadá (centro), Europa (Fráncfort) y América del Sur (São Paulo).

Para obtener información acerca de la versión 2 del motor Athena, consulte [Versión 2 del motor Athena](#).

## 25 de febrero de 2021

Publicado el 25/02/2021

Amazon Athena anuncia disponibilidad general de la versión 2 del motor Athena en las regiones Asia-Pacífico (Seúl), Asia-Pacífico (Singapur), Asia-Pacífico (Sídney), Europa (Londres) y Europa (París).

Para obtener información acerca de la versión 2 del motor Athena, consulte [Versión 2 del motor Athena](#).

## Notas de la versión de Athena para 2020

### 16 de diciembre de 2020

Publicado el 16/12/2020

Amazon Athena anuncia la disponibilidad de la versión 2 del motor Athena, la consulta federada de Athena y AWS PrivateLink en regiones adicionales.

## Versión 2 del motor Athena y consulta federada de Athena

Amazon Athena anuncia disponibilidad general de la versión 2 del motor Athena y la consulta federada de Athena en las regiones Asia-Pacífico (Bombay), Asia-Pacífico (Tokio), Europa (Irlanda) y Oeste de EE. UU. (Norte de California). La versión 2 del motor Athena y las consultas federadas ya están disponibles en las regiones Este de EE. UU. (Norte de Virginia), Este de EE. UU. (Ohio) y Oeste de EE. UU. (Oregón).

Para obtener más información, consulte [Versión 2 del motor Athena](#) y [Uso de consulta federada de Amazon Athena](#).

## AWS PrivateLink

AWS PrivateLink para Athena ahora está disponible en la región Europa (Estocolmo). Para obtener información acerca de AWS PrivateLink para Athena, consulte [Conexión a Amazon Athena mediante un punto de conexión de VPC de tipo interfaz](#).

## 24 de noviembre de 2020

Publicado el 24/11/2020

Se publicaron los controladores JDBC 2.0.16 y ODBC 1.1.6 para Athena. Estas versiones, en el nivel de cuenta, admiten la autenticación multifactor (MFA) de Okta Verify. También puede utilizar la MFA de Okta para configurar la autenticación SMS y la autenticación de Google Authenticator como factores.

Para descargar los nuevos controladores, las notas de la versión y la documentación, consulte [Conexión a Amazon Athena con JDBC](#) y [Conexión a Amazon Athena con ODBC](#).

## 11 de noviembre de 2020

Publicado el 11/11/2020

Amazon Athena anuncia disponibilidad general de la versión 2 del motor Athena y las consultas federadas en las regiones Este de EE. UU. (Norte de Virginia), Este de EE. UU. (Ohio) y Oeste de EE. UU. (Oregón).

## Versión 2 del motor Athena

Amazon Athena anuncia disponibilidad general de una nueva versión del motor de consultas, la versión 2 del motor Athena, en las regiones Este de EE. UU. (Norte de Virginia), Este de EE. UU. (Ohio) y Oeste de EE. UU. (Oregón).

La versión 2 del motor Athena incluye mejoras de rendimiento y nuevas características, como compatibilidad con la evolución de esquemas para datos con formato Parquet, funciones geoespaciales adicionales, compatibilidad con la lectura de esquemas anidados para reducir costos y mejoras de rendimiento en operaciones JOIN y AGGREGATE.

- Para obtener información acerca de las mejoras, los cambios sustanciales y las correcciones de errores, consulte [Versión 2 del motor Athena](#).
- Para obtener información acerca de cómo actualizar, consulte [Cambio de las versiones del motor Athena](#).
- Para obtener información acerca de las pruebas de consultas, consulte [Pruebas de consultas antes de una actualización de la versión del motor](#).

## Consultas de SQL federadas

Ahora puede utilizar la consulta federada de Athena en las regiones Este de EE. UU. (Norte de Virginia), Este de EE. UU. (Ohio) y Oeste de EE. UU. (Oregón) sin utilizar el grupo de trabajo AmazonAthenaPreviewFunctionality.

Utilice consultas de SQL federadas para ejecutar consultas de SQL en orígenes de datos relacionales, no relacionales, de objetos y personalizados. Con las consultas federadas, puede enviar una sola consulta SQL que examina los datos de varios orígenes alojados en la nube o que se ejecutan en las instalaciones.

La ejecución de análisis en datos repartidos entre aplicaciones puede ser compleja y consumir mucho tiempo por las siguientes razones:

- Los datos necesarios para un análisis suelen distribuirse entre almacenes de datos relacionales, en memoria, de clave-valor, de documentos, de búsqueda, de gráficos, de objetos, de series temporales y de libro mayor.
- A la hora de analizar los datos en estos orígenes, los analistas crean canalizaciones complejas para realizar extracciones, transformaciones y cargas en un almacenamiento de datos a fin de que los datos puedan consultarse.

- Acceder a los datos desde varios orígenes requiere aprender nuevos lenguajes de programación y constructos de acceso a los datos.

Las consultas SQL federadas en Athena eliminan esta complejidad, ya que permiten a los usuarios consultar los datos de forma local independientemente de su ubicación. Los analistas pueden utilizar constructos de SQL conocidos para datos JOIN de varios orígenes con el objetivo de realizar un análisis rápido y almacenar los resultados en Amazon S3 para su posterior uso.

### Conectores de origen de datos

Para procesar consultas federadas, Athena utiliza conectores de origen de datos de Athena que se ejecutan en [AWS Lambda](#). Los siguientes conectores preconstruidos de código abierto fueron escritos y probados por Athena. Puede utilizarlos para ejecutar consultas SQL en Athena en sus orígenes de datos correspondientes.

- [CloudWatch](#)
- [Métricas de CloudWatch](#)
- [DocumentDB](#)
- [DynamoDB](#)
- [OpenSearch](#)
- [HBase](#)
- [Neptune](#)
- [Redis](#)
- [Timestream](#)
- [TPC Benchmark DS \(TPC-DS\)](#)

### Conectores de orígenes de datos personalizados

Mediante el [SDK de Athena Query Federation](#), los desarrolladores pueden crear conectores a cualquier origen de datos para permitir que Athena ejecute consultas SQL en ese origen de datos. Athena Query Federation Connector extiende los beneficios de las consultas federadas más allá de los conectores provistos por AWS. Dado que los conectores se ejecutan en AWS Lambda, no es necesario administrar la infraestructura ni planificar la ampliación a las demandas máximas.

## Siguientes pasos

- Para obtener más información acerca de la característica de consulta federada, consulte [Uso de consulta federada de Amazon Athena](#).
- Para empezar a utilizar un conector existente, consulte [Implementación de un conector y conexión a un origen de datos](#).
- Para obtener información sobre cómo crear su propio conector de origen de datos mediante el SDK de Athena Query Federation, consulte [Ejemplo de conector Athena](#) en GitHub.

## 22 de octubre de 2020

Publicado el 22/10/2020

Llame a Athena con AWS Step Functions. AWS Step Functions puede controlar algunos Servicios de AWS directamente mediante [Amazon States Language](#). Puede utilizar Step Functions con Athena para iniciar y detener la ejecución de consultas, obtener resultados de consultas, ejecutar consultas de datos ad hoc o programadas y recuperar resultados de lagos de datos en Amazon S3.

Para obtener más información, consulte [Llamar a Athena con Step Functions](#) en la Guía para desarrolladores de AWS Step Functions.

## 29 de julio de 2020

Publicado el 29/07/2020

Se publicó la versión 2.0.13 del controlador JDBC. Esta versión admite el uso de varios [catálogos de datos registrados en Athena](#), el servicio Okta para autenticación y las conexiones a puntos de conexión de VPC.

Para descargar y utilizar la nueva versión del controlador, consulte [Conexión a Amazon Athena con JDBC](#).

## 9 de julio de 2020

Publicado el 09/07/2020

Amazon Athena agrega compatibilidad para consultar conjuntos de datos Hudi compactados y agrega el recurso AWS CloudFormation de `AWS::Athena::DataCatalog` para crear, actualizar o eliminar catálogos de datos que registre en Athena.

## Consulta de conjuntos de datos de Apache Hudi

Apache Hudi es un marco de administración de datos de código abierto que simplifica el procesamiento incremental de datos. Amazon Athena ahora admite la consulta de la vista optimizada para lectura de un conjunto de datos Apache Hudi en el lago de datos basado en Amazon S3.

Para obtener más información, consulte [Uso de Athena para consultar conjuntos de datos de Apache Hudi](#).

## Recurso AWS CloudFormation del catálogo de datos

Para utilizar la [característica de consulta federada](#) de Amazon Athena para consultar cualquier origen de datos, primero debe registrar el catálogo de datos en Athena. Ahora puede usar el recurso AWS CloudFormation de `AWS::Athena::DataCatalog` para crear, actualizar o eliminar catálogos de datos que registra en Athena.

Para obtener más información, consulte [AmazonWebService::Athena::DataCatalog](#) en la Guía del usuario de AWS CloudFormation.

### 1 de junio de 2020

Publicado el 01/06/2020

## Uso de metaalmacenes de Apache Hive con Amazon Athena

Ahora puede conectar Athena a uno o más metaalmacenes de Apache Hive además de AWS Glue Data Catalog con Athena.

Para conectarse a un metaalmacén de Hive autoalojado, necesita un conector de metaalmacén de Athena Hive. Athena ofrece un conector de [implementación de referencia](#) que puede utilizar. El conector se ejecuta como una función de AWS Lambda en su cuenta.

Para obtener más información, consulte [Uso del conector de datos de Athena para metaalmacén externo de Hive](#).

### 21 de mayo de 2020

Publicado el 21/05/2020

Amazon Athena agrega compatibilidad para la proyección de particiones. Utilice la proyección de particiones para acelerar el procesamiento de consultas de tablas altamente particionadas y

automatizar la administración de particiones. Para obtener más información, consulte [Proyección de particiones con Amazon Athena](#).

## 1 de abril de 2020

Publicado el 01/04/2020

Además de la región Este de EE. UU. (Norte de Virginia), las características de [consulta federada](#), [funciones definidas por el usuario \(UDF\)](#), [inferencia de machine learning](#) y [metaalmacén externo de Hive](#) de Amazon Athena ya están disponibles en versión preliminar en las regiones Asia-Pacífico (Bombay), Europa (Irlanda) y Oeste de EE. UU. (Oregón).

## 11 de marzo de 2020

Publicado el 11/03/2020

Amazon Athena ha publicado Amazon EventBridge para transiciones de estado de consulta. Cuando una consulta pasa de un estado a otro (por ejemplo, del estado “en ejecución” a un estado terminal, como “realizado con éxito” o “cancelado”), Athena publica un evento de cambio de estado de consulta en EventBridge. El evento contiene información acerca de la transición del estado de consulta. Para obtener más información, consulte [Supervisión de eventos de Athena con Amazon EventBridge](#).

## 6 de marzo de 2020

Publicado el 06/03/2020

Ahora puede crear y actualizar grupos de trabajo de Amazon Athena mediante el recurso AWS CloudFormation `AWS::Athena::WorkGroup`. Para obtener más información, consulte [AmazonWebService::Athena::WorkGroup](#) en la Guía del usuario de AWS CloudFormation.

# Notas de la versión de Athena para 2019

## 26 de noviembre de 2019

Publicado el 17/12/2019

Amazon Athena agrega compatibilidad para ejecutar consultas SQL en orígenes de datos relacionales, no relacionales, de objetos y personalizados, invocar modelos de machine learning



en consultas SQL, funciones definidas por el usuario (UDF) (vista previa), utilizar metaalmacenes de Apache Hive como catálogo de metadatos con Amazon Athena (vista previa), y cuatro métricas relacionadas con las consultas adicionales.

## Consultas de SQL federadas

Utilice consultas de SQL federadas para ejecutar consultas de SQL en orígenes de datos relacionales, no relacionales, de objetos y personalizados.

Ahora puede utilizar la consulta federada de Athena para analizar datos almacenados en orígenes de datos relacionales, no relacionales, de objetos y personalizados. Con las consultas federadas, puede enviar una sola consulta SQL que examina los datos de varios orígenes alojados en la nube o que se ejecutan en las instalaciones.

La ejecución de análisis en datos repartidos entre aplicaciones puede ser compleja y consumir mucho tiempo por las siguientes razones:

- Los datos necesarios para un análisis suelen distribuirse entre almacenes de datos relacionales, en memoria, de clave-valor, de documentos, de búsqueda, de gráficos, de objetos, de series temporales y de libro mayor.
- A la hora de analizar los datos en estos orígenes, los analistas crean canalizaciones complejas para realizar extracciones, transformaciones y cargas en un almacenamiento de datos a fin de que los datos puedan consultarse.
- Acceder a los datos desde varios orígenes requiere aprender nuevos lenguajes de programación y constructos de acceso a los datos.

Las consultas SQL federadas en Athena eliminan esta complejidad, ya que permiten a los usuarios consultar los datos de forma local independientemente de su ubicación. Los analistas pueden utilizar constructos de SQL conocidos para datos JOIN de varios orígenes con el objetivo de realizar un análisis rápido y almacenar los resultados en Amazon S3 para su posterior uso.

## Conectores de origen de datos

Athena procesa consultas federadas con conectores de origen de datos de Athena que se ejecutan en [AWS Lambda](#). Utilice estos conectores de orígenes de datos de código abierto para ejecutar consultas SQL federadas en Athena en [Amazon DynamoDB](#), [Apache HBase](#), [Amazon Document DB](#), [Amazon CloudWatch](#), [Amazon CloudWatch Metrics](#) y bases de datos relacionales conformes con los requisitos de [JDBC](#) como MySQL, y PostgreSQL con la licencia Apache 2.0.

## Conectores de orígenes de datos personalizados

Mediante el [SDK de Athena Query Federation](#), los desarrolladores pueden crear conectores a cualquier origen de datos para permitir que Athena ejecute consultas SQL en ese origen de datos. Athena Query Federation Connector extiende los beneficios de las consultas federadas más allá de los conectores provistos por AWS. Dado que los conectores se ejecutan en AWS Lambda, no es necesario administrar la infraestructura ni planificar la ampliación a las demandas máximas.

### Disponibilidad de la vista previa

La consulta federada de Athena está disponible en vista previa en la región Este de EE. UU. (Norte de Virginia).

### Siguientes pasos

- Para comenzar la vista previa, siga las instrucciones de las [Preguntas frecuentes sobre las características de la vista previa de Athena](#).
- Para obtener más información sobre la característica de consulta federada, consulte [Uso de consultas federadas de Amazon Athena \(vista previa\)](#).
- Para empezar a utilizar un conector existente, consulte [Implementación de un conector y conexión a un origen de datos](#).
- Para obtener información sobre cómo crear su propio conector de origen de datos mediante el SDK de Athena Query Federation, consulte [Ejemplo de conector Athena](#) en GitHub.

## Invocar modelos de Machine Learning en consultas de SQL

Ahora puede invocar modelos de machine learning para inferirlos directamente desde sus consultas de Athena. La posibilidad de utilizar modelos de machine learning en consultas de SQL permite que tareas complejas, como la detección de anomalías, el análisis de cohortes de clientes y las predicciones de ventas, sean tan simples como invocar una función en una consulta de SQL.

### Modelos de ML

Puede utilizar más de una docena de algoritmos de machine learning integrados que proporciona [Amazon SageMaker](#), entrenar sus propios modelos, o buscar y suscribirse a paquetes de modelos desde [AWS Marketplace](#) e implementarlos en los [servicios de alojamiento de Amazon SageMaker](#). No necesita realizar configuraciones adicionales. Puede invocar estos modelos de ML en las consultas de SQL desde la consola de Athena, las [API de Athena](#) y a través del [controlador JDBC de vista previa](#) de Athena.

## Disponibilidad de la vista previa

La funcionalidad de ML de Athena está disponible hoy en vista previa en la región Este de EE. UU. (Norte de Virginia).

### Siguientes pasos

- Para comenzar la vista previa, siga las instrucciones de las [Preguntas frecuentes sobre las características de la vista previa de Athena](#).
- Para obtener más información sobre la característica de machine learning, consulte [Uso de machine learning \(ML\) con Amazon Athena \(vista previa\)](#).

## Funciones definidas por el usuario (UDF) (Vista previa)

Ahora puede escribir funciones escalares personalizadas e invocarlas en sus consultas de Athena. Puede escribir sus UDF en Java utilizando el [SDK de Athena Query Federation](#). Cuando se utiliza una UDF en una consulta SQL enviada a Athena, se invoca y se ejecuta en [AWS Lambda](#). Las UDF se pueden utilizar en las cláusulas SELECT y FILTER de una consulta de SQL. Es posible invocar varias UDF en la misma consulta.

## Disponibilidad de la vista previa

La funcionalidad de UDF de Athena está disponible hoy en vista previa en la región Este de EE. UU. (Norte de Virginia).

### Siguientes pasos

- Para comenzar la vista previa, siga las instrucciones de las [Preguntas frecuentes sobre las características de la vista previa de Athena](#).
- Para obtener más información, consulte [Consultas con funciones definidas por el usuario \(vista previa\)](#).
- Para ver ejemplos de implementaciones de UDF, consulte [Conector de UDF de Amazon Athena en GitHub](#).
- Para obtener información sobre cómo escribir sus propias funciones mediante el SDK de Athena Query Federation, consulte [Creación e implementación de una UDF mediante Lambda](#).

## Uso del metaalmacén de Apache Hive como metacatálogo con Amazon Athena (vista previa)

Ahora puede conectar Athena a uno o más metaalmacenes de Apache Hive además de AWS Glue Data Catalog con Athena.

### Conector Metastore

Para conectarse a un metaalmacén de Hive autoalojado, necesita un conector de metaalmacén de Athena Hive. Athena ofrece un conector de implementación de [referencia](#) que puede utilizar. El conector se ejecuta como una función de AWS Lambda en su cuenta. Para obtener más información, consulte [Uso del conector de datos de Athena para metaalmacén externo de Hive \(vista previa\)](#).

### Disponibilidad de la vista previa

La característica de metaalmacén de Hive está disponible en la región Este de EE. UU. (Norte de Virginia).

### Siguientes pasos

- Para comenzar la vista previa, siga las instrucciones de las [Preguntas frecuentes sobre las características de la vista previa de Athena](#).
- Para obtener más información acerca de esta característica, visite [Uso del conector de datos de Athena para metaalmacén externo de Hive \(vista previa\)](#).

## Nuevas métricas relacionadas con consultas

Athena ahora publica métricas de consulta adicionales que pueden ayudarlo a entender el rendimiento de [Amazon Athena](#). Athena publica métricas relacionadas con las consultas en [Amazon CloudWatch](#). En esta versión, Athena publica las siguientes métricas de consulta adicionales:

- Tiempo de planificación de consultas: el tiempo requerido para planificar la consulta. Esto incluye el tiempo dedicado a recuperar las particiones de tabla del origen de datos.
- Tiempo de cola de consultas: el tiempo que la consulta estuvo en una cola en espera de recursos.
- Tiempo de procesamiento del servicio: el tiempo requerido para escribir los resultados una vez que el motor de consultas finaliza el procesamiento.
- Tiempo total de ejecución: el tiempo que Athena demoró en ejecutar la consulta.

Para utilizar estas métricas de consultas nuevas, puede crear paneles personalizados, establecer alarmas y desencadenadores en las métricas en CloudWatch, o utilizar paneles que se completan con antelación directamente desde la consola de Athena.

### Siguientes pasos

Para obtener más información, consulte [Monitoreo de métricas de Athena con Amazon CloudWatch](#).

## 12 de noviembre de 2019

Publicado el 17/12/2019

Amazon Athena ahora está disponible en la región de Medio Oriente (Baréin).

## 8 de noviembre de 2019

Publicado el 17/12/2019

Amazon Athena ahora está disponible en la región Oeste de EE. UU. (Norte de California) y Europa (París).

## 8 de octubre de 2019

Publicado el 17/12/2019

[Amazon Athena](#) ahora le permite conectarse directamente a Athena mediante un punto de conexión de VPC de tipo interfaz en su nube privada virtual (VPC). Mediante esta característica, puede enviar sus consultas a Athena de forma segura sin requerir una puerta de enlace de Internet en su VPC.

Para crear un punto de conexión de VPC de la interfaz para conectarse a Athena, puede usar la AWS Management Console o AWS Command Line Interface (AWS CLI). Para obtener información sobre la creación de un punto de conexión de tipo interfaz, consulte [Creación de un punto de conexión de interfaz](#).

Cuando utiliza un punto de conexión de VPC de tipo interfaz, la comunicación entre su VPC y las API de Athena es segura y permanece dentro de la red de AWS. No hay costos adicionales de Athena para usar esta característica. Se aplican [cargos](#) al punto de conexión de VPC de la interfaz.

Para obtener más información acerca de esta característica, consulte [Conectarse a Amazon Athena mediante un punto de conexión de VPC de tipo interfaz](#).

## 19 de septiembre de 2019

Publicado el 17/12/2019

Amazon Athena agrega compatibilidad para insertar nuevos datos en una tabla existente mediante la instrucción `INSERT INTO`. Puede insertar filas nuevas en una tabla de destino en función de una instrucción de consulta `SELECT` que se ejecute en una tabla de origen o en función de un conjunto de valores que se provee como parte de la instrucción de consulta. Los formatos de datos compatibles son Avro, JSON, ORC, Parquet y archivos de texto.

Las instrucciones `INSERT INTO` también pueden ayudarle a simplificar el proceso de ETL. Por ejemplo, puede usar `INSERT INTO` en una sola consulta para seleccionar datos a partir de una tabla de origen en formato JSON y escribir en una tabla de destino con formato Parquet.

Las instrucciones `INSERT INTO` se cargan en función del número de bytes que se analizan en la fase `SELECT`, de forma similar a cómo Athena cobra las consultas `SELECT`. Para obtener más información, consulte [Precios de Amazon Athena](#).

Para obtener más información sobre el uso de `INSERT INTO`, incluidos los formatos admitidos, SerDes y ejemplos, consulte [INSERT INTO](#) en la Guía del usuario de Athena.

## 12 de septiembre de 2019

Publicado el 17/12/2019

Amazon Athena ahora está disponible en la región Asia-Pacífico (Hong Kong).

## 16 de agosto de 2019

Publicado el 17/12/2019

[Amazon Athena](#) agrega compatibilidad para la consulta de datos en los buckets de pago por solicitante de Amazon S3.

Cuando un bucket de Amazon S3 se configura como pago por solicitante, el solicitante, no el propietario del bucket, paga los costos de solicitud y transferencia de datos de Amazon S3. En Athena, ahora los administradores de grupos de trabajo pueden ajustar la configuración del grupo de trabajo para permitir que los miembros del grupo de trabajo realicen consultas a los buckets de pagos por solicitante de S3.

Para obtener información acerca de cómo configurar la opción de pago por solicitante para el grupo de trabajo, consulte [Crear un grupo de trabajo](#) en la Guía del usuario de Amazon Athena. Para obtener más información sobre los buckets de pago por solicitante, consulte [Buckets de pago por solicitante](#) en la Guía para desarrolladores de Amazon Simple Storage Service.

## 9 de agosto de 2019

Publicado el 17/12/2019

Amazon Athena admite ahora la aplicación de las políticas de [AWS Lake Formation](#) para un control de acceso detallado a bases de datos, tablas y columnas nuevas o existentes definidas en [AWS Glue Data Catalog](#) para los datos almacenados en Amazon S3.

Puede utilizar esta característica en las siguientes Regiones de AWS: Este de EE. UU. (Ohio), Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón), Asia-Pacífico (Tokio) y Europa (Irlanda). El uso de esta característica no implica costos adicionales.

Para obtener más información sobre cómo usar esta característica, consulte [Uso de Athena para consultar datos registrados en AWS Lake Formation](#). Para obtener más información acerca de AWS Lake Formation, consulte [AWS Lake Formation](#).

## 26 de junio de 2019

Amazon Athena ahora está disponible en la región Europa (Estocolmo). Para obtener una lista de las regiones compatibles, consulte [Puntos de conexión y Regiones de AWS](#).

## 24 de mayo de 2019

Publicado el 24/05/2019

Amazon Athena ahora está disponible en las regiones AWS GovCloud (Este de EE. UU.) y AWS GovCloud (Oeste de EE. UU.). Para obtener una lista de las regiones compatibles, consulte [Puntos de conexión y Regiones de AWS](#).

## 05 de marzo de 2019

Publicado el 05/03/2019

Amazon Athena ahora está disponible en la región Canadá (Central). Para obtener una lista de las regiones compatibles, consulte [Puntos de conexión y Regiones de AWS](#). Se lanzó la nueva versión

del controlador ODBC con compatibilidad para grupos de trabajo de Athena. Para obtener más información, consulte [Notas de la versión del controlador ODBC](#).

Para descargar el nuevo controlador ODBC versión 1.0.5 y su documentación, consulte [Conexión a Amazon Athena con ODBC](#). Para obtener información acerca de esta versión, consulte el tema [Notas de la versión del controlador ODBC](#).

Para utilizar grupos de trabajo con el controlador ODBC, establezca la propiedad de la conexión nueva Workgroup, en la cadena de conexión tal y como se muestra en el siguiente ejemplo:

```
Driver=Simba Athena ODBC
Driver;AwsRegion=[Region];S3OutputLocation=[S3Path];AuthenticationType=IAM
Credentials;UID=[YourAccessKey];PWD=[YourSecretKey];Workgroup=[WorkgroupName]
```

Para obtener más información, busque “grupo de trabajo” en la [Guía de instalación y configuración de la versión 1.0.5 del controlador ODBC](#). No se hacen cambios en la cadena de conexión del controlador ODBC cuando utiliza etiquetas en grupos de trabajo. Para utilizar las etiquetas, actualice a la versión más reciente del controlador ODBC, que es esta versión actual.

Esta versión del controlador le permite utilizar las [acciones de grupo de trabajo de la API de Athena](#) para crear y administrar grupos de trabajo, y las [acciones de etiquetas de la API de Athena](#) para agregar, enumerar o retirar etiquetas de grupos de trabajo. Antes de comenzar, asegúrese de que dispone de permisos de nivel de recursos en IAM para acciones en grupos de trabajo y etiquetas.

Para obtener más información, consulte:

- [Uso de grupos de trabajo para la ejecución de consultas](#) y [Ejemplos de políticas de grupos de trabajo](#).
- [Etiquetado de recursos de Athena](#) y [Políticas de control de acceso de IAM basado en etiquetas](#).

Si utiliza el controlador JDBC o el SDK de AWS, actualice a la versión más reciente del controlador y el SDK, que incluyen compatibilidad para grupos de trabajo y etiquetas en Athena. Para obtener más información, consulte [Conexión a Amazon Athena con JDBC](#).

## 22 de febrero de 2019

Publicado el 22/02/2019

Se agregó compatibilidad con la etiqueta para grupos de trabajo en Amazon Athena. Una etiqueta consta de una clave y un valor, ambos definidos por el usuario. Al etiquetar un grupo de trabajo,



puede asignarle metadatos personalizados. Puede agregar etiquetas a grupos de trabajo para ayudar a categorizarlos siguiendo las [prácticas recomendadas para etiquetado](#) de AWS. Puede usar etiquetas para restringir el acceso a los grupos de trabajo y para realizar un seguimiento de los costos. Por ejemplo, cree un grupo de trabajo para cada centro de costos. A continuación, mediante la adición de etiquetas a estos grupos de trabajo, puede hacer un seguimiento del gasto de Athena para cada centro de costos. Para obtener más información, consulte [Uso de etiquetas para facturación](#) en la Guía del usuario de AWS Billing and Cost Management.

Puede trabajar con etiquetas mediante la consola de Athena o las operaciones de la API. Para obtener más información, consulte [Etiquetado de recursos de Athena](#).

En la consola de Athena, puede agregar una o más etiquetas a cada uno de los grupos de trabajo y buscar por etiquetas. Los grupos de trabajo son un recurso controlado por IAM en Athena. En IAM, puede restringir quién puede agregar, eliminar o enumerar etiquetas en los grupos de trabajo que cree. También puede utilizar la operación de la API `CreateWorkGroup` que tiene el parámetro de etiqueta opcional para añadir una o varias etiquetas al grupo de trabajo. Para añadir, eliminar o enumerar etiquetas, utilice `TagResource`, `UntagResource` y `ListTagsForResource`. Para obtener más información, consulte [Uso de operaciones de etiquetas](#).

Para permitir que los usuarios agreguen etiquetas al crear grupos de trabajo, asegúrese de que proporciona a cada usuario permisos de IAM a las acciones de la API `TagResource` y `CreateWorkGroup`. Para obtener más información y ejemplos, consulte [Políticas de control de acceso de IAM basado en etiquetas](#).

No se hacen cambios en el controlador JDBC cuando se utilizan etiquetas en grupos de trabajo. Si crea nuevos grupos de trabajo y utiliza el controlador JDBC o el SDK de AWS, deberá obtener la versión más reciente del controlador y del SDK. Para obtener más información, consulte [Conexión a Amazon Athena con JDBC](#).

## 18 de febrero de 2019

Publicado el 18/02/2019

Se añadió la capacidad de controlar los costos de las consultas mediante la ejecución de consultas en grupos de trabajo. Para obtener más información, consulte [Uso de grupos de trabajo para controlar el acceso a las consultas y los costos](#). Se ha mejorado el JSON OpenX SerDe utilizado en Athena, se ha corregido un problema que hacía que Athena no ignorara los objetos que hacían la transición a la clase de almacenamiento GLACIER y se han agregado ejemplos para realizar consultas a registros del Network Load Balancer.

Se realizaron los siguientes cambios:

- Se añadió compatibilidad para grupos de trabajo. Utilice grupos de trabajo para separar usuarios, equipos, aplicaciones o cargas de trabajo y establecer límites en la cantidad de datos que puede procesar cada consulta o todo el grupo de trabajo. Como los grupos de trabajo funcionan como recursos de IAM, puede utilizar permisos de nivel de recursos para controlar el acceso a un grupo de trabajo específico. También puede ver métricas relacionadas con las consultas en Amazon CloudWatch, controlar los costos de las consultas mediante la configuración de los límites de la cantidad de datos escaneados, crear los umbrales y desencadenar acciones, como alarmas de Amazon SNS, cuando se superan estos umbrales. Para obtener más información, consulte [Uso de grupos de trabajo para la ejecución de consultas](#) y [Control de costos y supervisión de consultas con métricas y eventos de CloudWatch](#).

Los grupos de trabajo son un recurso de IAM. Para obtener una lista completa de acciones relacionadas con grupos de trabajo, recursos y condiciones en IAM, consulte [Acciones, recursos y claves de condición de Amazon Athena](#) en la Referencia de autorizaciones de servicio. Antes de crear grupos de trabajo nuevos, asegúrese de que utiliza [políticas de IAM para grupos de trabajo](#) y la [Política administrada de AWS: AmazonAthenaFullAccess](#).

Puede comenzar a usar grupos de trabajo en la consola, con las [operaciones de la API de grupo de trabajo](#) o con el controlador JDBC. Para un procedimiento de alto nivel, consulte [Configuración de los grupos de trabajo](#). Para descargar el controlador JDBC con compatibilidad para grupo de trabajo, consulte [Conexión a Amazon Athena con JDBC](#).

Si utiliza grupos de trabajo con el controlador JDBC, debe establecer el nombre de grupo de trabajo en la cadena de conexión mediante el parámetro de configuración `Workgroup` tal y como se muestra en el siguiente ejemplo:

```
jdbc:awsathena://AwsRegion=<AWSREGION>;UID=<ACCESSKEY>;
PWD=<SECRETKEY>;S3OutputLocation=s3://DOC-EXAMPLE-BUCKET/<athena-
output>-<AWSREGION>;
Workgroup=<WORKGROUPNAME>;
```

No hay cambios en la forma de ejecutar instrucciones SQL o realizar llamadas a la API de JDBC para el controlador. El controlador pasa el nombre del grupo de trabajo a Athena.

Para obtener más información acerca de las diferencias introducidas con grupos de trabajo, consulte [API de grupos de trabajo de Athena](#) y [Solución de problemas de grupos de trabajo](#).

- Se mejoró el SerDe JSON de OpenX utilizado en Athena. Las mejoras incluyen, entre otras, lo siguiente:
  - Compatibilidad para la propiedad `ConvertDotsInJsonKeysToUnderscores`. Cuando se establece `TRUE`, permite que SerDe sustituya los puntos en los nombres de claves con guiones bajos. Por ejemplo, si el conjunto de datos de JSON contiene una clave con el nombre "a.b", puede utilizar esta propiedad para definir el nombre de la columna para que sea "a\_b" en Athena. El valor predeterminado es `FALSE`. De forma predeterminada, Athena no permite puntos en los nombres de columnas.
  - Compatibilidad para la propiedad `case.insensitive`. De forma predeterminada, Athena exige que todas las claves de su conjunto de datos JSON utilicen minúscula. El uso de `WITH SERDE PROPERTIES ("case.insensitive"= FALSE;)` le permite usar nombres de clave que distinguen entre mayúsculas y minúsculas en sus datos. El valor predeterminado es `TRUE`. Cuando se establece en `TRUE`, el SerDe convierte todas las columnas en mayúscula a minúscula.

Para obtener más información, consulte [El SerDe JSON de OpenX](#).

- Se corrigió un problema por el que Athena devolvía mensajes de error "access denied" al procesar objetos de Amazon S3 archivados en Glacier por políticas de ciclo de vida de Amazon S3. Al solucionar este problema, Athena ignora objetos que han hecho la transición a la clase de almacenamiento GLACIER. Athena no es compatible con la consulta de datos de la clase de almacenamiento GLACIER.

Para obtener más información, consulte [the section called "Requisitos para las tablas en Athena y los datos en Amazon S3"](#) y [Transición a la clase de almacenamiento GLACIER \(archivo de objetos\)](#) en la Guía del usuario de Amazon Simple Storage Service.

- Se agregaron ejemplos para consultar los registros de acceso del Equilibrador de carga de red que reciben información acerca de las solicitudes de seguridad de la capa de transporte (TLS). Para obtener más información, consulte [the section called "Network Load Balancer"](#).

## Notas de la versión de Athena para 2018

20 de noviembre de 2018

Publicado el 20/11/2018

Se lanzaron las nuevas versiones del controlador JDBC y ODBC con compatibilidad para acceso federado a la API de Athena con Active Directory Federation Services (AD FS) y SAML 2.0 (Lenguaje de marcado para confirmaciones de seguridad 2.0). Para obtener más información, consulte las [Notas de la versión del controlador JDBC](#) y las [Notas de la versión del controlador ODBC](#).

Con esta versión, el acceso federado a Athena es compatible con Active Directory Federation Service (AD FS 3.0). El acceso se establece a través de las versiones de los controladores JDBC u ODBC que admiten SAML 2.0. Para obtener información acerca de la configuración del acceso federado a la API de Athena, consulte [the section called “Habilitación de acceso federado a la API de Athena”](#).

Para descargar el nuevo controlador JDBC versión 2.0.6 y su documentación, consulte [Conexión a Amazon Athena con JDBC](#). Para obtener información acerca de esta versión, consulte el tema [Notas de la versión del controlador JDBC](#).

Para descargar el nuevo controlador ODBC versión 1.0.4 y su documentación, consulte [Conexión a Amazon Athena con ODBC](#). Para obtener información acerca de esta versión, consulte el tema [Notas de la versión del controlador ODBC](#).

Para obtener más información acerca de la compatibilidad con SAML 2.0 en AWS, consulte [Acerca de la federación SAML 2.0](#) en la Guía del usuario de IAM.

## 15 de octubre de 2018

Publicado el 15/10/2018

Si ha actualizado a AWS Glue Data Catalog, hay dos nuevas características que proporcionan compatibilidad para:

- Cifrado de los metadatos del catálogo de datos. Si decide cifrar los metadatos del catálogo de datos, debe agregar las políticas específicas a Athena. Para obtener más información, consulte [Acceso a metadatos cifrados del AWS Glue Data Catalog](#).
- Permisos detallados para el acceso a recursos de AWS Glue Data Catalog Ahora puede definir políticas basadas en identidad (IAM) que restrinjan o permitan el acceso a bases de datos y tablas específicas del catálogo de datos usado en Athena. Para obtener más información, consulte [Acceso detallado a las bases de datos y las tablas en AWS Glue Data Catalog](#).

**Note**

Los datos residen en los buckets de Amazon S3 y el acceso a ellos se rige por el [Acceso a Amazon S3](#). Para obtener acceso al contenido de las bases de datos y las tablas, siga utilizando las políticas de control de acceso a los buckets de Amazon S3 donde se almacenan.

## 10 de octubre de 2018

Publicado el 10/10/2018

Athena admite `CREATE TABLE AS SELECT`, que crea una tabla a partir del resultado de una instrucción de consulta `SELECT`. Para obtener información detallada, consulte [Creación de una tabla a partir de los resultados de una consulta \(CTAS\)](#).

Antes de crear consultas CTAS, es importante que conozca su comportamiento, descrito en la documentación de Athena. Esta documentación contiene información acerca de la ubicación para guardar los resultados de las consultas en Amazon S3, la lista de los formatos admitidos para almacenar los resultados de las consultas CTAS, el número de particiones que puede crear y los formatos de compresión admitidos. Para obtener más información, consulte [Consideraciones y limitaciones de las consultas CTAS](#).

Utilice consultas CTAS para:

- [Crear una tabla a partir de los resultados de una consulta](#) en un solo paso.
- [Crear consultas CTAS en la consola de Athena](#), utilizando los [ejemplos](#). Para obtener información sobre la sintaxis, consulte [CREATE TABLE AS](#).
- Transformar los resultados de las consultas en otros formatos de almacenamiento, como PARQUET, ORC, AVRO, JSON y TEXTFILE. Para obtener más información, consulte [Consideraciones y limitaciones de las consultas CTAS](#) y [Formatos de almacenamiento en columnas](#).

## 6 de septiembre de 2018

Publicado el 06/09/2018

Se ha publicado la nueva versión del controlador ODBC (versión 1.0.3). La nueva versión del controlador ODBC transmite los resultados de forma predeterminada, en lugar de paginarlos, lo que permite a las herramientas de inteligencia empresarial obtener más rápidamente grandes conjuntos de datos. Esta versión incluye también mejoras, correcciones de errores y una documentación actualizada para “Uso de SSL con un servidor proxy”. Para obtener más información, consulte las [notas de la versión](#) del controlador.

Para descargar el nuevo controlador ODBC versión 1.0.3 y su documentación, consulte [Conexión a Amazon Athena con ODBC](#).

La característica de transmisión de resultados está disponible con esta nueva versión del controlador ODBC. También está disponible con el controlador JDBC. Para obtener más información sobre los resultados de transmisión, consulte la [Guía de instalación y configuración del controlador ODBC](#) y busque UseResultsetStreaming.

La versión 1.0.3 del controlador ODBC sustituye directamente a la versión anterior. Se recomienda migrar al controlador actual.

#### Important

Para utilizar la versión del controlador ODBC 1.0.3, siga estos requisitos:

- Mantenga el puerto 444 abierto para el tráfico de salida.
- Agregue la acción de política `athena:GetQueryResultsStream` a la lista de políticas de Athena. Esta acción de política no se expone directamente con la API y solo se utiliza con los controladores ODBC y JDBC como parte de la funcionalidad de resultados en transmisión. Para ver una política de ejemplo, consulte [Política administrada de AWS: AWSQuicksightAthenaAccess](#).

## 23 de agosto de 2018

Publicado el 23/08/2018

Se agregó compatibilidad para características de DDL y se corrigieron algunos errores, como se indica:

- Se agregó compatibilidad para los tipos de datos BINARY y DATE de Parquet y para los tipos de datos DATE y TIMESTAMP de Avro.

- Se agregó compatibilidad para INT y DOUBLE en las consultas DDL. INTEGER es un alias de INT y DOUBLE PRECISION es un alias de DOUBLE.
- Mejora del rendimiento de las consultas DROP TABLE y DROP DATABASE.
- Se eliminó la creación de un objeto `_$folder$` en Amazon S3 cuando un bucket de datos está vacío.
- Corregido un problema por el que ALTER TABLE ADD PARTITION generaba un error cuando no se indicaba un valor de partición.
- Corregido un problema por el que DROP TABLE no tenía en cuenta el nombre de la base de datos al comprobar las particiones después de haberse especificado el nombre completo en la instrucción.

Para obtener más información sobre los tipos de datos admitidos en Athena, consulte [Tipos de datos en Amazon Athena](#).

Para obtener más información acerca de la asignación entre los tipos de datos admitidos en Athena, el controlador JDBC y los tipos de datos de Java, consulte la sección “Tipos de datos” en la [Guía de instalación y configuración del controlador JDBC](#).

## 16 de agosto de 2018

Publicado el 16/08/2018

Se ha publicado el controlador JDBC versión 2.0.5. La nueva versión del controlador JDBC transmite los resultados de forma predeterminada, en lugar de paginarlos, lo que permite a las herramientas de inteligencia empresarial obtener más rápidamente grandes conjuntos de datos. En comparación con la versión anterior del controlador JDBC, se introducen las siguientes mejoras en el rendimiento:

- Aproximadamente el doble de rendimiento al recuperar menos de 10 000 filas.
- Aproximadamente 5 o 6 veces más rendimiento al recuperar más de 10 000 filas.

La característica de transmisión de resultados solo está disponible con el controlador JDBC. No está disponible con el controlador ODBC. No puede utilizarla con la API de Athena. Para obtener más información sobre la transmisión de resultados, consulte la [guía de instalación y configuración del controlador JDBC](#) y busque `UseResultsetStreaming`.

Para descargar el nuevo controlador JDBC versión 2.0.5 y su documentación, consulte [Conexión a Amazon Athena con JDBC](#).

La versión 2.0.5 del controlador JDBC sustituye directamente a la versión anterior (2.0.2). Para asegurarse de poder utilizar la versión 2.0.5 del controlador JDBC, agregue la acción de política `athena:GetQueryResultsStream` a la lista de políticas para Athena. Esta acción de política no se expone directamente con la API y solo se utiliza con el controlador JDBC como parte de la funcionalidad de resultados de transmisión. Para ver una política de ejemplo, consulte [Política administrada de AWS: AWSQuicksightAthenaAccess](#). Para obtener más información acerca de cómo migrar de la versión 2.0.2 a la versión 2.0.5 del controlador, consulte la [Guía de migración del controlador JDBC](#).

Si va a migrar desde un controlador 1.x a un controlador 2.x, tendrá que migrar las configuraciones existentes a la nueva configuración. Recomendamos encarecidamente que migre a la versión actual del controlador. Para obtener más información, consulte la [Guía de migración del controlador JDBC](#).

## 7 de agosto de 2018

Publicado el 07/08/2018

Ahora puede almacenar directamente registros de flujo de Amazon Virtual Private Cloud en Amazon S3 con formato GZIP, lo que le permite consultarlos en Athena. Para obtener más información, consulte [Consulta de los registros de flujo de Amazon VPC](#) y [Ahora es posible entregar en S3 los registros de flujo de Amazon VPC](#).

## 5 de junio de 2018

Publicado el 05/06/2018

Temas

- [Compatibilidad con vistas](#)
- [Mejoras y actualizaciones de los mensajes de error](#)
- [Correcciones de errores](#)

### Compatibilidad con vistas

Se ha agregado compatibilidad con vistas. A partir de ahora, puede utilizar [CREATE VIEW](#), [DESCRIBE VIEW](#), [DROP VIEW](#), [SHOW CREATE VIEW](#) y [SHOW VIEWS](#) en Athena. La consulta que define la vista se ejecuta cada vez que se hace referencia a su vista en su consulta. Para obtener más información, consulte [Uso de vistas](#).



## Mejoras y actualizaciones de los mensajes de error

- Incluye una biblioteca GSON 2.8.0 en el CloudTrail SerDe para resolver un problema con el CloudTrail SerDe y habilitar el análisis de cadenas JSON.
- Mejora de la validación de esquemas de particiones en Athena para Parquet y, en algunos casos, para ORC, al permitir reordenar las columnas. De este modo, Athena puede afrontar mejor los cambios en la evolución de los esquemas y las tablas agregadas por el rastreador de AWS Glue. Para obtener más información, consulte [Gestión de las actualizaciones de los esquemas](#).
- Se añadió compatibilidad de análisis para SHOW VIEWS.
- Se realizaron las siguientes mejoras en los mensajes de error más comunes:
  - Se reemplazó un mensaje de Error interno por un mensaje de error descriptivo cuando un SerDe no logra analizar la columna de una consulta de Athena. Anteriormente, Athena generaba un error interno cuando se producían errores de análisis. El nuevo mensaje de error es el siguiente: `HIVE_BAD_DATA: Error parsing field value for field 0: java.lang.String cannot be cast to org.openx.data.jsonserde.json.JSONObject`.
  - Se mejoraron los mensajes de error acerca de la falta de permisos añadiendo más detalles.

## Correcciones de errores

Se corrigieron los siguientes errores:

- Se solucionó un problema que permite la traducción interna de tipos de datos REAL a FLOAT. Esto mejora la integración con el rastreador de AWS Glue, que devuelve tipos de datos FLOAT.
- Se corrigió un problema que hacía que Athena no convirtiera DECIMAL de AVRO (un tipo lógico) a un tipo DECIMAL.
- Se corrigió un problema que hacía que Athena no devolviera resultados para las consultas de datos Parquet con cláusulas WHERE que hacían referencia a valores con el tipo de datos TIMESTAMP.

## 17 de mayo de 2018

Publicado el 17/05/2018

Se incrementó la cuota de consultas simultáneas en Athena de cinco a veinte. Esto significa que puede enviar y ejecutar hasta veinte consultas DDL y veinte consultas SELECT a la vez. Tenga en cuenta que las cuotas de simultaneidad son independientes para las consultas DDL y SELECT.

Las cuotas de simultaneidad en Athena se definen como el número de consultas que se puede enviar al servicio de forma simultánea. Puede enviar hasta veinte consultas del mismo tipo (DDL o SELECT) de forma simultánea. Si envía una consulta que supera la cuota de consultas simultáneas, la API de Athena muestra un mensaje de error.

Una vez enviadas las consultas a Athena, este las procesa y les asigna recursos en función de la carga de servicio general y la cantidad de solicitudes entrantes. Monitorizamos de forma continua y realizamos ajustes del servicio para que sus consultas se procesen con la mayor rapidez posible.

Para obtener más información, consulte [Service Quotas](#). Esta es una cuota ajustable. Puede utilizar la [consola Service Quotas](#) para solicitar un aumento de cuota para consultas simultáneas.

## 19 de abril de 2018

Publicado el 19/04/2018

Se lanzó la nueva versión del controlador JDBC (versión 2.0.2) que permite devolver los datos de `ResultSet` como un tipo de datos `Array`, e incluye mejoras y correcciones de errores. Para obtener más información, consulte las [notas de la versión](#) del controlador.

Para obtener información acerca de la descarga del nuevo controlador JDBC versión 2.0.2 y tener acceso a su documentación, consulte [Conexión a Amazon Athena con JDBC](#).

La última versión del controlador JDBC es la 2.0.2. Si va a migrar desde un controlador 1.x a un controlador 2.x, tendrá que migrar las configuraciones existentes a la nueva configuración. Recomendamos encarecidamente que migre al controlador actual.

Para obtener información sobre los cambios introducidos en la nueva versión del controlador, conocer las diferencias entre las distintas versiones y ver ejemplos, consulte el documento [Guía de migración del controlador JDBC](#).

## 6 de abril de 2018

Publicado el 06/04/2018

Utilice la función de autocompletar para introducir las consultas en la consola de Athena.

## 15 de marzo de 2018

Publicado el 15/03/2018

Se agregó la capacidad de crear tablas de Athena de manera automática para archivos de registro de CloudTrail directamente desde la consola de CloudTrail. Para obtener más información, consulte [Uso de la consola de CloudTrail para crear una tabla de Athena para registros de CloudTrail](#).

## 2 de febrero de 2018

Publicado el 12/02/2018

Se ha añadido la capacidad de descargar de forma segura datos intermedios en disco para consultas con uso intensivo de memoria que utilicen la cláusula `GROUP BY`. Esto mejora la fiabilidad de dichas consultas, para evitar los errores “Recurso de consulta agotado”.

## 19 de enero de 2018

Publicado el 19/01/2018

Athena utiliza Presto, un motor de consultas distribuido de código abierto, para ejecutar consultas.

Con Athena, no hay que administrar versiones. Hemos actualizado de forma transparente el motor subyacente en Athena a una versión basada en la versión 0.172 de Presto. No tiene que hacer nada.

Con la actualización, ahora puede usar las funciones y operadores de Presto 0.172, incluidas las expresiones Lambda de Presto 0.172 en Athena.

Las principales actualizaciones de esta versión, incluidas las correcciones a las que ha contribuido la comunidad, incluyen:

- Compatibilidad para no tener en cuenta los encabezados. Puede utilizar la propiedad `skip.header.line.count` al definir tablas para permitir que Athena no tenga en cuenta los encabezados. Se admite para las consultas que utilizan el [LazySimpleSerDe](#) y el [SerDe de OpenCSV](#), pero no para Grok o Regex SerDes.
- Compatibilidad con el tipo de datos `CHAR(n)` en las funciones `STRING`. El intervalo de `CHAR(n)` es `[1,255]`, mientras que el intervalo de `VARCHAR(n)` es `[1,65535]`.
- Compatibilidad con subconsultas correlacionadas.
- Compatibilidad con funciones y expresiones Lambda de Presto.
- Mejor rendimiento de los operadores y el tipo `DECIMAL`.
- Compatibilidad con agregaciones filtradas como `SELECT sum(col_name) FILTER, donde id > 0`.
- Predicados de inserción abajo para los tipos de datos `DECIMAL`, `TINYINT`, `SMALLINT` y `REAL`.

- Compatibilidad con los predicados de comparación de cuantificación: ALL, ANY y SOME.
- Se añadieron las funciones: [arrays\\_overlap\(\)](#), [array\\_except\(\)](#), [levenshtein\\_distance\(\)](#), [codepoint\(\)](#), [skewness\(\)](#), [kurtosis\(\)](#) y [typeof\(\)](#).
- Se añadió una variante de la función [from\\_unixtime\(\)](#) que toma un argumento de zona horaria.
- Se añadieron las funciones de agregación [bitwise\\_and\\_agg\(\)](#) y [bitwise\\_or\\_agg\(\)](#).
- Se añadieron las funciones [xxhash64\(\)](#) y [to\\_big\\_endian\\_64\(\)](#).
- Se añadió compatibilidad con comillas o barras diagonales inversas de escape utilizando una barra diagonal inversa con un subíndice de ruta JSON a las funciones [json\\_extract\(\)](#) y [json\\_extract\\_scalar\(\)](#). Esto cambia la semántica de cualquier invocación que utilice una barra diagonal inversa, ya que anteriormente dichas barras se trataban como caracteres normales.

Para obtener una lista completa de las funciones y los operadores, consulte [Consultas, funciones y operadores de DML](#) en esta guía y [Funciones y operadores](#) en la documentación de Presto.

Athena no es compatible con todas las características de Presto. Para obtener más información, consulte la sección sobre [Límites](#).

## Notas de la versión de Athena para 2017

### 13 de noviembre de 2017

Publicado el 13/11/2017

Se añadió compatibilidad para conectar Athena al controlador ODBC. Para obtener más información, consulte [Conexión a Amazon Athena con ODBC](#).

### 1 de noviembre de 2017

Publicado el 01/11/2017

Se añadió compatibilidad con consultas de datos geoespaciales y con las regiones de Asia-Pacífico (Seúl), Asia-Pacífico (Bombay) y UE (Londres). Para obtener más información, consulte [Consulta de datos geoespaciales](#) y [Puntos de conexión y Regiones de AWS](#).

### 19 de octubre de 2017

Publicado el 19/10/2017

Se añadió compatibilidad con la región UE (Fráncfort). Para obtener una lista de las regiones compatibles, consulte [Puntos de conexión y Regiones de AWS](#).

## 3 de octubre de 2017

Publicado el 03/10/2017

Se han creado consultas de Athena con nombre por medio de AWS CloudFormation. Para obtener más información, consulte [AmazonWebService::Athena::NamedQuery](#) en la Guía del usuario de AWS CloudFormation.

## 25 de septiembre de 2017

Publicado el 25/09/2017

Se añadió compatibilidad con la región de Asia-Pacífico (Sídney). Para obtener una lista de las regiones compatibles, consulte [Puntos de conexión y Regiones de AWS](#).

## 14 de agosto de 2017

Publicado el 14/08/2017

Se agregó integración con AWS Glue Data Catalog y un asistente de migración para actualizar del catálogo de datos administrado por Athena a AWS Glue Data Catalog. Para obtener más información, consulte [Integración con AWS Glue](#).

## 4 de agosto de 2017

Publicado el 04/08/2017

Se añadió compatibilidad con el SerDe de Grok, que proporciona una asociación de patrones más sencilla de los registros que se encuentran en archivos de texto no estructurados como registros. Para obtener más información, consulte [El SerDe de Grok](#). Se añadieron métodos abreviados de teclado para desplazarse por el historial de consultas utilizando la consola (CTRL+↑/↓ con Windows, CMD+↑/↓ con Mac).

## 22 de junio de 2017

Publicado el 22/06/2017

Se añadió compatibilidad con las regiones de Asia-Pacífico (Tokio) y Asia-Pacífico (Singapur). Para obtener una lista de las regiones compatibles, consulte [Puntos de conexión y Regiones de AWS](#).

## 8 de junio de 2017

Publicado el 08/06/2017

Se agregó compatibilidad con la región de Europa (Irlanda). Para obtener más información, consulte [Puntos de conexión y Regiones de AWS](#).

## 19 de mayo de 2017

Publicado el 19/05/2017

Se agregó una API de Amazon Athena y compatibilidad con la AWS CLI para Athena; se actualizó el controlador JDBC a la versión 1.1.0; se solucionaron varios problemas.

- Amazon Athena permite la programación de aplicaciones para Athena. Para obtener más información, consulte la sección de [referencia de API de Amazon Athena](#). Los últimos SDK de AWS incluyen compatibilidad para la API de Athena. Para obtener enlaces con la documentación y las descargas, consulte la sección SDK de [Herramientas para Amazon Web Services](#).
- La AWS CLI contiene nuevos comandos para Athena. Para obtener más información, consulte la sección de [referencia de API de Amazon Athena](#).
- Un nuevo controlador JDBC 1.1.0 está disponible; es compatible con la nueva API de Athena, así como con las últimas características y correcciones de errores. Descargue el controlador en <https://downloads.athena.us-east-1.amazonaws.com/drivers/AthenaJDBC41-1.1.0.jar>. Le recomendamos que realice la actualización al último controlador JDBC de Athena; no obstante, puede seguir utilizando la versión anterior. Las versiones anteriores del controlador no son compatibles con la API de Athena. Para obtener más información, consulte [Conexión a Amazon Athena con JDBC](#).
- Las acciones específicas de instrucciones de políticas de las versiones anteriores de Athena se dejaron de utilizar. Si actualiza a la versión 1.1.0 del controlador JDBC y tiene políticas de IAM administradas por el cliente o insertadas asociadas a usuarios de JDBC, debe actualizar las políticas de IAM. En cambio, las versiones anteriores del controlador JDBC no son compatibles con la API de Athena, por lo que puede especificar únicamente acciones obsoletas en las políticas asociadas a usuarios de la versión anterior de JDBC. Por este motivo, no debería necesitar actualizar las políticas de IAM insertadas o administradas por el cliente.
- Estas acciones específicas de la política se utilizaban en Athena antes del lanzamiento de la API de Athena. Estas acciones obsoletas deben usarse únicamente en políticas con versiones del controlador JDBC anteriores a la 1.1.0. Si actualiza el controlador JDBC, sustituya

las instrucciones de política que permiten o deniegan acciones obsoletas por las acciones correspondientes de la API como se indica, o se producirán errores:

#### Acción específica de política obsoleta

`athena:RunQuery`

`athena:CancelQueryExecution`

`athena:GetQueryExecutions`

#### Acción de la API de Athena correspondiente

`athena:StartQueryExecution`

`athena:StopQueryExecution`

`athena:ListQueryExecutions`

## Mejoras

- Se aumentó la longitud de la cadena de consulta a 256 KB.

## Correcciones de errores

- Se corrigió un error que hacía que los resultados de las consultas parecieran tener un formato erróneo cuando el usuario se desplazaba por los resultados en la consola.
- Se corrigió un error en el que una cadena de caracteres `\u0000` de archivos de datos de Amazon S3 provocaba errores.
- Se corrigió un error que hacía que las solicitudes de cancelación de una consulta realizada a través del controlador JDBC generaran un error.
- Se corrigió un error que hacía que el SerDe de AWS CloudTrail generara un error con datos de Amazon S3 en la región Este de EE. UU. (Ohio).
- Se corrigió un error que hacía que `DROP TABLE` generara un error en tablas con particiones.

## 4 de abril de 2017

Publicado el 04/04/2017

Se ha añadido compatibilidad con el cifrado de datos de Amazon S3 y se ha publicado una actualización del controlador JDBC (versión 1.0.1) con compatibilidad con el cifrado, mejoras y correcciones de errores.

## Características

- Se añadieron las siguientes características de cifrado:
  - Compatibilidad con la consulta de datos cifrados en Amazon S3.
  - Compatibilidad con el cifrado de los resultados de consultas de Athena.
- Una nueva versión del controlador es compatible con las nuevas características de cifrado, añade mejoras y corrige problemas.
- Se añadió la capacidad para añadir, reemplazar y cambiar columnas utilizando ALTER TABLE. Para obtener más información, consulte la sección sobre [cómo alterar columnas](#) en la documentación de Hive.
- Se añadió compatibilidad para consultar datos comprimidos mediante LZ0.

Para obtener más información, consulte [Cifrado en reposo](#).

## Mejoras

- Mejor rendimiento de las consultas de JDBC con mejoras de tamaño de página, se devuelven 1000 filas en lugar de 100.
- Se añadió la capacidad de cancelar una consulta mediante la interfaz del controlador JDBC.
- Se añadió la capacidad de especificar opciones de JDBC en la URL de conexión de JDBC. Consulte [Conexión a Amazon Athena con JDBC](#) para obtener el controlador JDBC más reciente.
- Se agregó la configuración PROXY al controlador, que ahora se puede establecer utilizando [ClientConfiguration](#) en el AWS SDK para Java.

## Correcciones de errores

Se corrigieron los siguientes errores:

- Se producían errores de limitación controlada cuando se emitían varias consultas con la interfaz del controlador JDBC.
- El controlador JDBC se detenía al proyectar un tipo de datos decimal.



- El controlador JDBC devolvía todos los tipos de datos como una cadena, sin tener en cuenta cómo se había definido el tipo de datos en la tabla. Por ejemplo, si seleccionaba una columna definida como tipo de datos INT mediante `resultSet.getObject()`, se devolvía un tipo de datos STRING en lugar de INT.
- El controlador JDBC verificaba las credenciales en el momento en que se realizaba la conexión, en lugar de hacerlo en el momento de ejecución de una consulta.
- Las consultas realizadas mediante el controlador JDBC generaban un error cuando se especificaba un esquema junto con la dirección URL.

## 24 de marzo de 2017

Publicado el 24/03/2017

Se agregó el SerDe de AWS CloudTrail, se mejoró el rendimiento y se corrigieron problemas de partición.

### Características

- Se agregó el SerDe de AWS CloudTrail, que desde entonces ha sido reemplazado por el [El SerDe JSON de Hive](#) para leer los registros de CloudTrail. Para obtener más información sobre la consulta de registros de CloudTrail, consulte [Consulta de registros de AWS CloudTrail](#).

### Mejoras

- Se mejoró el rendimiento cuando se analiza un gran número de particiones.
- Se mejoró el rendimiento en la operación `MSCK Repair Table`.
- Se añadió la capacidad de consultar datos de Amazon S3 almacenados en regiones distintas de la región principal. Se aplican tasas estándar de transferencia de datos entre regiones para Amazon S3, además de las tasas estándar de Athena.

### Correcciones de errores

- Se corrigió un error del tipo “No se encontró la tabla” que podía producirse si no se cargaban particiones.
- Se corrigió un error para evitar que se produzca una excepción con las consultas `ALTER TABLE ADD PARTITION IF NOT EXISTS`.

- Se corrigió un error en DROP PARTITIONS.

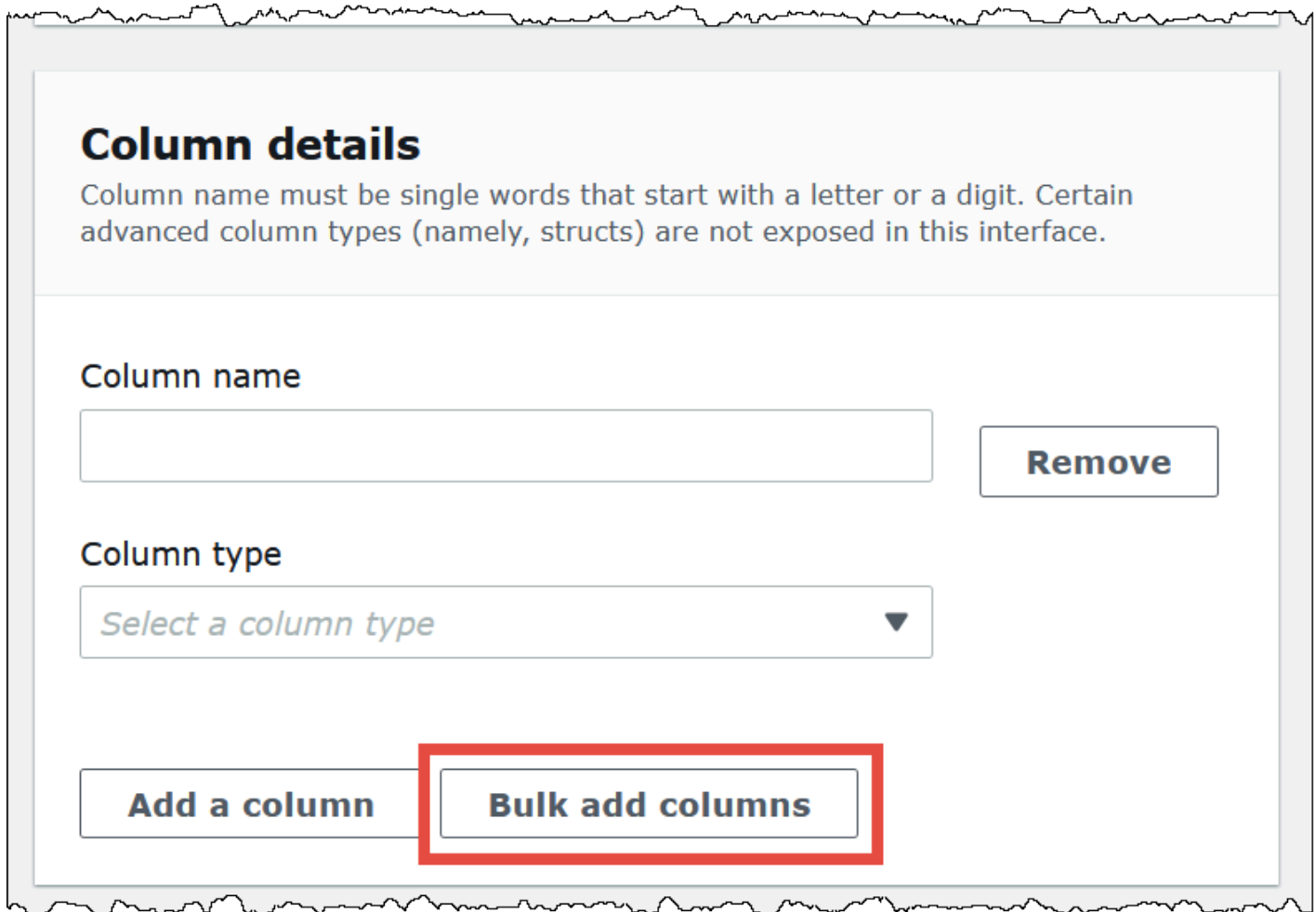
## 20 de febrero de 2017

Publicado el 20/02/2017

Se agregó compatibilidad para AvroSerDe y OpenCSVSerDe, la región Este de EE. UU. (Ohio) y la edición masiva de columnas en el asistente de la consola. Se ha mejorado el rendimiento en las tablas Parquet grandes.

### Características

- Se introdujo compatibilidad con los nuevos SerDes:
  - [El SerDe de Avro](#)
  - [OpenCSVSerDe para el procesamiento de CSV](#)
- Lanzamiento para la región Este de EE. UU. (Ohio) (us-east-2). Ahora ya puede ejecutar consultas en esta región.
- A partir de ahora, puede utilizar el formulario Crear tabla a partir de datos de bucket de S3 para definir esquemas de tablas en lote. En el editor de consultas, elija Crear, Datos del bucket de S3 y luego Agregar columnas por lotes en la sección Detalles de columna.



**Column details**

Column name must be single words that start with a letter or a digit. Certain advanced column types (namely, structs) are not exposed in this interface.

Column name

**Remove**

Column type

*Select a column type* ▼

**Add a column** **Bulk add columns**

Escriba los pares de valor y nombre en el cuadro de texto y elija Añadir.

## Bulk add columns ×

Define columns in name value pairs, using commas to separate definitions (col1\_name data\_type, col2\_name data\_type, ...). Certain advanced data types (namely, structs) are not supported in this interface, but are supported using DDL statements.

```
id int, name string
```

## Mejoras

- Se ha mejorado el rendimiento en las tablas Parquet grandes.

# Historial de documentos

Última actualización de la documentación: 28 de mayo de 2024.

Actualizamos la documentación con frecuencia para responder a sus comentarios. En la siguiente tabla, se describen los cambios importantes de la documentación de Amazon Athena. No aparecen todas las actualizaciones.

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se actualizó la política administrada de Amazon Athena FullAccess .	Se agregaron los permisos <code>datazone:ListDomains</code> , <code>datazone:ListProjects</code> y <code>datazone:ListAccountEnvironments</code> a la política administrada <a href="#">AmazonAthenaFullAccess</a> . Las acciones agregadas permiten a los usuarios de Athena trabajar con dominios, proyectos y entornos de Amazon DataZone. Para obtener más información, consulte <a href="#">Uso de Amazon DataZone en Athena</a> .	3 de enero de 2024
Se actualizó la política administrada de Amazon Athena FullAccess .	Agregados los permisos <code>glue:StartColumnStatisticsTaskRun</code> , <code>glue:GetColumnStatisticsTaskRun</code> y <code>glue:GetColumnStatisticsTaskRuns</code> a la política administrada <a href="#">AmazonAthenaFullAccess</a> . Las acciones agregadas permiten a Athena llamar a AWS Glue para recuperar estadísticas de la característica de optimización basada en costos. Para obtener más información, consulte <a href="#">Uso del optimizador basado en costes</a> .	3 de enero de 2024
Se agregó documentación para los grupos de trabajo de Athena habilitados	Puede crear grupos de trabajo de Athena SQL que utilicen el modo de autenticación de IAM Identity Center. Estos grupos de trabajo admiten el uso de la misma identidad entre servicios de AWS como Amazon Athena y Amazon EMR Studio. Para obtener más información,	5 de diciembre de 2023

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
para IAM Identity Center.	consulte <a href="#">Uso de grupos de trabajo de Athena habilitados para IAM Identity Center</a> .	
Se agregó documentación para consultar los datos de S3 Express One Zone	Puede utilizar Athena para consultar datos en la clase de almacenamiento de Amazon S3 Express One Zone. Para obtener más información, consulte <a href="#">Consulta de datos de S3 Express One Zone</a> .	28 de noviembre de 2023
Se agregó documentación para las vistas del Catálogo de datos de Glue.	Puede utilizar las vistas del Catálogo de datos de Glue para proporcionar una única vista común de los servicios de AWS como Amazon Athena y Amazon Redshift. Para obtener más información, consulte <a href="#">Uso de vistas del AWS Glue Data Catalog</a> .	27 de noviembre de 2023
Se agregó documentación para la característica de optimizador basado en costos.	Puede utilizar las estadísticas de AWS Glue para optimizar sus consultas en Athena SQL. Para obtener más información, consulte <a href="#">Uso del optimizador basado en costes</a> .	17 de noviembre de 2023
Se agregó documentación para el controlador Athena JDBC 3.x	Puede utilizar el controlador Athena JDBC 3.x para leer los resultados de las consultas directamente desde Amazon S3. El controlador JDBC 3.x es compatible con casi todos los métodos de autenticación que admite el controlador JDBC 2.x. Para obtener más información, consulte <a href="#">Controlador JDBC 3.x de Athena</a> .	16 de noviembre de 2023

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se agregó documentación para usar DataZone en Athena.	Puede usar DataZone para simplificar su experiencia en servicios de análisis de AWS como Athena, AWS Glue y Lake Formation. Para obtener más información, consulte <a href="#">Uso de Amazon DataZone en Athena</a> .	4 de octubre de 2023
Se agregó documentación para las reservas de capacidad.	Ahora puede utilizar las reservas de capacidad en Amazon Athena para ejecutar consultas SQL en una capacidad de procesamiento totalmente administrada. Para obtener más información, consulte <a href="#">Administración de la capacidad de procesamiento de consultas</a> .	28 de abril de 2023
Se agregó documentación para las consultas de vistas federadas.	Ahora puede crear y consultar vistas en orígenes de datos federados en Athena. Para obtener más información, consulte <a href="#">Consulta de vistas federadas</a> .	4 de abril de 2023
Se agregó documentación sobre cómo evitar la limitación en Amazon S3.	Para obtener más información, consulte <a href="#">Evitar la limitación de Amazon S3</a> .	24 de marzo de 2023
Se actualizó la política administrada de Amazon AthenaFullAccess.	Se agregó <code>pricing:GetProducts</code> a la política administrada <a href="#">AmazonAthenaFullAccess</a> . La acción agregada proporciona acceso a AWS Billing and Cost Management. Para obtener más información, consulte <a href="#">GetProducts</a> en la Referencia de la API de AWS Billing and Cost Management.	25 de enero de 2023

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Ampliación de la documentación sobre compatibilidad de compresión de Athena.	Se agregaron temas individuales para <a href="#">Compresión de tablas de Hive</a> , <a href="#">Compresión de tablas de Iceberg</a> y <a href="#">Niveles de compresión ZSTD</a> . Para obtener más información, consulte <a href="#">Compatibilidad con la compresión de Athena</a> .	20 de enero de 2023
Se agregó documentación de Amazon Athena para Apache Spark.	Ahora puede crear y ejecutar aplicaciones de Apache Spark y cuadernos compatibles con Jupyter de forma interactiva en Amazon Athena. Para obtener más información, consulte <a href="#">Uso de Apache Spark en Amazon Athena</a> .	30 de noviembre de 2022
Se agregó documentación sobre el conector para IBM Db2 de Athena.	Puede utilizar el conector de Amazon Athena para IBM Db2 a fin de realizar consultas de Db2 desde Athena. Para obtener más información, consulte <a href="#">Conector para IBM Db2 de Amazon Athena</a>	18 de noviembre de 2022
Se agregó documentación sobre la reutilización de los resultados de las consultas.	Al volver a ejecutar una consulta en Athena, ahora puede optar por reutilizar el último resultado almacenado de la consulta. Esto puede aumentar el rendimiento y reducir los costos en cuanto al número de bytes analizados. Para obtener más información, consulte <a href="#">Reutilización de los resultados de las consultas</a> .	8 de noviembre de 2022
Documentación actualizada sobre los registros de CloudTrail.	Se ha actualizado el DDL de CREATE TABLE para consultar los registros de CloudTrail con el fin de usar el SerDe de JSON en lugar del SerDe de CloudTrail. Para obtener más información, consulte <a href="#">Consulta de registros de AWS CloudTrail</a> .	3 de noviembre de 2022



Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se agregó documentación para la versión 3 del motor Athena.	Para obtener información acerca de la versión 3 del motor Athena, consulte <a href="#">Versión 3 del motor Athena</a> .	13 de octubre de 2022
Se agregó un tutorial sobre la configuración de single sign-on (SSO, inicio de sesión único) para ODBC mediante el complemento Okta.	Configure el controlador ODBC de Amazon Athena y el complemento Okta para el módulo de inicio de sesión único (SSO) mediante el proveedor de identidad de Okta. Para obtener más información, consulte <a href="#">Configuración de SSO para ODBC mediante el complemento Okta y el proveedor de identidad Okta</a> .	23 de agosto de 2022
Se agregó documentación para ver planes de consultas y estadísticas en la consola de Athena.	Puede usar el editor de consultas de Athena para ver representaciones gráficas que muestren cómo se ejecutarán las consultas, además de gráficos, detalles y estadísticas de cómo se ejecutaron las consultas completadas. Para obtener más información, consulte <a href="#">Ver planes de ejecución para consultas SQL</a> y <a href="#">Visualización de estadísticas y detalles de ejecución de consultas completadas</a> .	21 de julio de 2022
Se agregó documentación para consultar las vistas de Apache Hive en los almacenes de metadatos externos de Hive.	Puede utilizar Athena para consultar las vistas de Apache creadas en los almacenes de metadatos externos de Hive. Algunas funciones de Hive no son compatibles ni requieren un manejo especial. Para obtener más información, consulte <a href="#">Trabajo con vistas de Hive</a> .	22 de abril de 2022

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se agregó documentación para consultas guardadas.	Puede utilizar la característica de consultas guardadas de Athena para guardar, recuperar, editar y cambiar el nombre de las consultas. Para obtener más información, consulte <a href="#">Uso de consultas guardadas</a> en esta guía y <a href="#">UpdateNamedQuery</a> en la Referencia de API de Amazon Athena.	28 de febrero de 2022
Se agregó documentación de versión preliminar para la compatibilidad con Apache Iceberg.	Athena admite consultas de lectura, viaje en el tiempo y escritura para tablas de Apache Iceberg que utilizan el formato Apache Parquet para los datos y el catálogo de AWS Glue para su metaalmacén. Para obtener más información, consulte <a href="#">Uso de tablas de Apache Iceberg</a> .	26 de noviembre de 2021
Se agregó documentación sobre consultas federadas entre cuentas.	Puede utilizar la característica de consulta federada entre cuentas para consultar orígenes de datos en otra cuenta. Para obtener información sobre cómo configurar permisos para habilitar esta característica, consulte <a href="#">Habilitación de las consultas federadas entre cuentas</a> .	12 de noviembre de 2021
Se agregó documentación para la instrucción UNLOAD de Athena.	Utilice la instrucción UNLOAD para escribir los resultados de una instrucción SELECT a los formatos Apache Parquet, ORC, Apache Avro y JSON. Para obtener más información, consulte <a href="#">UNLOAD</a> .	5 de agosto de 2021
Se agregó documentación para la característica de la instrucción EXPLAIN de Athena.	Para obtener más información, consulte <a href="#">Uso de EXPLAIN y EXPLAIN ANALYZE en Athena</a> y <a href="#">Descripción de los resultados de la instrucción EXPLAIN de Athena</a> .	5 de abril de 2021

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se agregaron páginas sobre resolución de problemas y ajuste de rendimiento en Athena.	Para obtener más información, consulte <a href="#">Solución de problemas en Athena</a> y <a href="#">Ajuste del rendimiento en Athena</a> .	30 de diciembre de 2020
Se agregó documentación para el control de versiones del motor Athena y la versión 2 del motor Athena.	Para obtener más información, consulte <a href="#">Control de versiones del motor Athena</a> .	11 de noviembre de 2020
Se actualizó la documentación de consulta federada para la versión de disponibilidad general.	Para obtener más información, consulte <a href="#">Uso de consulta federada de Amazon Athena</a> y <a href="#">Uso de Athena con claves de contexto CalledVia</a> .	11 de noviembre de 2020
Se agregó documentación para utilizar el controlador JDBC con Lake Formation para el acceso federado a Athena.	Para obtener más información, consulte <a href="#">Uso de Lake Formation y los controladores JDBC y ODBC de Athena para el acceso federado a Athena</a> y <a href="#">Tutorial: Configuración de acceso federado a Athena para usuarios de Okta mediante Lake Formation y JDBC</a> .	25 de septiembre de 2020

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se agregó documentación para el conector de datos OpenSearch de Amazon Athena.	Para obtener más información, consulte <a href="#">Conector para OpenSearch de Amazon Athena</a> .	21 de julio de 2020
Se agregó documentación para consultas de conjuntos de datos de Hudi.	Para obtener más información, consulte <a href="#">Uso de Athena para consultar conjuntos de datos de Apache Hudi</a> .	9 de julio de 2020
Se agregó documentación sobre la consulta de registros del servidor web Apache y registros del servidor web IIS almacenados en Amazon S3.	Para obtener más información, consulte <a href="#">Consulta de registros de Apache almacenados en Amazon S3</a> y <a href="#">Consulta de los registros de Internet Information Services (IIS) almacenados en Simple Storage Service (Amazon S3)</a> .	8 de julio de 2020
Se agregó documentación para la versión general de Conector de datos de Athena para metaalmacén externo de Hive.	Para obtener más información, consulte <a href="#">Uso del conector de datos de Athena para metaalmacén externo de Hive</a> .	1 de junio de 2020

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se agregó documentación para etiquetar recursos del catálogo de datos.	Para obtener más información, consulte <a href="#">Etiquetado de recursos de Athena</a> .	1 de junio de 2020
Se añadió documentación sobre la proyección de particiones.	Para obtener más información, consulte <a href="#">Proyección de particiones con Amazon Athena</a> .	21 de mayo de 2020
Se actualizaron los ejemplos de código Java para Athena.	Para obtener más información, consulte <a href="#">Ejemplos de código</a> .	11 de mayo de 2020
Se agregó un tema sobre la consulta de resultados de Amazon GuardDuty.	Para obtener más información, consulte <a href="#">Consulta de los resultados de Amazon GuardDuty</a> .	19 de marzo de 2020
Se agregó un tema sobre el uso de CloudWatch Events para monitorear las transiciones de estado de consultas de Athena.	Para obtener más información, consulte <a href="#">Supervisión de eventos de Athena con Amazon EventBridge</a> .	11 de marzo de 2020

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se agregó un tema sobre consultas de registros de flujo de AWS Global Accelerator con Athena.	Para obtener más información, consulte <a href="#">Consulta de los registros de flujo de AWS Global Accelerator</a> .	6 de febrero de 2020

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
<ul style="list-style-type: none"> <li>Se añadió documentación sobre el uso de CTAS con INSERT INTO para agregar datos de un origen no particionado a un destino particionado.</li> <li>Se añadieron enlaces de descarga para la versión de previsualización 1.1.0 del controlador ODBC para Athena.</li> <li>Se corrigió la descripción de la expresión regular SHOW DATABASES LIKE.</li> <li>Se corrigió la sintaxis <code>partitioned_by</code> en el tema de CTA.</li> </ul>	<p>Las actualizaciones de documentación incluyen, entre otros, los siguientes temas:</p> <ul style="list-style-type: none"> <li><a href="#">Uso de CTAS e INSERT INTO en ETL y análisis de datos</a></li> <li><a href="#">Conexión a Amazon Athena con ODBC</a> (ahora se incluyen las características de vista previa 1.1.0 en el controlador ODBC 1.1.2.)</li> <li><a href="#">SHOW DATABASES</a></li> <li><a href="#">CREATE TABLE AS</a></li> </ul>	<p>4 de febrero de 2020</p>

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
<ul style="list-style-type: none"> <li>Se realizaron otras correcciones menores.</li> </ul>		
<p>Se añadió documentación sobre el uso de CTAS con INSERT INTO para agregar datos de un origen particionado a un destino particionado.</p>	<p>Para obtener más información, consulte <a href="#">Uso de CTAS e INSERT INTO para evitar el límite de 100 particiones</a>.</p>	<p>22 de enero de 2020</p>
<p>Se actualizó la información de ubicación de los resultados de consulta.</p>	<p>Athena ya no crea una ubicación de los resultados de consulta “predeterminada”. Para obtener más información, consulte <a href="#">Especificación de una ubicación de resultados de consulta</a>.</p>	<p>20 de enero de 2020</p>
<p>Se añadió un tema sobre consultas en AWS Glue Data Catalog. Se actualizó la información sobre las cuotas de servicio (anteriormente, “límites de servicio”) en Athena.</p>	<p>Para obtener más información, consulte los temas siguientes:</p> <ul style="list-style-type: none"> <li><a href="#">Consulta de AWS Glue Data Catalog</a></li> <li><a href="#">Service Quotas</a></li> </ul>	<p>17 de enero de 2020</p>



Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se corrigió el tema sobre OpenCSVSerDe para recordar que el tipo TIMESTAMP debería especificarse en formato numérico UNIX.	Para obtener más información, consulte <a href="#">OpenCSVSerDe para el procesamiento de CSV</a> .	15 de enero de 2020
Se actualizó el tema de seguridad sobre el cifrado para recordar que Athena no admite claves asimétricas.	Athena admite solo claves simétricas para lectura y escritura de datos. Para obtener más información, consulte <a href="#">Opciones de cifrado de Simple Storage Service (Amazon S3) compatibles</a> .	8 de enero de 2020
Se agregó información sobre el acceso entre cuentas a buckets de Amazon S3 cifrados con una clave de AWS KMS personalizada.	Para obtener más información, consulte <a href="#">Acceso entre cuentas a un bucket cifrado con una clave de AWS KMS personalizada</a> .	13 de diciembre de 2019

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
<p>Se añadió documentación para consultas federadas, metaalmacenes externos de Hive, machine learning y funciones definidas por el usuario. Se agregaron nuevas métricas de CloudWatch.</p>	<p>Para obtener más información, consulte los temas siguientes:</p> <ul style="list-style-type: none"><li>• <a href="#">Uso de consulta federada de Amazon Athena</a></li><li>• <a href="#">Conectores de orígenes de datos disponibles</a></li><li>• <a href="#">Uso del conector de datos de Athena para metaalmacén externo de Hive</a></li><li>• <a href="#">Uso de Machine Learning (ML) con Amazon Athena</a></li><li>• <a href="#">Consulta con funciones definidas por el usuario</a></li><li>• <a href="#">Lista de métricas y dimensiones de CloudWatch para Athena</a></li></ul>	<p>26 de noviembre de 2019</p>
<p>Se añadió una sección sobre el nuevo comando INSERT INTO y se actualizó la información de ubicación del resultado de la consulta para admitir los archivos de manifiesto de datos.</p>	<p>Para obtener más información, consulte <a href="#">INSERT INTO</a> y <a href="#">Trabajo con resultados de las consultas, consultas recientes y archivos de salida</a>.</p>	<p>18 de septiembre de 2019</p>

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
<p>Se añadió una sección sobre la compatibilidad de los puntos de conexión de VPC de tipo interfaz (PrivateLink). Se actualizaron los controladores JDBC. Se actualizó la información sobre los registros de flujo de VPC enriquecidos.</p>	<p>Para obtener más información, consulte <a href="#">Conexión a Amazon Athena mediante un punto de conexión de VPC de tipo interfaz</a>, <a href="#">Consulta de los registros de flujo de Amazon VPC</a> y <a href="#">Conexión a Amazon Athena con JDBC</a>.</p>	<p>11 de septiembre de 2019</p>
<p>Se añadió una sección sobre la integración con AWS Lake Formation.</p>	<p>Para obtener más información, consulte <a href="#">Uso de Athena para consultar datos registrados en AWS Lake Formation</a>.</p>	<p>26 de junio de 2019</p>
<p>Se actualizó la sección de seguridad para que sea coherente con otros servicios de AWS.</p>	<p>Para obtener más información, consulte <a href="#">Seguridad de Amazon Athena</a>.</p>	<p>26 de junio de 2019</p>

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se añadió una sección acerca de cómo consultar los registros de AWS WAF.	Para obtener más información, consulte <a href="#">Consulta de registros de AWS WAF</a> .	31 de mayo de 2019
Se publicó la nueva versión del controlador ODBC con soporte para grupos de trabajo de Athena.	<p>Para descargar el nuevo controlador ODBC versión 1.0.5 y su documentación, consulte <a href="#">Conexión a Amazon Athena con ODBC</a>. No se hacen cambios en la cadena de conexión del controlador ODBC cuando utiliza etiquetas en grupos de trabajo. Para utilizar las etiquetas, actualice a la versión más reciente del controlador ODBC, que es esta versión actual.</p> <p>Esta versión del controlador le permite utilizar las <a href="#">acciones de grupo de trabajo de la API de Athena</a> para crear y administrar grupos de trabajo, y las <a href="#">acciones de etiquetas de la API de Athena</a> para agregar, enumerar o retirar etiquetas de grupos de trabajo. Antes de comenzar, asegúrese de que dispone de permisos de nivel de recursos en IAM para acciones en grupos de trabajo y etiquetas.</p>	5 de marzo de 2019

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se agregó compatibilidad con la etiqueta para grupos de trabajo en Amazon Athena.	Una etiqueta consta de una clave y un valor, ambos definidos por el usuario. Al etiquetar un grupo de trabajo, puede asignarle metadatos personalizados. Por ejemplo, cree un grupo de trabajo para cada centro de costos. A continuación, mediante la adición de etiquetas a estos grupos de trabajo, puede hacer un seguimiento del gasto de Athena para cada centro de costos. Para obtener más información, consulte <a href="#">Uso de etiquetas para facturación</a> en la Guía del usuario de AWS Billing and Cost Management.	22 de febrero de 2019

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se mejoró el SerDe JSON de OpenX utilizado en Athena.	<p>Las mejoras incluyen, entre otras, lo siguiente:</p> <ul style="list-style-type: none"><li>• Compatibilidad para la propiedad <code>ConvertDotToJsonKeysToUnderscores</code>. Cuando se establece <code>TRUE</code>, permite que SerDe sustituya los puntos en los nombres de claves con guiones bajos. Por ejemplo, si el conjunto de datos de JSON contiene una clave con el nombre "a.b", puede utilizar esta propiedad para definir el nombre de la columna para que sea "a_b" en Athena. El valor predeterminado es <code>FALSE</code>. De forma predeterminada, Athena no permite puntos en los nombres de columnas.</li><li>• Compatibilidad para la propiedad <code>case.insensitive</code>. De forma predeterminada, Athena exige que todas las claves de su conjunto de datos JSON utilicen minúscula. El uso de <code>WITH SERDE PROPERTIES ("case.insensitive"=FALSE;)</code> le permite usar nombres de clave que distinguen entre mayúsculas y minúsculas en sus datos. El valor predeterminado es <code>TRUE</code>. Cuando se establece en <code>TRUE</code>, el SerDe convierte todas las columnas en mayúscula a minúscula.</li></ul> <p>Para obtener más información, consulte <a href="#">El SerDe JSON de OpenX</a>.</p>	18 de febrero de 2019

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se añadió compatibilidad para grupos de trabajo.	Utilice grupos de trabajo para separar usuarios, equipos, aplicaciones o cargas de trabajo y establecer límites en la cantidad de datos que puede procesar cada consulta o todo el grupo de trabajo. Como los grupos de trabajo funcionan como recursos de IAM, puede utilizar permisos de nivel de recursos para controlar el acceso a un grupo de trabajo específico. También puede ver métricas relacionadas con las consultas en Amazon CloudWatch, controlar los costos de las consultas mediante la configuración de los límites de la cantidad de datos escaneados, crear los umbrales y desencadenar acciones, como alarmas de Amazon SNS, cuando se superan estos umbrales. Para obtener más información, consulte <a href="#">Uso de grupos de trabajo para la ejecución de consultas</a> y <a href="#">Control de costos y supervisión de consultas con métricas y eventos de CloudWatch</a> .	18 de febrero de 2019
Se agregó compatibilidad con el análisis de registros del Equilibrador de carga de red.	Se agregaron consultas de Athena de ejemplo para analizar registros del Equilibrador de carga de red. Estos registros reciben información detallada de las solicitudes de seguridad de la capa de transporte (TLS) enviadas al Equilibrador de carga de red. Puede utilizar estos registros de acceso para analizar los patrones de tráfico y solucionar problemas. Para obtener más información, consulte <a href="#">the section called “Network Load Balancer”</a> .	24 de enero de 2019

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
<p>Se lanzaron las nuevas versiones del controlador JDBC y ODBC con compatibilidad para acceso federado a la API de Athena con Active Directory Federation Services (AD FS) y SAML 2.0 (Lenguaje de marcado para confirmaciones de seguridad 2.0).</p>	<p>Con esta versión de los controladores, el acceso federado a Athena es compatible con Active Directory Federation Service (AD FS 3.0). El acceso se establece a través de las versiones de los controladores JDBC u ODBC que admiten SAML 2.0. Para obtener información acerca de la configuración del acceso federado a la API de Athena, consulte <a href="#">the section called “Habilitación de acceso federado a la API de Athena”</a>.</p>	<p>10 de noviembre de 2018</p>



Cambio	Descripción	Fecha de lanzamiento de la nueva versión
<p>Se agregó compatibilidad con el control de acceso detallado a bases de datos y tablas en Athena. Además, se agregaron políticas en Athena que permiten cifrar los metadatos de las bases de datos y tablas en el Catálogo de datos.</p>	<p>Se agregó compatibilidad con la creación de políticas basadas en identidad (IAM) que proporcionan un control de acceso detallado a los recursos de AWS Glue Data Catalog, como las bases de datos y tablas que se utilizan en Athena.</p> <p>Además, puede cifrar los metadatos de las base de datos y tablas en el Catálogo de datos mediante la adición de políticas específicas a Athena.</p> <p>Para obtener más información, consulte <a href="#">Acceso detallado a las bases de datos y las tablas en AWS Glue Data Catalog</a>.</p>	<p>15 de octubre de 2018</p>
<p>Se agregó compatibilidad con las instrucciones CREATE TABLE AS SELECT.</p> <p>Mejoras adicionales en la documentación.</p>	<p>Se agregó compatibilidad con las instrucciones CREATE TABLE AS SELECT. Consulte <a href="#">Creación de una tabla a partir de los resultados de una consulta (CTAS)</a>, <a href="#">Consideraciones y limitaciones de las consultas CTAS</a> y <a href="#">Ejemplos de consultas CTAS</a>.</p>	<p>10 de octubre de 2018</p>

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
<p>Se lanzó la versión 1.0.3 del controlador ODBC con compatibilidad para la transmisión de resultados en lugar de obtenerlos por páginas.</p> <p>Mejoras adicionales en la documentación.</p>	<p>La versión 1.0.3 del controlador ODBC permite la transmisión de resultados e incluye también mejoras, corrección de errores y documentación actualizada para “Uso de SSL con un servidor proxy”.</p> <p>Para descargar el nuevo controlador ODBC versión 1.0.3 y su documentación, consulte <a href="#">Conexión a Amazon Athena con ODBC</a>.</p>	<p>6 de septiembre de 2018</p>
<p>Se lanzó la versión 2.0.5 del controlador JDBC con compatibilidad predeterminada para la transmisión de resultados en lugar de obtenerlos por páginas.</p> <p>Mejoras adicionales en la documentación.</p>	<p>Se lanzó la versión 2.0.5 del controlador JDBC con compatibilidad predeterminada para la transmisión de resultados en lugar de obtenerlos por páginas. Para obtener más información, consulte <a href="#">Conexión a Amazon Athena con JDBC</a>.</p>	<p>16 de agosto de 2018</p>

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
<p>Se actualizó la documentación para consultar los registros de flujo de Amazon Virtual Private Cloud, que se pueden almacenar directamente en Amazon S3 con formato GZIP.</p> <p>Se actualizaron los ejemplos de consulta de registros ALB.</p>	<p>Se actualizó la documentación para consultar los registros de flujo de Amazon Virtual Private Cloud, que se pueden almacenar directamente en Amazon S3 con formato GZIP. Para obtener más información, consulte <a href="#">Consulta de los registros de flujo de Amazon VPC</a>.</p> <p>Se actualizaron los ejemplos de consulta de registros ALB. Para obtener más información, consulte <a href="#">Consulta de los registros de Application Load Balancer</a>.</p>	7 de agosto de 2018
<p>Se ha agregado compatibilidad con vistas. Se añadieron instrucciones sobre las manipulaciones de esquemas para varios formatos de almacenamiento de datos.</p>	<p>Se ha agregado compatibilidad con vistas. Para obtener más información, consulte <a href="#">Uso de vistas</a>.</p> <p>Se actualizó esta guía con información sobre la gestión de las actualizaciones de esquema para varios formatos de almacenamiento de datos. Para obtener más información, consulte <a href="#">Gestión de las actualizaciones de los esquemas</a>.</p>	5 de junio de 2018

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se incrementó el límite predeterminado de consultas simultáneas de cinco a veinte.	Puede enviar y ejecutar hasta veinte consultas DDL y veinte consultas SELECT a la vez. Para obtener más información, consulte <a href="#">Service Quotas</a> .	17 de mayo de 2018
Se añadieron pestañas de consulta y la posibilidad de configurar la función de autocompletar en el editor de consultas.	Se añadieron pestañas de consulta y la posibilidad de configurar la función de autocompletar en el editor de consultas. Para obtener más información, consulte <a href="#">Introducción</a> .	8 de mayo de 2018
Se lanzó la versión 2.0.2 del controlador JDBC.	Se lanzó la nueva versión del controlador JDBC (versión 2.0.2). Para obtener más información, consulte <a href="#">Conexión a Amazon Athena con JDBC</a> .	19 de abril de 2018
Se agregó la función de autocompletar al escribir consultas en la consola de Athena.	Se agregó la función de autocompletar al escribir consultas en la consola de Athena.	6 de abril de 2018

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se agregó la capacidad de crear tablas de Athena para archivos de registro de CloudTrail directamente desde la consola de CloudTrail.	Se agregó la capacidad de crear tablas de Athena de manera automática para archivos de registro de CloudTrail directamente desde la consola de CloudTrail. Para obtener más información, consulte <a href="#">Uso de la consola de CloudTrail para crear una tabla de Athena para registros de CloudTrail</a> .	15 de marzo de 2018
Se añadió compatibilidad para la descarga de datos intermedios en disco de forma segura para las consultas con GROUP BY.	Se ha añadido la capacidad de descargar de forma segura datos intermedios en disco para consultas con uso intensivo de memoria que utilicen la cláusula GROUP BY. Esto mejora la fiabilidad de dichas consultas, para evitar los errores "Recurso de consulta agotado". Para obtener más información, consulte las notas de la versión de <a href="#">2 de febrero de 2018</a> .	2 de febrero de 2018
Se añadió compatibilidad para Presto versión 0.172.	Se actualizó el motor subyacente en Amazon Athena a una versión basada en Presto versión 0.172. Para obtener más información, consulte las notas de la versión de <a href="#">19 de enero de 2018</a> .	19 de enero de 2018
Se añadió compatibilidad para el controlador ODBC.	Se añadió compatibilidad para conectar Athena al controlador ODBC. Para obtener más información, consulte <a href="#">Conexión a Amazon Athena con ODBC</a> .	13 de noviembre de 2017

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
<p>Se añadió compatibilidad con las regiones de Asia-Pacífico (Bombay), Asia-Pacífico (Seúl) y Europa (Londres) . Se añadió compatibilidad para consultar datos geoespaciales.</p>	<p>Se agregó compatibilidad con consultas de datos geoespaciales y con las regiones de Asia-Pacífico (Bombay), Asia-Pacífico (Seúl) y Europa (Londres). Para obtener información, consulte <a href="#">Consultas de datos geoespaciales</a> y <a href="#">Puntos de conexión y Regiones de AWS</a>.</p>	<p>1 de noviembre de 2017</p>
<p>Se agregó compatibilidad con la región de Europa (Fráncfort).</p>	<p>Se agregó compatibilidad con la región de Europa (Fráncfort). Para obtener una lista de las regiones compatibles, consulte <a href="#">Puntos de conexión y Regiones de AWS</a>.</p>	<p>19 de octubre de 2017</p>
<p>Se agregó compatibilidad para las consultas de Athena con nombre con AWS CloudFormation.</p>	<p>Se agregó compatibilidad para la creación de consultas de Athena con nombre con AWS CloudFormation. Para obtener más información, consulte <a href="#">AmazonWebService::Athena::NamedQuery</a> en la Guía del usuario de AWS CloudFormation.</p>	<p>3 de octubre de 2017</p>
<p>Se agregó compatibilidad con la región de Asia Pacífico (Sídney).</p>	<p>Se añadió compatibilidad con la región de Asia-Pacífico (Sídney). Para obtener una lista de las regiones compatibles, consulte <a href="#">Puntos de conexión y Regiones de AWS</a>.</p>	<p>25 de septiembre de 2017</p>

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se agregó una sección a esta guía para consultar los registros del Servicio de AWS y diferentes tipos de datos, como mapas, matrices, datos anidados y datos que contienen JSON.	Se agregaron ejemplos para <a href="#">Consulta de registros de Servicio de AWS</a> y para consultar distintos tipos de datos en Athena. Para obtener más información, consulte <a href="#">Ejecución de consultas SQL mediante Amazon Athena</a> .	5 de septiembre de 2017
Se agregó compatibilidad con AWS Glue Data Catalog.	Se agregó integración con AWS Glue Data Catalog y un asistente de migración para actualizar del catálogo de datos administrado por Athena a AWS Glue Data Catalog. Para obtener más información, consulte <a href="#">Integración con AWS Glue</a> y <a href="#">AWS Glue</a> .	14 de agosto de 2017
Se añadió compatibilidad con SerDe de Grok.	Se añadió compatibilidad con el SerDe de Grok, que proporciona una asociación de patrones más sencilla de los registros que se encuentran en archivos de texto no estructurados como registros. Para obtener más información, consulte <a href="#">El SerDe de Grok</a> . Se agregaron métodos abreviados de teclado para desplazarse por el historial de consultas utilizando la consola.	4 de agosto de 2017
Se añadió compatibilidad con la región de Asia-Pacífico (Tokio).	Se añadió compatibilidad con las regiones de Asia-Pacífico (Tokio) y Asia-Pacífico (Singapur). Para obtener una lista de las regiones compatibles, consulte <a href="#">Puntos de conexión y Regiones de AWS</a> .	22 de junio de 2017

Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se agregó compatibilidad con la región de Europa (Irlanda).	Se agregó compatibilidad con la región de Europa (Irlanda). Para obtener más información, consulte <a href="#">Puntos de conexión y Regiones de AWS</a> .	8 de junio de 2017
Se agregó una API de Amazon Athena y compatibilidad con AWS CLI.	Se agregó una API de Amazon Athena y compatibilidad con AWS CLI para Athena. Se actualizó el controlador JDBC a la versión 1.1.0.	19 de mayo de 2017
Se agregó compatibilidad con el cifrado de datos de Amazon S3.	Se agregó compatibilidad con el cifrado de datos de Amazon S3 y se lanzó una actualización del controlador JDBC (versión 1.0.1) con compatibilidad con el cifrado, mejoras y correcciones de errores. Para obtener más información, consulte <a href="#">Cifrado en reposo</a> .	4 de abril de 2017



Cambio	Descripción	Fecha de lanzamiento de la nueva versión
Se agregó el SerDe de AWS CloudTrail.	<p>Se añadió el SerDe de AWS CloudTrail, se mejoró el rendimiento y se corrigieron problemas de partición.</p> <ul style="list-style-type: none"> <li>• El SerDe de AWS CloudTrail ha sido reemplazado por el <a href="#">El SerDe JSON de Hive</a> de lectura de registros de CloudTrail. Para obtener más información sobre la consulta de registros de CloudTrail, consulte <a href="#">Consulta de registros de AWS CloudTrail</a>.</li> <li>• Se mejoró el rendimiento cuando se analiza un gran número de particiones.</li> <li>• Se mejoró el rendimiento en la operación <code>MSCK Repair Table</code>.</li> <li>• Se añadió la capacidad de consultar datos de Amazon S3 almacenados en regiones distintas de la región principal. Se aplican tasas estándar de transferencia de datos entre regiones para Amazon S3, además de las tasas estándar de Athena.</li> </ul>	24 de marzo de 2017
Se agregó compatibilidad con la región Este de EE. UU. (Ohio).	Se agregó compatibilidad para <a href="#">El SerDe de Avro</a> y <a href="#">OpenCSVSerDe para el procesamiento de CSV</a> , la región Este de EE. UU. (Ohio) y la edición masiva de columnas en el asistente de la consola. Se ha mejorado el rendimiento en las tablas Parquet grandes.	20 de febrero de 2017
	La versión inicial de la Guía del usuario de Amazon Athena.	Noviembre de 2016

# Glosario de AWS

Para ver la terminología más reciente de AWS, consulte el [Glosario de AWS](#) en la Referencia de Glosario de AWS.