Verwenden Sie Apache Iceberg auf AWS

AWS Präskriptive Leitlinien

AWS Präskriptive Leitlinien: Verwenden Sie Apache Iceberg auf AWS

Table of Contents

Apache Iceberg verwenden auf AWS

Amazon Web Services (Mitwirkende)

November 2025 (Verlauf der Dokumente)

Apache Iceberg ist ein Open-Source-Tabellenformat, das die Tabellenverwaltung vereinfacht und gleichzeitig die Leistung verbessert. AWS Analysedienste wie Amazon EMR AWS Glue, Amazon Athena und Amazon Redshift bieten native Unterstützung für Iceberg, sodass Sie auf einfache Weise transaktionale Data Lakes auf Amazon Simple Storage Service (Amazon S3) aufbauen können. AWS

Darüber hinaus SageMaker basiert die nächste Generation von Amazon auf einer offenen Lakehouse-Architektur, die den Datenzugriff über Data Lakes, Data Warehouses sowie Drittanbieter- und Verbundquellen hinweg AWS vereinheitlicht. The Lakehouse ist vollständig mit Iceberg kompatibel und bietet Ihnen die Flexibilität, mithilfe der Iceberg-REST-API vor Ort auf Daten zuzugreifen und diese abzufragen.

Dieser technische Leitfaden bietet Anleitungen für die ersten Schritte mit Iceberg auf verschiedenen AWS-Services Plattformen und enthält bewährte Methoden und Empfehlungen für den skalierbaren Betrieb von Iceberg AWS bei gleichzeitiger Kosten- und Leistungsoptimierung.

Egal, ob Sie gerade erst mit Iceberg beginnen oder ein erfahrener Benutzer sind, der Ihre bestehenden Iceberg-Workloads optimieren möchte AWS, dieser Leitfaden bietet wertvolle Einblicke für jede Phase Ihres Projekts

In diesem Leitfaden:

Moderne Data Lakes

Fortgeschrittene Anwendungsfälle in modernen Data Lakes

Die Entwicklung der Datenspeicherung hat sich von Datenbanken hin zu Data Warehouses und Data Lakes entwickelt, bei denen jede Technologie auf einzigartige Geschäfts- und Datenanforderungen zugeschnitten ist. Herkömmliche Datenbanken waren bei der Verarbeitung strukturierter Daten und transaktionaler Workloads hervorragend, standen jedoch mit zunehmenden Datenmengen vor Leistungsproblemen. Data Warehouses wurden entwickelt, um Leistungs- und Skalierbarkeitsprobleme zu lösen, aber wie Datenbanken stützten sie sich auf proprietäre Formate in vertikal integrierten Systemen.

Data Lakes bieten eine der besten Optionen für die Speicherung von Daten in Bezug auf Kosten, Skalierbarkeit und Flexibilität. Sie können einen Data Lake verwenden, um große Mengen strukturierter und unstrukturierter Daten kostengünstig zu speichern und diese Daten für verschiedene Arten von Analytics-Workloads zu verwenden, von Business Intelligence-Berichten über Big-Data-Verarbeitung, Echtzeitanalysen, maschinelles Lernen und generative künstliche Intelligenz (KI), um bessere Entscheidungen zu treffen.

Trotz dieser Vorteile wurden Data Lakes ursprünglich nicht mit datenbankähnlichen Funktionen konzipiert. Ein Data Lake bietet keine Unterstützung für die Verarbeitungssemantik Atomicity, Consistency, Isolation, Durability (ACID), die Sie möglicherweise benötigen, um Ihre Daten mithilfe vieler verschiedener Technologien effektiv zu optimieren und zu verwalten, und zwar im großen Maßstab für Hunderte oder Tausende von Benutzern. Data Lakes bieten keine native Unterstützung für die folgenden Funktionen:

- Durchführung effizienter Aktualisierungen und Löschungen auf Datensatzebene, wenn sich Daten in Ihrem Unternehmen ändern
- Verwaltung der Abfrageleistung, wenn Tabellen auf Millionen von Dateien und Hunderttausende von Partitionen anwachsen
- Sicherstellung der Datenkonsistenz bei mehreren gleichzeitigen Schreib- und Lesegeräten
- Verhinderung von Datenbeschädigungen, wenn Schreibvorgänge während des Vorgangs fehlschlagen
- Weiterentwicklung von Tabellenschemas im Laufe der Zeit, ohne Datensätze (teilweise) neu zu schreiben

Diese Herausforderungen haben sich in Anwendungsfällen wie der Handhabung von Change Data Capture (CDC) oder in Anwendungsfällen, die den Datenschutz, das Löschen von Daten und die Aufnahme von Streaming-Daten betreffen, besonders ausgeprägt, was zu suboptimalen Tabellen führen kann.

Data Lakes, die traditionelle Tabellen im HIVE-Format verwenden, unterstützen Schreibvorgänge nur für ganze Dateien. Dies macht die Implementierung von Aktualisierungen und Löschungen schwierig, zeitaufwändig und kostspielig. Darüber hinaus sind Parallelitätskontrollen und -garantien erforderlich, die in ACID-konformen Systemen angeboten werden, um die Integrität und Konsistenz der Daten zu gewährleisten.

Diese Herausforderungen stellen die Benutzer vor ein Dilemma: Sie müssen sich zwischen einer vollständig integrierten, aber proprietären Plattform entscheiden oder sich für einen herstellerneutralen, aber ressourcenintensiven, selbst erstellten Data Lake entscheiden, der ständig gewartet und migriert werden muss, um seinen potenziellen Wert auszuschöpfen.

Um diese Herausforderungen zu bewältigen, bietet Iceberg zusätzliche datenbankähnliche Funktionen, die den Optimierungs- und Verwaltungsaufwand von Data Lakes vereinfachen und gleichzeitig die Speicherung auf kostengünstigen Systemen wie Amazon S3 unterstützen.

Einführung in Apache Iceberg

Apache Iceberg ist ein Open-Source-Tabellenformat, das Funktionen in Data-Lake-Tabellen bereitstellt, die in der Vergangenheit nur in Datenbanken oder Data Warehouses verfügbar waren. Es ist auf Skalierbarkeit und Leistung ausgelegt und eignet sich hervorragend für die Verwaltung von Tabellen mit einer Größe von mehr als Hunderten von Gigabyte. Einige der Hauptmerkmale von Iceberg-Tabellen sind:

- Löschen, aktualisieren und zusammenführen.Iceberg unterstützt Standard-SQL-Befehle für Data Warehousing zur Verwendung mit Data-Lake-Tabellen.
- Schnelle Scanplanung und erweiterte Filterung. Iceberg speichert Metadaten wie Statistiken auf Partitions- und Spaltenebene, die von Engines verwendet werden können, um die Planung und Ausführung von Abfragen zu beschleunigen.
- Vollständige Entwicklung des Schemas. Iceberg unterstützt das Hinzufügen, Löschen, Aktualisieren oder Umbenennen von Spalten ohne Nebenwirkungen.
- Entwicklung der Partition. Sie können das Partitionslayout einer Tabelle aktualisieren, wenn sich das Datenvolumen oder die Abfragemuster ändern. Iceberg unterstützt das Ändern der Spalten,

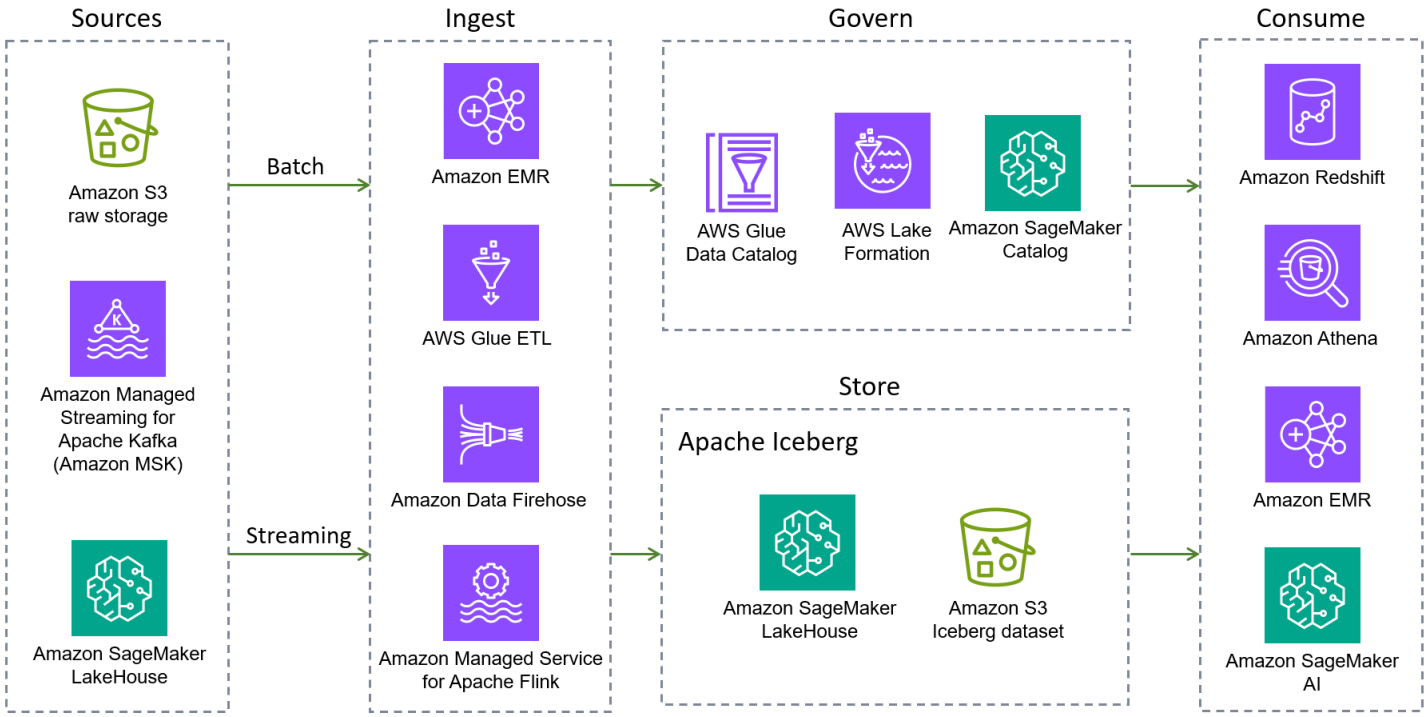
auf die eine Tabelle partitioniert ist, oder das Hinzufügen von Spalten zu oder das Entfernen von Spalten aus zusammengesetzten Partitionen.

- Versteckte Partitionierung.Diese Funktion verhindert, dass unnötige Partitionen automatisch gelesen werden. Dadurch müssen Benutzer die Partitionierungsdetails der Tabelle nicht verstehen oder ihren Abfragen zusätzliche Filter hinzufügen.
- Rollback der Version. Benutzer können Probleme schnell korrigieren, indem sie zu einem Zustand vor der Transaktion zurückkehren.
- Zeitreise. Benutzer können eine bestimmte frühere Version einer Tabelle abfragen.
- Serialisierbare Isolierung. Tabellenänderungen sind atomar, sodass Leser keine teilweisen oder nicht festgeschriebenen Änderungen sehen.
- Gleichzeitige Autoren. Iceberg verwendet optimistische Parallelität, um den Erfolg mehrerer Transaktionen zu ermöglichen. Im Falle von Konflikten muss einer der Autoren die Transaktion erneut versuchen.
- Öffnen Sie Dateiformate. Iceberg unterstützt mehrere Open-Source-Dateiformate, darunter Apache Parquet, Apache Avro und Apache ORC.

Zusammenfassend lässt sich sagen, dass Data Lakes, die das Iceberg-Format verwenden, von Transaktionskonsistenz, Geschwindigkeit, Skalierbarkeit und Schemaentwicklung profitieren. Weitere Informationen zu diesen und anderen Iceberg-Funktionen finden Sie in der Apache Iceberg-Dokumentation.

AWS Unterstützung für Apache Iceberg

Apache Iceberg wird unter AWS-Services anderem von Amazon EMR, Amazon Athena, Amazon Redshift und Amazon unterstützt. AWS Glue SageMaker Das folgende Diagramm zeigt eine vereinfachte Referenzarchitektur eines Data Lakes, der auf Iceberg basiert.



Im Folgenden finden Sie native AWS-Services Iceberg-Integrationen. Es gibt weitere AWS-Services , die entweder indirekt oder durch Paketierung der Iceberg-Bibliotheken mit Iceberg interagieren können.

- Amazon S3 ist aufgrund seiner Beständigkeit, Verfügbarkeit, Skalierbarkeit, Sicherheit, Compliance und Auditfunktionen der beste Ort für den Aufbau von Data Lakes. Iceberg wurde für die nahtlose Interaktion mit Amazon S3 konzipiert und entwickelt und bietet Unterstützung für viele Amazon S3 S3-Funktionen, die in der Iceberg-Dokumentation aufgeführt sind. Darüber hinaus bieten Amazon S3 S3-Tabellen den ersten Cloud-Objektspeicher mit integrierter Iceberg-Unterstützung und rationalisieren die Speicherung von Tabellendaten in großem Maßstab. Mit der Unterstützung von S3 Tables für Iceberg können Sie Ihre Tabellendaten ganz einfach abfragen, indem Sie beliebte AWS Abfrage-Engines und Abfrage-Engines von Drittanbietern verwenden.
- Die nächste Generation von SageMaker basiert auf einer offenen Lakehouse-Architektur, die den Datenzugriff über Amazon S3 S3-Data Lakes, Amazon Redshift Redshift-Data Warehouses sowie Drittanbieter- und Verbunddatenquellen vereinheitlicht. Diese Funktionen helfen Ihnen dabei, leistungsstarke Analysen und AI/ML Anwendungen auf einer einzigen Datenkopie zu erstellen. The Lakehouse ist vollständig mit Iceberg kompatibel, sodass Sie mithilfe der Iceberg-REST-API flexibel auf Daten zugreifen und diese abfragen können.
- Amazon EMR ist eine Big-Data-Lösung für Datenverarbeitung im Petabyte-Bereich, interaktive Analysen und maschinelles Lernen unter Verwendung von Open-Source-Frameworks wie Apache

Spark, Flink, Trino und Hive. Amazon EMR kann auf benutzerdefinierten Amazon Elastic Compute Cloud (Amazon EC2) -Clustern, Amazon Elastic Kubernetes Service (Amazon EKS) oder Amazon EMR AWS Outposts Serverless ausgeführt werden.

- Amazon Athena ist ein serverloser, interaktiver Analysedienst, der auf Open-Source-Frameworks basiert. Er unterstützt Open-Table- und Dateiformate und bietet eine vereinfachte, flexible Möglichkeit, Petabyte an Daten dort zu analysieren, wo sie sich befinden. Athena bietet native Unterstützung für Lese-, Zeitreise-, Schreib- und DDL-Abfragen für Iceberg und verwendet den Metastore AWS Glue Data Catalog for the Iceberg.
- Amazon Redshift ist ein Cloud-Data Warehouse im Petabyte-Bereich, das sowohl clusterbasierte als auch serverlose Bereitstellungsoptionen unterstützt. Amazon Redshift Spectrum kann externe Tabellen abfragen, die bei Amazon S3 registriert AWS Glue Data Catalog und dort gespeichert sind. Redshift Spectrum unterstützt auch das Iceberg-Speicherformat.
- AWS Glueist ein serverloser Datenintegrationsdienst, der es einfacher macht, Daten aus verschiedenen Quellen für Analysen, maschinelles Lernen (ML) und Anwendungsentwicklung zu finden, aufzubereiten, zu verschieben und zu integrieren. Er ist vollständig in Iceberg integriert. Insbesondere können Sie Lese- und Schreiboperationen an Iceberg-Tabellen mithilfe von AWS Glue Jobs ausführen, Tabellen über den AWS Glue Data Catalog(Hive-Metastore-kompatibel) verwalten, Tabellen mithilfe von AWS Glue Crawlern automatisch erkennen und registrieren und die Datenqualität in Iceberg-Tabellen mithilfe der Datenqualitätsfunktion bewerten. AWS Glue Das unterstützt AWS Glue Data Catalog auch das Sammeln von Spaltenstatistiken, das Berechnen und Aktualisieren der Anzahl unterschiedlicher Werte (NDVs) für jede Spalte in Iceberg-Tabellen sowie automatische Tabellenoptimierungen (Komprimierung, Aufbewahrung von Snapshots, Löschen verwaister Dateien). AWS Glue unterstützt auch Zero-ETL-Integrationen aus einer Liste von Anwendungen und Drittanbieteranwendungen in Iceberg-Tabellen AWS-Services .
- Amazon Data Firehose ist ein vollständig verwalteter Service für die Bereitstellung von Echtzeit-Streaming-Daten an Ziele wie Amazon S3, Amazon Redshift, Amazon OpenSearch Service, Amazon OpenSearch Serverless, Splunk, Apache Iceberg-Tabellen und alle benutzerdefinierten HTTP- oder HTTP-Endpunkte unterstützter Drittanbieter wie Datadog, Dynatrace, MongoDB, New Relic, Coralogix und Elastic. LogicMonitor Mit Firehose müssen Sie keine Anwendungen schreiben oder Ressourcen verwalten. Sie konfigurieren Ihre Datenproduzenten zum Senden von Daten an Firehose. Die Daten werden dann automatisch an das angegebene Ziel geliefert. Sie können Firehose auch so konfigurieren, dass die Daten vor der Bereitstellung transformiert werden.
- Amazon Managed Service für Apache Flink ist ein vollständig verwalteter Amazon-Service, mit dem Sie eine Apache Flink-Anwendung zur Verarbeitung von Streaming-Daten verwenden können. Er

unterstützt sowohl das Lesen aus Iceberg-Tabellen als auch das Schreiben in Iceberg-Tabellen und ermöglicht Datenverarbeitung und -analyse in Echtzeit.

- Amazon SageMaker AI unterstützt die Speicherung von Feature-Sets im Amazon SageMaker AI Feature Store mithilfe des Iceberg-Formats.
- AWS Lake Formationbietet grobe und detaillierte Zugriffsberechtigungen für den Zugriff auf Daten, einschließlich Iceberg-Tabellen, die von Athena oder Amazon Redshift verwendet werden. Weitere Informationen zur Unterstützung von Berechtigungen für Iceberg-Tabellen finden Sie in der Dokumentation zu Lake Formation.

AWS bietet eine breite Palette von Diensten, die Iceberg unterstützen, aber das Abdecken all dieser Dienste würde den Rahmen dieses Leitfadens sprengen. Die folgenden Abschnitte behandeln Spark (Batch- und strukturiertes Streaming) auf Amazon EMR und AWS Glue Athena SQL. Der folgende Abschnitt bietet einen kurzen Überblick über die Iceberg-Unterstützung in Athena SQL.

Erste Schritte mit Iceberg-Tabellen in Amazon Athena SQL

Amazon Athena bietet integrierten Support für Iceberg. Sie können Iceberg ohne zusätzliche Schritte oder Konfiguration verwenden, mit Ausnahme der Einrichtung der Servicevoraussetzungen, die im Abschnitt Erste Schritte der Athena-Dokumentation beschrieben sind. Dieser Abschnitt bietet eine kurze Einführung in das Erstellen von Tabellen in Athena. Weitere Informationen finden Sie unter Arbeiten mit Iceberg-Tabellen mithilfe von Athena SQL weiter unten in diesem Handbuch.

Sie können Iceberg-Tabellen AWS mithilfe verschiedener Engines erstellen. Diese Tabellen funktionieren problemlos. AWS-Services Um Ihre ersten Iceberg-Tabellen mit Athena SQL zu erstellen, können Sie den folgenden Boilerplate-Code verwenden.

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

Die folgenden Abschnitte enthalten Beispiele für die Erstellung von partitionierten und unpartitionierten Iceberg-Tabellen in Athena. Weitere Informationen finden Sie in der Iceberg-Syntax in der Athena-Dokumentation.

Eine unpartitionierte Tabelle erstellen

Die folgende Beispielanweisung passt den SQL-Standardcode an, um eine unpartitionierte Iceberg-Tabelle in Athena zu erstellen. Sie können diese Anweisung zum Abfrage-Editor in der Athena-Konsole hinzufügen, um die Tabelle zu erstellen.

```
CREATE TABLE athena_iceberg_table (
    color string,
    date string,
    name string,
    price bigint,
    product string,
```

```
    ts timestamp)
LOCATION 's3://DOC_EXAMPLE_BUCKET/ice_warehouse/iceberg_db/athena_iceberg_table/'
TBLPROPERTIES (
    'table_type' ='ICEBERG'
)
```

step-by-stepAnweisungen zur Verwendung des Abfrage-Editors finden Sie unter Erste Schritte in der Athena-Dokumentation.

Eine partitionierte Tabelle erstellen

Die folgende Anweisung erstellt eine partitionierte Tabelle auf der Grundlage des Datums, wobei das Konzept der versteckten Partitionierung von Iceberg verwendet wird. Sie verwendet die `day()` Transformation, um tägliche Partitionen unter Verwendung des `dd-mm-yyyy` Formats aus einer Zeitstempelspalte abzuleiten. Iceberg speichert diesen Wert nicht als neue Spalte im Datensatz. Stattdessen wird der Wert spontan abgeleitet, wenn Sie Daten schreiben oder abfragen.

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

Erstellen einer Tabelle und Laden von Daten mit einer einzigen CTAS-Anweisung

In den partitionierten und unpartitionierten Beispielen in den vorherigen Abschnitten wurden die Iceberg-Tabellen als leere Tabellen erstellt. Sie können Daten in die Tabellen laden, indem Sie die Anweisung OR verwenden. `INSERT MERGE` Alternativ können Sie eine `CREATE TABLE AS SELECT (CTAS)` Anweisung verwenden, um Daten in einem einzigen Schritt zu erstellen und in eine Iceberg-Tabelle zu laden.

CTAS ist die beste Methode in Athena, um eine Tabelle zu erstellen und Daten in einer einzigen Anweisung zu laden. Das folgende Beispiel zeigt, wie CTAS verwendet wird, um eine Iceberg-Tabelle (`iceberg_ctas_table`) aus einer vorhandenen Hive/Parquet Tabelle (`hive_table`) in Athena zu erstellen.

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

Weitere Informationen zu CTAS finden Sie in der Athena CTAS-Dokumentation.

Daten einfügen, aktualisieren und löschen

Athena unterstützt verschiedene Arten, Daten mithilfe der Anweisungen,`INSERT INTO`, `UPDATE` und `DELETE FRO M` in eine Iceberg-Tabelle zu schreiben. `MERGE INTO`

> **Note**
>
> Athena SQL unterstützt den copy-on-write Ansatz derzeit nicht. `UPDATEMERGE INTO`, und `DELETE FROM` Operationen verwenden immer den merge-on-read Ansatz mit Positionslöschungen, unabhängig von den angegebenen Tabelleneigenschaften. Wenn Sie Tabelleneigenschaften wie`write.update.mode`, und `write.delete.mode` zur Verwendung copy-on-write einrichten`write.merge.mode`, schlagen Ihre Abfragen nicht fehl, aber Athena ignoriert sie und verwendet merge-on-read sie weiterhin.

Mit der folgenden Anweisung werden Daten `INSERT INTO` zu einer Iceberg-Tabelle hinzugefügt:

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

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

Beispielausgabe:

Results (1)

Copy Download results

Search rows

1

#	color	date	name	price	product	ts
1	red	222022-07-19T03:47:29	PersonNew	178	Tuna	2023-10-11 11:35:01.298000 UTC

Weitere Informationen finden Sie in der Athena-Dokumentation.

Iceberg-Tabellen abfragen

Sie können reguläre SQL-Abfragen für Ihre Iceberg-Tabellen ausführen, indem Sie Athena SQL verwenden, wie im vorherigen Beispiel dargestellt.

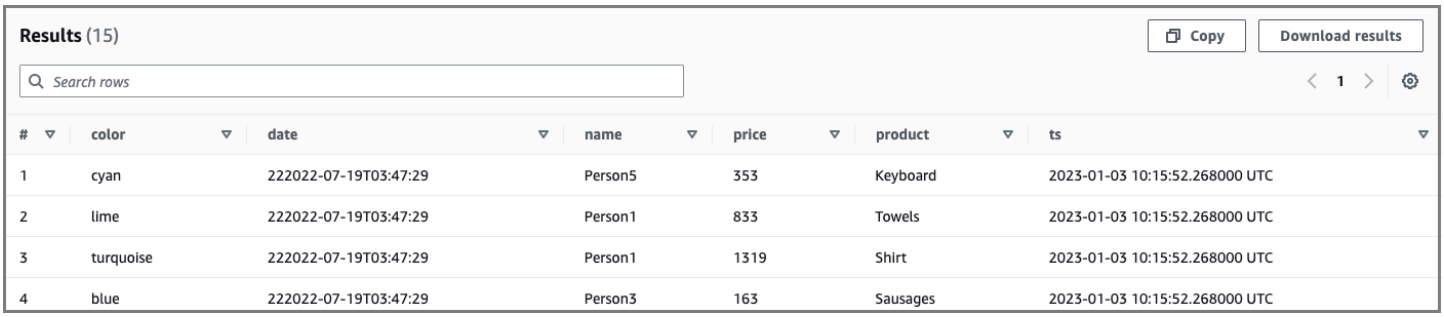
Zusätzlich zu den üblichen Abfragen unterstützt Athena auch Zeitreiseabfragen für Eisberg-Tabellen. Wie bereits erwähnt, können Sie vorhandene Datensätze durch Aktualisierungen oder Löschungen in einer Iceberg-Tabelle ändern. Daher ist es praktisch, Zeitreiseabfragen zu verwenden, um anhand eines Zeitstempels oder einer Snapshot-ID in ältere Versionen Ihrer Tabelle zurückzuschauen.

Die folgende Anweisung aktualisiert beispielsweise einen Farbwert für `Person5` und zeigt dann einen früheren Wert vom 4. Januar 2023 an:

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

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

Beispielausgabe:



Results (15)

Copy Download results

Search rows

1

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

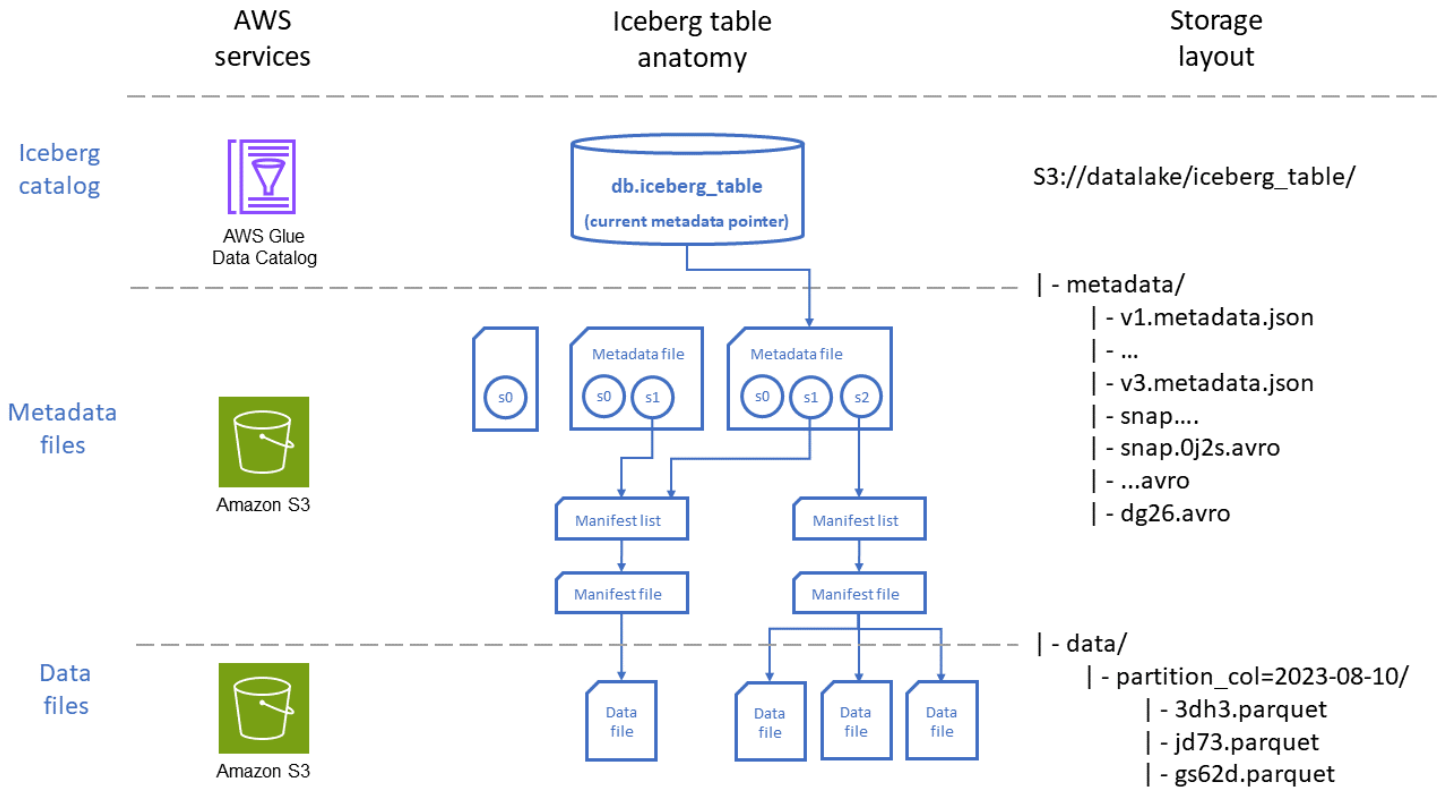
Syntax und weitere Beispiele für Zeitreiseabfragen finden Sie in der Athena-Dokumentation.

Anatomie der Eisbergtabelle

Nachdem wir nun die grundlegenden Schritte der Arbeit mit Iceberg-Tischen behandelt haben, wollen wir uns eingehender mit den komplizierten Details und dem Design eines Iceberg-Tisches befassen.

Um die zuvor in diesem Handbuch beschriebenen Funktionen nutzen zu können, wurde Iceberg mit hierarchischen Ebenen von Daten- und Metadatendateien entworfen. Diese Ebenen verwalten Metadaten auf intelligente Weise, um die Planung und Ausführung von Abfragen zu optimieren.

Das folgende Diagramm zeigt die Organisation einer Iceberg-Tabelle aus zwei Perspektiven: die zum Speichern der Tabelle AWS-Services verwendete und die Dateiplatzierung in Amazon S3.



Wie in der Abbildung dargestellt, besteht eine Eisbergtabelle aus drei Hauptebenen:

- Iceberg-Katalog: AWS Glue Data Catalog lässt sich nativ in Iceberg integrieren und ist in den meisten Anwendungsfällen die beste Option für Workloads, die auf ausgeführt werden. AWS Dienste, die mit Iceberg-Tabellen interagieren (z. B. Athena), verwenden den Katalog, um die aktuelle Snapshot-Version der Tabelle zu finden, entweder um Daten zu lesen oder zu schreiben.
- Metadatenebene: Metadatendateien, nämlich die Manifestdateien und die Manifestlistendateien, verfolgen Informationen wie das Schema der Tabellen, die Partitionsstrategie und den Speicherort der Datendateien sowie Statistiken auf Spaltenebene wie Mindest- und Höchstbereiche für die

Datensätze, die in jeder Datendatei gespeichert sind. Diese Metadatendateien werden in Amazon S3 innerhalb des Tabellenpfads gespeichert.

- Manifestdateien enthalten einen Datensatz für jede Datendatei, einschließlich Speicherort, Format, Größe, Prüfsumme und anderer relevanter Informationen.
- Manifestlisten stellen einen Index der Manifestdateien bereit. Wenn die Anzahl der Manifestdateien in einer Tabelle zunimmt, trägt die Aufteilung dieser Informationen in kleinere Unterabschnitte dazu bei, die Anzahl der Manifestdateien zu reduzieren, die durch Abfragen gescannt werden müssen.
- Metadatendateien enthalten Informationen über die gesamte Iceberg-Tabelle, einschließlich der Manifestlisten, Schemas, Partitionsmetadaten, Snapshot-Dateien und anderer Dateien, die zur Verwaltung der Tabellenmetadaten verwendet werden.

- Datenschicht: Diese Ebene enthält die Dateien mit den Datensätzen, für die Abfragen ausgeführt werden. Diese Dateien können in verschiedenen Formaten gespeichert werden, darunter Apache Parquet, Apache Avro und Apache ORC.
  - Datendateien enthalten die Datensätze für eine Tabelle.
  - Mit gelöschten Dateien werden Lösch - und Aktualisierungsvorgänge auf Zeilenebene in einer Iceberg-Tabelle kodiert. In Iceberg gibt es zwei Arten von Löschdateien, wie in der Iceberg-Dokumentation beschrieben. Diese Dateien werden durch Operationen unter Verwendung des merge-on-read Modus erstellt.

Arbeiten mit Iceberg in Amazon EMR

Amazon EMR bietet Datenverarbeitung im Petabyte-Bereich, interaktive Analysen und maschinelles Lernen in der Cloud mithilfe von Open-Source-Frameworks wie Apache Spark, Apache Hive, Flink und Trino.

> **Note**
>
> In diesem Handbuch wird Apache Spark als Beispiele verwendet.

Amazon EMR unterstützt mehrere Bereitstellungsoptionen: Amazon EMR on EC2, Amazon EMR on EKS, Amazon EMR Serverless und Amazon EMR on. AWS Outposts Informationen zur Auswahl einer Bereitstellungsoption für Ihren Workload finden Sie in den häufig gestellten Fragen zu Amazon EMR.

Versions- und Funktionskompatibilität

Amazon EMR Version 6.5.0 und spätere Versionen unterstützen Apache Iceberg nativ. Eine Liste der unterstützten Iceberg-Versionen für jede Amazon EMR-Version finden Sie im Iceberg-Versionsverlauf in der Amazon EMR-Dokumentation. Lesen Sie auch die Abschnitte unter Einen Cluster mit Iceberg verwenden, um zu erfahren, welche Iceberg-Funktionen in Amazon EMR auf verschiedenen Frameworks unterstützt werden.

Wir empfehlen Ihnen, die neueste Amazon EMR-Version zu verwenden, um von der neuesten unterstützten Iceberg-Version zu profitieren. Bei den Codebeispielen und Konfigurationen in diesem Abschnitt wird davon ausgegangen, dass Sie die Amazon EMR-Version emr-7.8.0 verwenden.

Erstellen eines Amazon EMR-Clusters mit Iceberg

Um einen Amazon EMR-Cluster auf Amazon EC2 mit installiertem Iceberg zu erstellen, folgen Sie den Anweisungen in der Amazon EMR-Dokumentation.

Insbesondere sollte Ihr Cluster mit der folgenden Klassifizierung konfiguriert sein:

```
[{
    "Classification": "iceberg-defaults",
    "Properties": {
```

```
        "iceberg.enabled": "true"
    }
}]
```

Ab Amazon EMR 6.6.0 können Sie auch Amazon EMR Serverless oder Amazon EMR on EKS als Bereitstellungsoptionen für Ihre Iceberg-Workloads verwenden.

Entwicklung von Iceberg-Anwendungen in Amazon EMR

Um den Spark-Code für Ihre Iceberg-Anwendungen zu entwickeln, können Sie Amazon EMR Studio verwenden, eine webbasierte integrierte Entwicklungsumgebung (IDE) für vollständig verwaltete Jupyter-Notebooks, die auf Amazon EMR-Clustern ausgeführt werden.

Verwenden von Amazon EMR Studio-Notizbüchern

Sie können Spark-Anwendungen interaktiv in Amazon EMR Studio Workspace-Notebooks entwickeln und diese Notebooks mit Ihren Amazon EMR auf EC2 Clustern oder Amazon EMR auf EKS verwalteten Endpunkten verbinden. Anweisungen zur Einrichtung von EMR Studio für Amazon EMR on EC2 und Amazon EMR on EKS finden Sie in der AWS-Service Dokumentation.

Gehen Sie wie folgt vor, um Iceberg in EMR Studio zu verwenden:

1. Starten Sie einen Amazon EMR-Cluster mit aktiviertem Iceberg, wie unter Verwenden Sie einen Cluster mit installiertem Iceberg beschrieben.
2. Richten Sie ein EMR Studio ein. Anweisungen finden Sie unter Amazon EMR Studio einrichten.
3. Öffnen Sie ein EMR Studio Workspace-Notizbuch und führen Sie den folgenden Code als erste Zelle im Notizbuch aus, um Ihre Spark-Sitzung für die Verwendung von Iceberg zu konfigurieren:

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

Wobei:

- `<catalog_name>`ist der Name Ihres Iceberg Spark-Sitzungskatalogs. Ersetzen Sie ihn durch einen Namen Ihrer Wahl und denken Sie daran, die Verweise in allen Konfigurationen, die mit diesem Katalog verknüpft sind, zu ändern. In Ihrem Code können Sie wie folgt auf Ihre Iceberg-Tabellen verweisen, indem Sie den vollqualifizierten Tabellennamen, einschließlich des Namens des Spark-Sitzungskatalogs, verwenden:

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

  Alternativ können Sie den Standardkatalog in den Iceberg-Katalog ändern, den Sie definiert haben, indem Sie `spark.sql.defaultCatalog` auf Ihren Katalognamen setzen. Dieser zweite Ansatz ermöglicht es Ihnen, auf Tabellen ohne das Katalogpräfix zu verweisen, was Ihre Abfragen vereinfachen kann.

- `<catalog_name>.warehouse`verweist auf den Amazon S3 S3-Pfad, in dem Sie Ihre Daten und Metadaten speichern möchten.
- Um den Katalog zu einem zu machen AWS Glue Data Catalog, setzen Sie `spark.sql.catalog.<catalog_name>.type` auf`glue`. Dieser Schlüssel ist erforderlich, um auf eine Implementierungsklasse für jede benutzerdefinierte Katalogimplementierung zu verweisen. Im Abschnitt Allgemeine bewährte Methoden weiter unten in diesem Handbuch werden die verschiedenen von Iceberg unterstützten Kataloge beschrieben.

4. Sie können jetzt wie bei jeder anderen Spark-Anwendung mit der interaktiven Entwicklung Ihrer Spark-Anwendung für Iceberg im Notizbuch beginnen.

Weitere Informationen zur Konfiguration von Spark für Apache Iceberg mithilfe von Amazon EMR Studio finden Sie im Blogbeitrag Build a high-performance, ACID-compliant, evolving data lake using Apache Iceberg on Amazon EMR.

Iceberg-Jobs in Amazon EMR ausführen

Nachdem Sie den Spark-Anwendungscode für Ihren Iceberg-Workload entwickelt haben, können Sie ihn auf jeder Amazon EMR-Bereitstellungsoption ausführen, die Iceberg unterstützt (siehe häufig gestellte Fragen zu Amazon EMR).

Wie bei anderen Spark-Jobs können Sie Arbeit an einen Amazon EMR auf einem EC2 Cluster einreichen, indem Sie Schritte hinzufügen oder Spark-Jobs interaktiv an den Master-Knoten senden.

Informationen zum Ausführen eines Spark-Jobs finden Sie auf den folgenden Amazon EMR-Dokumentationsseiten:

- Eine Übersicht über die verschiedenen Optionen zum Einreichen von Arbeit an einen Amazon EMR auf einem EC2 Cluster und detaillierte Anweisungen für jede Option finden Sie unter Arbeit an ein Cluster einreichen.
- Informationen zu Amazon EMR auf EKS finden Sie unter Spark-Jobs ausführen mit StartJobRun.
- Informationen zu EMR Serverless finden Sie unter Jobs ausführen.

Die folgenden Abschnitte enthalten ein Beispiel für jede Amazon EMR-Bereitstellungsoption.

Amazon EMR auf EC2

Gehen Sie wie folgt vor, um den Iceberg Spark-Job einzureichen:

1. Erstellen Sie die Datei `emr_step_iceberg.json` mit dem folgenden Inhalt auf Ihrer Workstation:

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

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

2. Ändern Sie die Konfigurationsdatei für Ihren speziellen Spark-Job, indem Sie die fett hervorgehobenen Iceberg-Konfigurationsoptionen anpassen.

3. Reichen Sie den Schritt mit AWS Command Line Interface ()AWS CLI ein. Führen Sie den Befehl in dem Verzeichnis aus, in dem sich die `emr_step_iceberg.json` Datei befindet.

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

Amazon EMR Serverless

Um einen Iceberg Spark-Job an EMR Serverless zu senden, verwenden Sie: AWS CLI

1. Erstellen Sie die Datei `emr_serverless_iceberg.json` mit dem folgenden Inhalt auf Ihrer Workstation:

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

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

```
            }
        }
    }
}
```

2. Ändern Sie die Konfigurationsdatei für Ihren speziellen Spark-Job, indem Sie die fett hervorgehobenen Iceberg-Konfigurationsoptionen anpassen.
3. Senden Sie den Job mit dem. AWS CLI Führen Sie den Befehl in dem Verzeichnis aus, in dem sich die `emr_serverless_iceberg.json` Datei befindet:

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

So senden Sie mithilfe der EMR Studio-Konsole einen Iceberg Spark-Job an EMR Serverless:

1. Folgen Sie den Anweisungen in der EMR Serverless-Dokumentation.
2. Verwenden Sie für die Jobkonfiguration die für Spark bereitgestellte Iceberg-Konfiguration für Spark AWS CLI und passen Sie die hervorgehobenen Felder für Iceberg an. Eine ausführliche Anleitung finden Sie unter Using Apache Iceberg with EMR Serverless in der Amazon EMR-Dokumentation.

Amazon EMR in EKS

Um einen Iceberg Spark-Job an Amazon EMR auf EKS zu senden, verwenden Sie: AWS CLI

1. Erstellen Sie die Datei `emr_eks_iceberg.json` mit dem folgenden Inhalt auf Ihrer Workstation:

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

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

2. Ändern Sie die Konfigurationsdatei für Ihren Spark-Job, indem Sie die fett hervorgehobenen Iceberg-Konfigurationsoptionen anpassen.
3. Senden Sie den Job mit dem. AWS CLI Führen Sie den folgenden Befehl in dem Verzeichnis aus, in dem sich die `emr_eks_iceberg.json` Datei befindet:

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

Eine ausführliche Anleitung finden Sie unter Using Apache Iceberg with Amazon EMR on EKS in der Dokumentation zu Amazon EMR on EKS.

Bewährte Methoden für Amazon EMR

Dieser Abschnitt enthält allgemeine Richtlinien für die Optimierung von Spark-Jobs in Amazon EMR, um das Lesen und Schreiben von Daten in Iceberg-Tabellen zu optimieren. Iceberg-spezifische Best Practices finden Sie im Abschnitt „Bewährte Methoden“ weiter unten in diesem Handbuch.

- Verwenden Sie die neueste Version von Amazon EMR — Amazon EMR bietet mit der Amazon EMR Spark-Laufzeit sofort Spark-Optimierungen. AWS verbessert die Leistung der Spark-Runtime-Engine mit jeder neuen Version.
- Ermitteln Sie die optimale Infrastruktur für Ihre Spark-Workloads — Spark-Workloads benötigen möglicherweise unterschiedliche Hardwaretypen für unterschiedliche Jobmerkmale, um eine optimale Leistung zu gewährleisten. Amazon EMR unterstützt verschiedene Instance-Typen (z. B. rechenoptimiert, speicheroptimiert, universell einsetzbar und speicheroptimiert), um alle Arten von Verarbeitungsanforderungen abzudecken. Wenn Sie neue Workloads integrieren, empfehlen wir, einen Benchmark mit allgemeinen Instance-Typen wie M5 oder M6g durchzuführen. Überwachen Sie das Betriebssystem (OS) und die YARN-Metriken von Ganglia und Amazon, CloudWatch um die Systemengpässe (CPU, Speicher, Speicher und I/O) bei Spitzenlast zu ermitteln und geeignete Hardware auszuwählen.
- Tune `spark.sql.shuffle.partitions` — Legen Sie die `spark.sql.shuffle.partitions` Eigenschaft auf die Gesamtzahl der virtuellen Kerne (vCores) in Ihrem Cluster oder auf ein Vielfaches dieses Werts fest (in der Regel das 1- bis 2-fache der Gesamtzahl der vCores). Diese Einstellung wirkt sich auf die Parallelität von Spark aus, wenn Sie Hash- und Bereichspartitionierung als Schreibverteilungsmodus verwenden. Vor dem Schreiben wird ein Shuffle angefordert, um die Daten zu organisieren, wodurch die Ausrichtung der Partitionen gewährleistet wird.
- Verwaltete Skalierung aktivieren — Für fast alle Anwendungsfälle empfehlen wir, die verwaltete Skalierung und die dynamische Zuweisung zu aktivieren. Wenn Sie jedoch einen Workload haben, der ein vorhersehbares Muster aufweist, empfehlen wir Ihnen, die automatische Skalierung und die dynamische Zuweisung zu deaktivieren. Wenn die verwaltete Skalierung aktiviert ist, empfehlen wir die Verwendung von Spot-Instances, um die Kosten zu senken. Verwenden Sie Spot-Instances für Task-Knoten anstelle von Core- oder Master-Knoten. Wenn Sie Spot-Instances verwenden, verwenden Sie Instance-Flotten mit mehreren Instance-Typen pro Flotte, um die Spot-Verfügbarkeit sicherzustellen.
- Verwenden Sie nach Möglichkeit Broadcast-Join — Der Broadcast-Join (mapside) ist der optimalste Join, sofern eine Ihrer Tabellen klein genug ist, um in den Speicher Ihres kleinsten Knotens (in der Reihenfolge von MBs) zu passen, und Sie einen Equi (=) -Join durchführen. Alle Join-Typen mit Ausnahme von vollständigen Outer-Joins werden unterstützt. Ein Broadcast-Join überträgt die kleinere Tabelle als Hashtabelle an alle Worker-Knoten im Speicher. Nachdem die kleine Tabelle übertragen wurde, können Sie keine Änderungen mehr daran vornehmen. Da sich die Hashtabelle lokal in der Java Virtual Machine (JVM) befindet, kann sie anhand der Join-Bedingung mithilfe eines Hash-Joins problemlos mit der großen Tabelle zusammengeführt werden. Broadcast-Joins bieten aufgrund des minimalen Shuffle-Overheads eine hohe Leistung.

- Optimieren Sie den Garbage-Collector — Wenn die Garbage-Collection-Zyklen (GC) langsam sind, sollten Sie erwägen, für eine bessere Leistung vom standardmäßigen parallel Garbage-Collector auf G1GC umzusteigen. Um die GC-Leistung zu optimieren, können Sie die GC-Parameter fein abstimmen. Um die GC-Leistung zu verfolgen, können Sie sie mithilfe der Spark-Benutzeroberfläche überwachen. Idealerweise sollte die GC-Zeit weniger als oder gleich 1 Prozent der gesamten Aufgabenlaufzeit sein.

Zusammenarbeit mit Iceberg in AWS Glue

AWS Glueist ein serverloser Datenintegrationsdienst, der es einfacher macht, Daten aus verschiedenen Quellen für Analysen, maschinelles Lernen (ML) und Anwendungsentwicklung zu entdecken, aufzubereiten, zu verschieben und zu integrieren. Eine der Kernfunktionen von AWS Glue ist die Fähigkeit, ETL-Operationen (Extrahieren, Transformieren und Laden) auf einfache und kostengünstige Weise durchzuführen. Auf diese Weise können Sie Ihre Daten kategorisieren, bereinigen, anreichern und zuverlässig zwischen verschiedenen Datenspeichern und Datenströmen verschieben.

AWS Glue Jobs kapseln Skripten, die Transformationslogik mithilfe einer Apache Spark - oder Python-Laufzeit definieren. AWS Glue Jobs können sowohl im Batch- als auch im Streaming-Modus ausgeführt werden.

Wenn Sie Iceberg-Jobs in erstellen AWS Glue, können Sie je nach Version von AWS Glue entweder die native Iceberg-Integration oder eine benutzerdefinierte Iceberg-Version verwenden, um Iceberg-Abhängigkeiten an den Job anzuhängen.

Verwenden Sie die native Iceberg-Integration

AWS Glue Die Versionen 3.0, 4.0 und 5.0 unterstützen nativ transaktionale Data-Lake-Formate wie Apache Iceberg, Apache Hudi und Linux Foundation Delta Lake in for Spark. AWS Glue Diese Integrationsfunktion vereinfacht die Konfigurationsschritte, die erforderlich sind, um mit der Verwendung dieser Frameworks in zu beginnen. AWS Glue

Um die Iceberg-Unterstützung für Ihren AWS Glue Job zu aktivieren, legen Sie den Job fest: Wählen Sie den Tab Jobdetails für Ihren AWS Glue Job, scrollen Sie unter Erweiterte Eigenschaften zu Job-Parametern und setzen Sie den Schlüssel `--datalake-formats` und seinen Wert auf`iceberg`.

Wenn Sie einen Job mit einem Notizbuch erstellen, können Sie den Parameter in der ersten Notebookzelle konfigurieren, indem Sie die `%%configure` Magie wie folgt verwenden:

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

Die `iceberg` Konfiguration für `--datalake-formats` in AWS Glue entspricht bestimmten Iceberg-Versionen, die auf Ihrer AWS Glue Version basieren:

AWS Glue Version	Standardversion von Iceberg
5.0	1.7.1
4,0	1.0.0
3.0	0.13.1

Verwenden Sie eine benutzerdefinierte Iceberg-Version

In einigen Situationen möchten Sie möglicherweise die Kontrolle über die Iceberg-Version für den Job behalten und sie in Ihrem eigenen Tempo aktualisieren. Ein Upgrade auf eine neuere Version kann beispielsweise den Zugriff auf neue Funktionen und Leistungsverbesserungen freischalten. Um eine bestimmte Iceberg-Version mit zu verwenden AWS Glue, können Sie Ihre eigenen JAR-Dateien bereitstellen.

Bevor Sie eine benutzerdefinierte Iceberg-Version implementieren, überprüfen Sie die Kompatibilität mit Ihrer AWS Glue Umgebung, indem Sie den Abschnitt AWS Glue Versionen der AWS Glue Dokumentation überprüfen. AWS Glue 5.0 erfordert beispielsweise Kompatibilität mit Spark 3.5.4.

Gehen Sie beispielsweise wie folgt vor, um AWS Glue Jobs auszuführen, die Iceberg Version 1.9.1 verwenden:

1. Besorgen Sie sich die erforderlichen JAR-Dateien und laden Sie sie auf Amazon S3 hoch:
   a. Laden Sie iceberg-spark-runtime-3.5_2.12-1.9.1.jar und -1.9.1.jar aus dem Apache Maven-Repository herunter. iceberg-aws-bundle
   b. Laden Sie diese Dateien an den von Ihnen angegebenen S3-Bucket-Speicherort hoch (z. B.). `s3://your-bucket-name/jars/`
2. Richten Sie die Job-Parameter für Ihren AWS Glue Job wie folgt ein:
   a. Geben Sie im `--extra-jars` Parameter den vollständigen S3-Pfad zu beiden JAR-Dateien an und trennen Sie sie durch ein Komma (z. B.`s3://your-bucket-name/jars/iceberg-spark-runtime-3.5_2.12-1.9.1.jar,s3://your-bucket-name/jars/iceberg-aws-bundle-1.9.1.jar`).
   b. Schließen Sie `iceberg` nicht als Wert für den Auftragsparameter `--datalake-formats` ein.

c. Wenn Sie AWS Glue 5.0 verwenden, müssen Sie den `--user-jars-first` Parameter auf `true` setzen.

Spark-Konfigurationen für Iceberg in AWS Glue

In diesem Abschnitt werden die Spark-Konfigurationen beschrieben, die für die Erstellung eines AWS Glue ETL-Jobs für einen Iceberg-Datensatz erforderlich sind. Sie können diese Konfigurationen festlegen, indem Sie den `--conf` Spark-Schlüssel mit einer kommagetrennten Liste aller Spark-Konfigurationsschlüssel und -werte verwenden. Sie können die `%%configure` Magie in einem Notizbuch oder im Bereich Job-Parameter der AWS Glue Studio Konsole verwenden.

```
%glue_version 5.0

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

Konfigurieren Sie die Spark-Sitzung mit den folgenden Eigenschaften:

- `<catalog_name>`ist der Name Ihres Iceberg Spark-Sitzungskatalogs. Ersetzen Sie ihn durch einen Namen Ihrer Wahl und denken Sie daran, die Verweise in allen Konfigurationen, die mit diesem Katalog verknüpft sind, zu ändern. In Ihrem Code können Sie wie folgt auf Ihre Iceberg-Tabellen verweisen, indem Sie den vollqualifizierten Tabellennamen, einschließlich des Namens des Spark-Sitzungskatalogs, verwenden:

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

  Alternativ können Sie den Standardkatalog in den Iceberg-Katalog ändern, den Sie definiert haben, indem Sie `spark.sql.defaultCatalog` auf Ihren Katalognamen setzen. Sie können diesen zweiten Ansatz verwenden, um auf Tabellen ohne das Katalogpräfix zu verweisen, was Ihre Abfragen vereinfachen kann.
- `<catalog_name>.<warehouse>`verweist auf den Amazon S3 S3-Pfad, in dem Sie Ihre Daten und Metadaten speichern möchten.
- Um den Katalog zu einem zu machen AWS Glue Data Catalog, setzen Sie `spark.sql.catalog.<catalog_name>.type` auf`glue`. Dieser Schlüssel ist erforderlich, um auf eine Implementierungsklasse für jede benutzerdefinierte Katalogimplementierung zu

verweisen. Informationen zu Katalogen, die von Iceberg unterstützt werden, finden Sie im Abschnitt Allgemeine bewährte Methoden weiter unten in diesem Handbuch.

Wenn Sie beispielsweise einen Katalog aufgerufen haben`glue_iceberg`, können Sie Ihren Job mithilfe mehrerer `--conf` Schlüssel wie folgt konfigurieren:

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

Alternativ können Sie Code verwenden, um die obigen Konfigurationen wie folgt zu Ihrem Spark-Skript hinzuzufügen:

```
spark = SparkSession.builder\

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

Bewährte Methoden für AWS Glue Jobs

Dieser Abschnitt enthält allgemeine Richtlinien für die Optimierung von Spark-Jobs, AWS Glue um das Lesen und Schreiben von Daten in Iceberg-Tabellen zu optimieren. Iceberg-spezifische Best Practices finden Sie im Abschnitt Bewährte Methoden weiter unten in diesem Handbuch.

- Verwenden Sie die neueste Version von AWS Glue und aktualisieren Sie sie, wann immer dies möglich ist. Neue Versionen von AWS Glue bieten Leistungsverbesserungen, kürzere Startzeiten und neue Funktionen. Sie unterstützen auch neuere Spark-Versionen, die möglicherweise für die

neuesten Iceberg-Versionen erforderlich sind. Eine Liste der verfügbaren AWS Glue Versionen und der Spark-Versionen, die sie unterstützen, finden Sie in der AWS Glue Dokumentation.

- AWS Glue Arbeitsspeicher optimieren — Folgen Sie den Empfehlungen im AWS Blogbeitrag Optimieren Sie die Speicherverwaltung in AWS Glue.
- AWS Glue Auto Scaling verwenden — Wenn Sie Auto Scaling aktivieren, passt die Anzahl der AWS Glue Worker AWS Glue automatisch dynamisch an Ihre Arbeitslast an. Dies trägt dazu bei, die Kosten Ihrer AWS Glue Arbeit bei Spitzenauslastung zu senken, da AWS Glue die Anzahl der Mitarbeiter reduziert wird, wenn die Arbeitslast gering ist und die Mitarbeiter untätig sitzen. Um AWS Glue Auto Scaling zu verwenden, geben Sie eine maximale Anzahl von Mitarbeitern an, auf die Ihr AWS Glue Job skaliert werden kann. Weitere Informationen finden Sie AWS Glue in der AWS Glue Dokumentation unter Using Auto Scaling for.
- Verwenden Sie die gewünschte Iceberg-Version — die AWS Glue native Integration für Iceberg eignet sich am besten für den Einstieg in Iceberg. Für Produktions-Workloads empfehlen wir jedoch, Bibliotheksabhängigkeiten hinzuzufügen (wie weiter oben in diesem Handbuch beschrieben), um die volle Kontrolle über die Iceberg-Version zu erhalten. Dieser Ansatz hilft Ihnen, bei Ihren Jobs von den neuesten Iceberg-Funktionen und Leistungsverbesserungen zu profitieren. AWS Glue
- Aktivieren Sie die Spark-Benutzeroberfläche für die Überwachung und das Debuggen — Sie können die Spark-Benutzeroberfläche auch verwenden, AWS Glue um Ihren Iceberg-Job zu überprüfen, indem Sie die verschiedenen Phasen eines Spark-Jobs in einem gerichteten azyklischen Graphen (DAG) visualisieren und die Jobs im Detail überwachen. Die Spark-Benutzeroberfläche bietet eine effektive Methode zur Fehlerbehebung und Optimierung von Iceberg-Jobs. So können Sie beispielsweise Engpass-Phasen identifizieren, in denen es zu häufigem Durcheinander oder zu viel Festplattenverlust kommt, um Optimierungsmöglichkeiten zu identifizieren. Weitere Informationen finden Sie in der Dokumentation unter Überwachen von Aufträgen mithilfe der Apache Spark-Weboberfläche. AWS Glue

Arbeiten mit Iceberg-Tabellen mithilfe von Apache Spark

Dieser Abschnitt bietet einen Überblick über die Verwendung von Apache Spark für die Interaktion mit Iceberg-Tabellen. Bei den Beispielen handelt es sich um Standardcode, der auf Amazon EMR oder ausgeführt werden kann. AWS Glue

Hinweis: Die primäre Schnittstelle für die Interaktion mit Iceberg-Tabellen ist SQL, daher kombinieren die meisten Beispiele Spark SQL mit der API. DataFrames

Iceberg-Tabellen erstellen und schreiben

Sie können Spark SQL und Spark verwenden, um Daten DataFrames zu Iceberg-Tabellen zu erstellen und hinzuzufügen.

Verwenden von Spark SQL

Verwenden Sie standardmäßige Spark-SQL-Anweisungen wie `CREATE TABLE` und`INSERT INTO`, um einen Iceberg-Datensatz zu schreiben.

Unpartitionierte Tabellen

Hier ist ein Beispiel für die Erstellung einer unpartitionierten Iceberg-Tabelle mit Spark SQL:

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

Verwenden Sie eine Standardanweisung, um Daten in eine unpartitionierte Tabelle einzufügen: `INSERT INTO`

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

```
INSERT INTO {CATALOG_NAME}.{DB_NAME}.{TABLE_NAME}_nopartitions
SELECT c_customer_sk, c_customer_id, c_first_name, c_last_name, c_birth_country,
 c_email_address
FROM another_table
""")
```

Partitionierte Tabellen

Hier ist ein Beispiel für die Erstellung einer partitionierten Iceberg-Tabelle mit Spark SQL:

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

Verwenden Sie eine Standardanweisung, um Daten mit Spark SQL in eine partitionierte Iceberg-Tabelle einzufügen: `INSERT INTO`

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

> **Note**
>
> Ab Iceberg 1.5.0 ist der `hash` Schreibverteilungsmodus der Standard, wenn Sie Daten in partitionierte Tabellen einfügen. Weitere Informationen finden Sie unter Verteilungsmodi schreiben in der Iceberg-Dokumentation.

Verwenden der API DataFrames

Um einen Iceberg-Datensatz zu schreiben, können Sie die `DataFrameWriterV2` API verwenden.

Verwenden Sie die Funktion `df.writeTo(` t), um eine Iceberg-Tabelle zu erstellen und Daten in diese zu schreiben. Wenn die Tabelle existiert, verwenden Sie die `.append()` Funktion. Ist dies nicht `.create().` der Fall`.createOrReplace()`, verwenden Sie die folgenden Beispiele use, was einer `.create()` Variante entsprich`CREATE OR REPLACE TABLE AS SELECT`.

Unpartitionierte Tabellen

So erstellen und füllen Sie eine unpartitionierte Iceberg-Tabelle mithilfe der API: `DataFrameWriterV2`

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

So fügen Sie mithilfe der API Daten in eine bestehende unpartitionierte Iceberg-Tabelle ein: `DataFrameWriterV2`

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

Partitionierte Tabellen

Um eine partitionierte Iceberg-Tabelle mithilfe der API zu erstellen und aufzufüllen: `DataFrameWriterV2`

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

So fügen Sie mithilfe der API Daten in eine partitionierte Iceberg-Tabelle ein: `DataFrameWriterV2`

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

Daten in Iceberg-Tabellen aktualisieren

Das folgende Beispiel zeigt, wie Daten in einer Iceberg-Tabelle aktualisiert werden. In diesem Beispiel werden alle Zeilen geändert, die eine gerade Zahl in der `c_customer_sk` Spalte haben.

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

Dieser Vorgang verwendet die copy-on-write Standardstrategie, sodass alle betroffenen Datendateien neu geschrieben werden.

Daten in Iceberg-Tabellen werden aktualisiert

Daten aktualisieren bezieht sich auf das Einfügen neuer Datensätze und das Aktualisieren vorhandener Datensätze in einer einzigen Transaktion. Um Daten in eine Iceberg-Tabelle hochzuladen, verwenden Sie die Anweisung. `SQL MERGE INTO`

Im folgenden Beispiel wird der Inhalt der Tabelle`{UPSERT_TABLE_NAME}` innerhalb der Tabelle verschoben: `{TABLE_NAME}`

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

- Wenn ein Kundendatensatz, der sich in `{UPSERT_TABLE_NAME}` befindet, bereits `{TABLE_NAME}` mit demselben vorhanden ist`c_customer_id`, überschreibt der `{UPSERT_TABLE_NAME}` `c_email_address` Datensatzwert den vorhandenen Wert (Aktualisierungsvorgang).
- Wenn ein Kundendatensatz, der `{UPSERT_TABLE_NAME}` sich in befindet, nicht existiert`{TABLE_NAME}`, wird der `{UPSERT_TABLE_NAME}` Datensatz hinzugefügt `{TABLE_NAME}` (Vorgang einfügen).

Löschen von Daten in Iceberg-Tabellen

Um Daten aus einer Iceberg-Tabelle zu löschen, verwenden Sie den `DELETE FROM` Ausdruck und geben Sie einen Filter an, der den zu löschenden Zeilen entspricht.

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

Wenn der Filter auf eine gesamte Partition zutrifft, führt Iceberg eine reine Metadaten-Löschung durch und belässt die Datendateien an Ort und Stelle. Andernfalls werden nur die betroffenen Datendateien neu geschrieben.

Die Methode delete verwendet die Datendateien, die von der `WHERE` Klausel betroffen sind, und erstellt eine Kopie davon ohne die gelöschten Datensätze. Anschließend wird ein neuer Tabellen-Snapshot erstellt, der auf die neuen Datendateien verweist. Daher sind die gelöschten Datensätze immer noch in den älteren Snapshots der Tabelle vorhanden. Wenn Sie beispielsweise den vorherigen Snapshot der Tabelle abrufen, werden Ihnen die Daten angezeigt, die Sie gerade gelöscht haben. Informationen zum Entfernen nicht benötigter alter Snapshots mit den zugehörigen Datendateien zu Säuberungszwecken finden Sie im Abschnitt Dateien mithilfe der Komprimierung verwalten weiter unten in diesem Handbuch.

Lesen von Daten

Sie können den aktuellen Status Ihrer Iceberg-Tabellen in Spark sowohl mit Spark SQL als auch lesen. DataFrames

Beispiel mit Spark SQL:

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

Beispiel mit der DataFrames API:

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

Zeitreisen nutzen

Jeder Schreibvorgang (Einfügen, Aktualisieren, Hochsetzen, Löschen) in einer Iceberg-Tabelle erstellt einen neuen Snapshot. Sie können diese Snapshots dann für Zeitreisen verwenden, um in die Vergangenheit zu reisen und den Status einer Tabelle in der Vergangenheit zu überprüfen.

Informationen zum Abrufen des Verlaufs von Snapshots für Tabellen mithilfe von Werten `snapshot-id` und Zeitangaben finden Sie im Abschnitt Zugreifen auf Metadaten weiter unten in diesem Handbuch.

Die folgende Zeitreiseabfrage zeigt den Status einer Tabelle auf der Grundlage eines bestimmten `snapshot-id` Status an.

Verwenden von Spark SQL:

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

Mithilfe der DataFrames API:

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

Die folgende Zeitreiseabfrage zeigt den Status einer Tabelle auf der Grundlage des letzten Snapshots, der vor einem bestimmten Zeitstempel erstellt wurde, in Millisekunden () an. `as-of-timestamp`

Verwenden von Spark SQL:

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

Mithilfe der DataFrames API:

```
df_1st_snapshot_ts = spark.read.option("as-of-timestamp", snapshot_ts) \
```

```
                        .format("iceberg") \
                        .load(f"dev.{DB_NAME}.{TABLE_NAME}") \
                        .limit(5)
```

Verwendung inkrementeller Abfragen

Sie können Iceberg-Snapshots auch verwenden, um angehängte Daten inkrementell zu lesen.

Hinweis: Derzeit unterstützt dieser Vorgang das Lesen von Daten aus Snapshots. `append` Das Abrufen von Daten aus Operationen wie `replace` `overwrite`, oder wird nicht unterstützt. `delete` Darüber hinaus werden inkrementelle Lesevorgänge in der Spark-SQL-Syntax nicht unterstützt.

Im folgenden Beispiel werden alle Datensätze abgerufen, die an eine Iceberg-Tabelle zwischen dem Snapshot `start-snapshot-id` (exklusiv) und `end-snapshot-id` (inklusive) angehängt wurden.

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

Zugreifen auf Metadaten

Iceberg bietet Zugriff auf seine Metadaten über SQL. Sie können auf die Metadaten für jede beliebige Tabelle (`<table_name>`) zugreifen, indem Sie den Namespace abfragen. `<table_name>.<metadata_table>` Eine vollständige Liste der Metadatentabellen finden Sie in der Iceberg-Dokumentation unter Tabellen überprüfen.

Das folgende Beispiel zeigt, wie Sie auf die Iceberg-Historien-Metadatentabelle zugreifen können, in der der Verlauf der Commits (Änderungen) für eine Iceberg-Tabelle angezeigt wird.

Verwenden von Spark SQL (mit der `%%sql` Magie) von einem Amazon EMR Studio-Notizbuch aus:

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

Mithilfe der DataFrames API:

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

Beispielausgabe:

Type: Table Pie Scatter Line Area Bar

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

Arbeiten mit Iceberg-Tabellen mithilfe von Trino

In diesem Abschnitt wird beschrieben, wie Sie Iceberg-Tabellen mithilfe von Trino auf Amazon EMR einrichten und betreiben. Bei den Beispielen handelt es sich um Standardcode, den Sie auf einem Amazon EMR on EC2-Cluster ausführen können. Bei den Codebeispielen und Konfigurationen in diesem Abschnitt wird davon ausgegangen, dass Sie die Amazon EMR-Version emr-7.9.0 verwenden.

Einrichtung von Amazon EMR auf EC2

1. Erstellen Sie eine `iceberg.properties` Datei mit dem folgenden Inhalt. Die `iceberg.file-format=parquet` Einstellung bestimmt das Standardspeicherformat für neue Tabellen, wenn das Format in der `CREATE TABLE` Anweisung nicht explizit angegeben ist.

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

2. Laden Sie die Datei `iceberg.properties` in Ihren S3-Bucket hoch.
3. Erstellen Sie eine Bootstrap-Aktion, die die `iceberg.properties` Datei aus Ihrem S3-Bucket kopiert und als Trino-Konfigurationsdatei auf dem Amazon EMR-Cluster speichert, den Sie erstellen werden. Stellen Sie sicher, dass Sie es durch Ihren `<S3-bucket-name>` S3-Bucket-Namen ersetzen.

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

4. Erstellen Sie einen Amazon EMR-Cluster mit installiertem Trino und geben Sie die Ausführung des vorherigen Skripts als Bootstrap-Aktion an. Hier ist ein Beispielbefehl AWS Command Line Interface (AWS CLI) zum Erstellen des Clusters:

```
aws emr create-cluster --release-label emr-7.9.0 \
--applications Name=Trino \
--region <region> \
--name Trino_Iceberg_Cluster \
```

```
--bootstrap-actions '[{"Path":"s3://<S3-bucket-name>/bootstrap.sh","Name":"Add
 iceberg.properties"}]' \
--instance-groups
 '[{"InstanceGroupType":"MASTER","InstanceCount":1,"InstanceType":"m5.xlarge"},
{"InstanceGroupType":"CORE","InstanceCount":3,"InstanceType":"m5.xlarge"}]' \
--service-role "<IAM-service-role>" \
--ec2-attributes '{"KeyName":"<key-name>","InstanceProfile":"<EMR-EC2-instance-
profile>"}'
```

wo Sie ersetzen:

- `<S3-bucket-name>`mit Ihrem S3-Bucket-Namen
- `<region>`mit deinem spezifischen AWS-Region
- `<key-name>`mit deinem key pair. Wenn das key pair nicht existiert, wird es erstellt.
- `<IAM-service-role>`mit Ihrer Amazon EMR-Servicerolle, die dem Prinzip der geringsten Rechte folgt.
- `<EMR-EC2-instance-profile>`mit Ihrem Instance-Profil.

5. Wenn der Amazon EMR-Cluster initialisiert wurde, können Sie eine Trino-Sitzung initialisieren, indem Sie den folgenden Befehl ausführen:

```
trino-cli
```

6. In der Trino CLI können Sie die Kataloge anzeigen, indem Sie Folgendes ausführen:

```
SHOW CATALOGS;
```

Erstellen von Iceberg-Tabellen

Um eine Iceberg-Tabelle zu erstellen, können Sie die Anweisung verwenden. `CREATE TABLE` Hier ist ein Beispiel für die Erstellung einer partitionierten Tabelle, die die versteckte Iceberg-Partitionierung verwendet:

```
CREATE TABLE iceberg.iceberg_db.iceberg_table (
             userid int,
             firstname varchar,
             city varchar)
        WITH (
             format = 'PARQUET',
             partitioning = ARRAY['city', 'bucket(userid, 16)'],
```

```
            location = 's3://<S3-bucket>/<prefix>');
```

> **Note**
>
> Wenn Sie das Format nicht angeben, wird der `iceberg.file-format` Wert verwendet, den Sie im vorherigen Abschnitt konfiguriert haben.

Verwenden Sie den `INSERT INTO` Befehl, um Daten einzufügen. Hier ein Beispiel:

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

Aus Iceberg-Tabellen lesen

Sie können den aktuellen Status Ihrer Iceberg-Tabelle mithilfe einer `SELECT` Anweisung wie folgt ablesen:

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

Daten in Iceberg-Tabellen importieren

Sie können einen Upsert-Vorgang ausführen (gleichzeitig neue Datensätze einfügen und bestehende aktualisieren), indem Sie die Anweisung verwenden. `MERGE INTO` Hier ein Beispiel:

```
MERGE INTO iceberg.iceberg_db.iceberg_table target
USING (
      VALUES
```

```
        (1001, 'John Updated', 'Boston'),       -- Update existing user
        (1002, 'Mary Updated', 'Seattle'),      -- Update existing user
        (1011, 'Martha', 'Portland'),           -- Insert new user
        (1012, 'Paulo', 'Austin')               -- Insert new user
    ) AS source (userid, firstname, city)
   ON target.userid = source.userid
 WHEN MATCHED THEN
      UPDATE SET
        firstname = source.firstname,
        city = source.city
 WHEN NOT MATCHED THEN
      INSERT (userid, firstname, city)
      VALUES (source.userid, source.firstname, source.city);
```

Löschen von Datensätzen aus Iceberg-Tabellen

Um Daten aus einer Iceberg-Tabelle zu löschen, verwenden Sie den `DELETE FROM` Ausdruck und geben Sie einen Filter an, der den zu löschenden Zeilen entspricht. Hier ein Beispiel:

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

Iceberg-Tabellen-Metadaten abfragen

Iceberg bietet Zugriff auf seine Metadaten über SQL. Sie können auf die Metadaten für jede beliebige Tabelle (`<table_name>`) zugreifen, indem Sie den Namespace abfragen. `"<table_name>.$<metadata_table>"` Eine vollständige Liste der Metadatentabellen finden Sie in der Iceberg-Dokumentation unter Tabellen überprüfen.

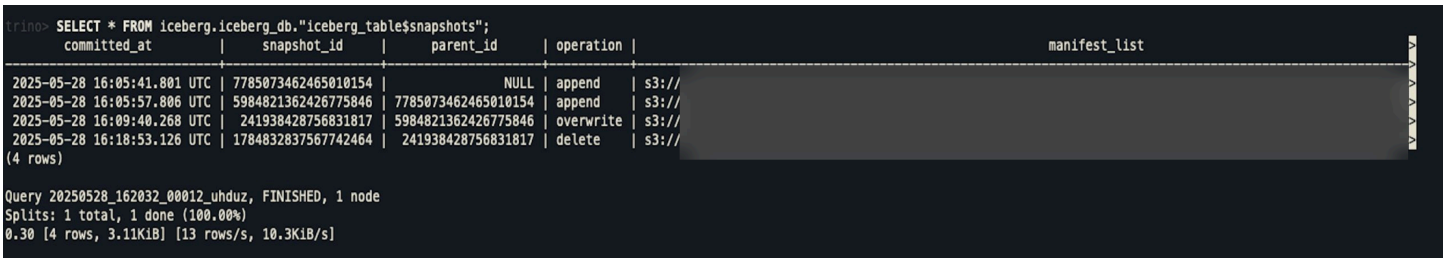
Hier ist ein Beispiel für eine Liste von Abfragen zur Untersuchung von Iceberg-Metadaten:

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

Beispielsweise würde diese Abfrage:

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

liefert die Ausgabe:



Zeitreisen nutzen

Jeder Schreibvorgang (Einfügen, Aktualisieren, Hochsetzen oder Löschen) in einer Iceberg-Tabelle erstellt einen neuen Snapshot. Sie können diese Snapshots dann für Zeitreisen verwenden, um in die Vergangenheit zu reisen und den Status einer Tabelle in der Vergangenheit zu überprüfen.

Die folgende Zeitreise-Abfrage zeigt den Status einer Tabelle auf der Grundlage eines bestimmten Status an: `snapshot_id`

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

Die folgende Zeitreiseabfrage zeigt den Status einer Tabelle auf der Grundlage eines bestimmten Zeitstempels an:

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

Überlegungen zur Verwendung von Iceberg mit Trino

Die Schreiboperationen von Trino auf Iceberg-Tabellen folgen dem merge-on-readDesign, sodass sie positionsabhängige Löschdateien erstellen, anstatt ganze Datendateien neu zu schreiben, die von Aktualisierungen oder Löschungen betroffen sind. Wenn Sie diesen copy-on-write Ansatz verwenden möchten, sollten Sie die Verwendung von Spark für Schreiboperationen in Betracht ziehen.

Arbeiten mit Iceberg-Tabellen mithilfe von Amazon Data Firehose

Amazon Data Firehose ist ein serverloser Service ohne Code für die Bereitstellung von Datenströmen aus über 20 Quellen wie AWS WAF Logs, Amazon CloudWatch Logs, Amazon Kinesis Data Streams und Amazon Managed Streaming for Apache Kafka (Amazon MSK) an Ziele wie Amazon S3, Amazon Redshift, Snowflake und Splunk. AWS IoT

Sie können Firehose verwenden, um Streaming-Daten direkt an Apache Iceberg-Tabellen in Amazon S3 zu übertragen. Mit Firehose können Sie Datensätze aus einem einzelnen Stream in verschiedene Apache Iceberg-Tabellen weiterleiten und automatisch Einfüge-, Aktualisierungs- und Löschvorgänge auf Datensätze in den Tabellen anwenden. Firehose garantiert die exakte Einmallieferung an die Tische von Iceberg. Diese Funktion erfordert die Verwendung von. AWS Glue Data Catalog

Firehose kann Streaming-Daten auch direkt an Amazon S3-Tabellen liefern. Diese Tabellen bieten Speicher, der für umfangreiche Analyse-Workloads optimiert ist, und enthalten Funktionen, die die Abfrageleistung kontinuierlich verbessern und die Speicherkosten für tabellarische Daten senken.

Informationen zum Einrichten eines Firehose-Streams zur Bereitstellung von Daten an Apache Iceberg-Tabellen finden Sie unter Einrichten des Firehose-Streams in der Firehose-Dokumentation oder im Blogbeitrag Streamen von Echtzeitdaten in Apache Iceberg-Tabellen in Amazon S3 mit Amazon Data Firehose.

Arbeiten mit Iceberg-Tabellen mithilfe von Athena SQL

Amazon Athena bietet integrierte Unterstützung für Apache Iceberg und erfordert keine zusätzlichen Schritte oder Konfigurationen. Dieser Abschnitt bietet einen detaillierten Überblick über die unterstützten Funktionen und allgemeine Anleitungen zur Verwendung von Athena für die Interaktion mit Iceberg-Tabellen.

Versions- und Funktionskompatibilität

Unterstützung der Iceberg-Tabellenspezifikationen

Die Apache Iceberg-Tabellenspezifikation spezifiziert, wie sich Iceberg-Tabellen verhalten sollen. Athena unterstützt das Tabellenformat Version 2, sodass jede Iceberg-Tabelle, die Sie mit der Konsole, der CLI oder dem SDK erstellen, grundsätzlich diese Version verwendet.

Wenn Sie eine Iceberg-Tabelle verwenden, die mit einer anderen Engine wie Apache Spark auf Amazon EMR oder erstellt wurde AWS Glue, stellen Sie sicher, dass Sie die Tabellenformatversion mithilfe der Tabelleneigenschaften festlegen. Als Referenz finden Sie den Abschnitt Erstellen und Schreiben von Iceberg-Tabellen weiter oben in diesem Handbuch.

Unterstützung für Iceberg-Funktionen

Sie können Athena verwenden, um aus Eisberg-Tabellen zu lesen und in sie zu schreiben. Wenn Sie Daten mithilfe der `DELETE FROM` Anweisungen`UPDATE`, und ändern`MERGE INTO`, unterstützt Athena nur merge-on-read den Modus. Diese Eigenschaft kann nicht geändert werden. Um Daten mit zu aktualisieren oder zu löschen copy-on-write, müssen Sie andere Engines wie Apache Spark auf Amazon EMR oder AWS Glue verwenden. In der folgenden Tabelle wird die Unterstützung der Iceberg-Funktionen in Athena zusammengefasst.

		DDL-Unterstützung		DML-Unterstützung		AWS Lake Formation aus Sicherheitsgründen (optional)
	Tabellenformat	Tabelle erstellen	Schemaentwicklung	Lesen von Daten	Daten schreiben	Zugriffskontrolle für Zeilen und Spalten
Amazon Athena	Version 2	✓	✓	✓	X Copy-on-write	✓
					✓ Merge-on-read	✓

> **Note**
>
> - Athena unterstützt keine inkrementellen Abfragen.
> - In Athena verwenden Aktualisierungs-, Lösch- und Zusammenführungsoperationen immer standardmäßig Merge on Read (MoR), unabhängig von den Copy-on-Write-Einstellungen (CoW) in den Tabelleneigenschaften, da CoW nicht unterstützt wird.

Mit Iceberg-Tabellen arbeiten

Einen schnellen Einstieg in die Verwendung von Iceberg in Athena finden Sie im Abschnitt Erste Schritte mit Iceberg-Tabellen in Athena SQL weiter oben in diesem Handbuch.

In der folgenden Tabelle sind Einschränkungen und Empfehlungen aufgeführt.

Szenario	Einschränkung	Empfehlung
Tabelle DDL-Generierung	Iceberg-Tabellen, die mit anderen Engines erstellt wurden, können Eigenschaften haben, die in Athena nicht verfügbar sind. Für diese Tabellen ist es nicht möglich, die DDL zu generieren.	Verwenden Sie die entsprechende Anweisung in der Engine, die die Tabelle erstellt hat (z. B. die `SHOW CREATE TABLE` Anweisung für Spark).
Zufällige Amazon S3 S3-Präfixe in Objekten, die in eine Iceberg-Tabelle geschrieben werden	In Iceberg-Tabellen, die mit Athena erstellt wurden, ist die `write.object-storage.enabled` Eigenschaft standardmäßig aktiviert.	Um dieses Verhalten zu deaktivieren und die volle Kontrolle über die Eigenschaften von Iceberg-Tabellen zu erlangen, erstellen Sie eine Iceberg-Tabelle mit einer anderen Engine wie Spark on Amazon EMR oder. AWS Glue
Inkrementelle Abfragen	Wird derzeit in Athena nicht unterstützt.	Um inkrementelle Abfragen zu verwenden, um inkrementelle Datenerfassungspipelines zu aktivieren, verwenden Sie Spark auf Amazon EMR oder. AWS Glue

Arbeiten mit Iceberg-Tabellen mithilfe von PyIceberg

In diesem Abschnitt wird erklärt, wie Sie mit Iceberg-Tabellen interagieren können, indem Sie PyIceberg Bei den bereitgestellten Beispielen handelt es sich um Standardcode, den Sie auf Amazon Linux EC2 2023-Instances, AWS Lambda-Funktionen oder einer beliebigen Python-Umgebung mit ordnungsgemäß konfigurierten AWS Anmeldeinformationen ausführen können.

Voraussetzungen

> **Note**
>
> In diesen Beispielen wird 1.9.1 verwendetPyIceberg .

Um damit arbeiten zu können PyIceberg, benötigen Sie PyIceberg und AWS SDK für Python (Boto3) haben es installiert. Hier ist ein Beispiel dafür, wie Sie eine virtuelle Python-Umgebung einrichten können, um mit PyIceberg und zu arbeiten AWS Glue Data Catalog:

1. Laden Sie es mit PyIcebergdem Pip-Python-Paketinstallationsprogramm herunter. Sie benötigen außerdem Boto3, um damit zu interagieren. AWS-Services Mit den folgenden Befehlen können Sie eine lokale virtuelle Python-Umgebung zum Testen konfigurieren:

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

2. Führen Sie auspython, um die Python-Shell zu öffnen und die Befehle zu testen.

Verbindung zum Datenkatalog herstellen

Um mit der Arbeit mit Iceberg-Tabellen in zu beginnen AWS Glue, müssen Sie zunächst eine Verbindung mit dem AWS Glue Data Catalog herstellen.

Die `load_catalog` Funktion initialisiert eine Verbindung zum Datenkatalog, indem sie ein Katalogobjekt erstellt, das als primäre Schnittstelle für alle Iceberg-Operationen dient:

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

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

Datenbanken auflisten und erstellen

Verwenden Sie die `list_namespaces` Funktion, um bestehende Datenbanken aufzulisten:

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

Verwenden Sie die `create_namespace` Funktion, um eine neue Datenbank zu erstellen:

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

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

Iceberg-Tabellen erstellen und schreiben

Unpartitionierte Tabellen

Hier ist ein Beispiel für die Erstellung einer unpartitionierten Iceberg-Tabelle mithilfe der Funktion: `create_table`

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

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

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

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

Sie können die `list_tables` Funktion verwenden, um die Liste der Tabellen in einer Datenbank zu überprüfen:

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

Sie können die append Funktion und PyArrow zum Einfügen von Daten in eine Iceberg-Tabelle verwenden:

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

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

Partitionierte Tabellen

Hier ist ein Beispiel für die Erstellung einer partitionierten Iceberg-Tabelle mit versteckter Partitionierung mithilfe der Funktion und: `create_table` `PartitionSpec`

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

```
    DoubleType,
    TimestampType,
)

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

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

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

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

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

Sie können Daten in eine partitionierte Tabelle genauso einfügen wie in eine unpartitionierte Tabelle. Die Partitionierung wird automatisch durchgeführt.

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

```
    pa.field("drone_id", pa.string(), nullable=False),
    pa.field("lat", pa.float64()),
    pa.field("lon", pa.float64()),
    pa.field("height", pa.float32()),
])

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

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

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

Lesen von Daten

Sie können die Pylceberg `scan` Funktion verwenden, um Daten aus Ihren Iceberg-Tabellen zu lesen. Sie können Zeilen filtern, bestimmte Spalten auswählen und die Anzahl der zurückgegebenen Datensätze einschränken.

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

print(scan_df)
```

Löschen von Daten

Mit Pylceberg `delete` dieser Funktion können Sie Datensätze aus Ihrer Tabelle entfernen, indem Sie Folgendes verwenden`delete_filter`:

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

Zugreifen auf Metadaten

Pylceberg bietet mehrere Funktionen für den Zugriff auf Tabellenmetadaten. So können Sie Informationen zu Tabellenschnappschüssen anzeigen:

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

#Current snapshot
table.current_snapshot()

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

Eine ausführliche Liste der verfügbaren Metadaten finden Sie im Abschnitt mit der Referenz zum Metadatencode der PyIceberg Dokumentation.

Zeitreisen nutzen

Sie können Tabellenschnappschüsse für Zeitreisen verwenden, um auf frühere Status Ihrer Tabelle zuzugreifen. So zeigen Sie den Tabellenstatus vor der letzten Operation an:

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

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

print(time_travel_df)
```

Eine vollständige Liste der verfügbaren Funktionen finden Sie in der PyIceberg Python-API-Dokumentation.

Arbeiten mit der Iceberg-Tabellenformatspezifikation Version 3

Die neueste Version der Apache Iceberg-Tabellenformatspezifikation ist Version 3. Diese Version bietet erweiterte Funktionen für den Aufbau von Data Lakes im Petabyte-Bereich mit verbesserter Leistung und geringerem Betriebsaufwand. Sie behebt häufig auftretende Leistungsengpässe bei Version 2, insbesondere bei Batch-Updates und Compliance-Löschvorgängen.

AWS bietet Unterstützung für Löschvektoren und Zeilenabstammung, wie sie in der Iceberg-Spezifikation Version 3 definiert sind. Diese Funktionen sind mit Apache Spark unter folgenden Bedingungen verfügbar. AWS-Services

AWS-Service	Unterstützung für Version 3.
Amazon EMR für Apache Spark	Amazon EMR Version 7.12 und höher
AWS Glue	Ja
AWS Glue: Iceberg REST API, Tabellenverwaltung	Ja
Amazon SageMaker Unified Studio-Notizbücher	Ja
Amazon S3 S3-Tabellen: Iceberg-REST-API, Tabellenverwaltung	Ja
Amazon Athena (Trino)	Nein

Die wichtigsten Funktionen von Version 3

Löschvektoren ersetzen die positionsbezogenen Löschdateien, die in Version 2 verwendet wurden, durch ein effizientes Binärformat, das als Puffin-Dateien gespeichert wurde. Dadurch entfällt der Schreibaufwand bei zufälligen Batch-Updates und Löschungen im Einklang mit der Datenschutz-Grundverordnung (DSGVO). Außerdem wird der Aufwand für die Pflege aktueller Daten erheblich reduziert. Organizations, die häufig Updates verarbeiten, werden durch weniger kleine Dateien sofortige Verbesserungen der Schreibleistung und geringere Speicherkosten feststellen.

Die Zeilenherkunft ermöglicht eine präzise Nachverfolgung von Änderungen auf Zeilenebene. Ihre nachgelagerten Systeme können Änderungen inkrementell verarbeiten, wodurch Datenpipelines beschleunigt und die Rechenkosten für Workflows zur Erfassung von Änderungsdaten (CDC) gesenkt werden. Diese integrierte Funktion macht benutzerdefinierte Implementierungen zur Änderungsverfolgung überflüssig.

Versionskompatibilität

Version 3 behält die Abwärtskompatibilität mit den Tabellen der Version 2 bei. AWS-Services unterstützen Tabellen der Versionen 2 und 3 gleichzeitig, sodass Sie:

- Führen Sie Abfragen für Tabellen der Versionen 2 und 3 aus.
- Führen Sie ein Upgrade vorhandener Tabellen der Version 2 auf Version 3 durch, ohne dass Daten neu geschrieben werden müssen.
- Führen Sie Zeitreiseabfragen aus, die Snapshots von Version 2 und Version 3 umfassen.
- Verwenden Sie Schemaentwicklung und versteckte Partitionierung für alle Tabellenversionen.

Erste Schritte mit Version 3

Voraussetzungen

Bevor Sie mit Tabellen der Version 3 arbeiten, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Und AWS-Konto mit entsprechenden AWS Identity and Access Management (IAM-) Berechtigungen.
- Zugriff auf einen oder mehrere AWS Analysedienste (Amazon EMR AWS Glue, Amazon SageMaker Unified Studio-Notebooks oder Amazon S3 S3-Tabellen).
- Ein S3-Bucket zum Speichern von Tabellendaten und Metadaten.
- Ein Tabellen-Bucket für die ersten Schritte mit Amazon S3 S3-Tabellen oder ein Allzweck-S3-Bucket, wenn Sie Ihre eigene Iceberg-Infrastruktur aufbauen.
- Ein konfigurierter Katalog. AWS Glue

Tabellen der Version 3 werden erstellt

Erstellen neuer Tabellen

Um eine neue Iceberg-Tabelle der Version 3 zu erstellen, setzen Sie die `format-version` Tabelleneigenschaft auf 3.

Verwenden von Spark SQL:

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

Aktualisierung der Tabellen von Version 2 auf Version 3

Sie können bestehende Tabellen der Version 2 atomar auf Version 3 aktualisieren, ohne Daten neu schreiben zu müssen.

Verwenden von Spark SQL:

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

> **Important**
>
> Version 3 ist ein unidirektionales Upgrade. Nachdem eine Tabelle von Version 2 auf Version 3 aktualisiert wurde, kann sie nicht mit Standardoperationen wieder auf Version 2 herabgestuft werden.

Was passiert während des Upgrades:

- Ein neuer Metadaten-Snapshot wird atomar erstellt.
- Bestehende Parquet-Datendateien werden wiederverwendet.
- Felder für die Zeilenherkunft werden zu den Metadaten der Tabelle hinzugefügt.

Nach dem Upgrade:

- Bei der nächsten Komprimierung werden die gelöschten Dateien der alten Version 2 entfernt.
- Bei neuen Änderungen werden die Löschvektordateien der Version 3 verwendet.

Bei der Aktualisierung werden keine historischen Daten von Datensätzen zur Nachverfolgung von Änderungen der Zeilenabstammung aufgefüllt.

Löschvektoren werden aktiviert

Um die Vorteile von Löschvektoren für Aktualisierungen, Löschungen und Zusammenführungen zu nutzen, konfigurieren Sie Ihren Schreibmodus.

Verwenden von Spark SQL:

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

Diese Einstellungen stellen sicher, dass bei Aktualisierungs-, Lösch- und Zusammenführungsvorgängen Löschvektordateien erstellt werden, anstatt ganze Datendateien neu zu schreiben.

Verwendung der Zeilenabstammung für die Nachverfolgung von Änderungen

Version 3 fügt automatisch Metadatenfelder für die Zeilenherkunft hinzu, um Änderungen nachzuverfolgen.

Verwenden von Spark SQL:

```
# Query with parameter value provided
```

```
last_processed_sequence = 47

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

Das `_row_id` Feld identifiziert jede Zeile eindeutig und `_last_updated_sequence_number` verfolgt, wann die Zeile zuletzt geändert wurde. Verwenden Sie diese Felder, um:

- Identifizieren Sie geänderte Zeilen für die inkrementelle Verarbeitung.
- Verfolgen Sie die Datenherkunft aus Gründen der Einhaltung gesetzlicher Vorschriften.
- Optimieren Sie CDC-Pipelines.
- Reduzieren Sie die Rechenkosten, indem Sie nur Änderungen verarbeiten.

Bewährte Methoden für Version 3

Wann sollte Version 3 verwendet werden

Erwägen Sie, auf Version 3 zu aktualisieren oder mit Version 3 zu beginnen, wenn:

- Sie führen häufig Batch-Updates oder Löschungen durch.
- Sie müssen die Anforderungen der DSGVO oder der Einhaltung gesetzlicher Vorschriften zur Löschung erfüllen.
- Ihre Workloads beinhalten häufig auftretende Störungen.
- Sie benötigen effiziente CDC-Workflows.
- Sie möchten die Speicherkosten für kleine Dateien reduzieren.
- Sie benötigen bessere Funktionen zur Nachverfolgung von Änderungen.

Optimierung der Schreibleistung

- Aktivieren Sie Löschvektoren für aktualisierungsintensive Workloads:

```
SET TBLPROPERTIES (
```

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

- Konfigurieren Sie die entsprechenden Dateigrößen:

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

Optimierung der Leseleistung

- Verwenden Sie die Zeilenabstammung für die inkrementelle Verarbeitung.
- Verwenden Sie Zeitreisen, um auf historische Daten zuzugreifen, ohne sie zu kopieren.
- Aktivieren Sie die Erfassung von Statistiken für eine bessere Abfrageplanung.

Migrationsstrategie

Beachten Sie bei der Migration von Version 2 zu Version 3 die folgenden bewährten Methoden:

- Testen Sie zunächst in einer Umgebung außerhalb der Produktionsumgebung, um den Upgrade-Prozess und die Leistung zu überprüfen.
- Führen Sie das Upgrade in Zeiten geringer Aktivität durch, um die Auswirkungen auf den gleichzeitigen Betrieb zu minimieren.
- Überwachen Sie die anfängliche Leistung und verfolgen Sie die Kennzahlen nach dem Upgrade.
- Führen Sie die Komprimierung durch, um gelöschte Dateien nach dem Upgrade zu konsolidieren.
- Aktualisieren Sie Ihre Teamdokumentation, sodass sie die Funktionen von Version 3 wiedergibt.

Erwägungen zur Kompatibilität

- Engine-Versionen — Stellen Sie sicher, dass alle Engines, die auf die Tabelle zugreifen, Version 3 unterstützen.
- Tools von Drittanbietern — Vergewissern Sie sich vor dem Upgrade, dass Ihr Tool mit Version 3 kompatibel ist.

- Backup-Strategie — Testen Sie Snapshot-basierte Wiederherstellungsverfahren.
- Überwachung — Aktualisieren Sie die Überwachungs-Dashboards für Version 3-spezifische Messwerte.

Fehlerbehebung

Häufige Probleme

Fehler: „Formatversion 3 wird nicht unterstützt“

- Stellen Sie sicher, dass Ihre Engine-Version Version 3 unterstützt. Einzelheiten finden Sie in der Tabelle am Anfang dieses Abschnitts.
- Überprüfen Sie die Katalogkompatibilität.
- Stellen Sie sicher, dass Sie die neuesten Versionen von verwenden AWS-Services.

Leistungseinbußen nach dem Upgrade

- Stellen Sie sicher, dass keine Verdichtungsfehler vorliegen. Weitere Informationen finden Sie unter Protokollierung und Überwachung für S3-Tabellen in der Amazon S3 S3-Dokumentation.
- Vergewissern Sie sich, dass Löschvektoren aktiviert sind. Die folgenden Eigenschaften sollten festgelegt werden:

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

  Sie können die Tabelleneigenschaften mit dem folgenden Code überprüfen:

```
DESCRIBE FORMATTED myns.orders_v3
```

- Überprüfen Sie Ihre Partitionsstrategie. Eine übermäßige Partitionierung kann zu kleinen Dateien führen. Führen Sie die folgende Abfrage aus, um die durchschnittliche Dateigröße für Ihre Tabelle zu ermitteln:

```
SELECT avg(file_size_in_bytes) as avg_file_size_bytes
```

```
FROM myns.orders_v3.files
```

Inkompatibilität mit Tools von Drittanbietern

- Stellen Sie sicher, dass das Tool die Version 3-Spezifikation unterstützt.
- Erwägen Sie, die Tabellen der Version 2 für Tools zu verwalten, die nicht unterstützt werden.
- Erkundigen Sie sich beim Tool-Anbieter nach dem Zeitplan für den Support für Version 3.

Hilfe erhalten

- Bei AWS-Service spezifischen Problemen wenden Sie sich an AWS Support.
- Um Hilfe von der Iceberg-Community zu erhalten, verwende den Iceberg-Slack-Kanal.
- Informationen zur Verwendung AWS-Services zur Verwaltung deiner Analytics-Workloads findest du unter Analytics on. AWS

Preisgestaltung

- Preise für Datenverarbeitungs- und Speicherlösungen von Amazon EMR
- SageMakerAmazon-Preisgestaltung
- AWS Glue Auftragsausführung und Preisgestaltung für den Datenkatalog
- Preise für Speicher und Anfragen für S3-Tabellen

Verfügbarkeit

Die Unterstützung der Iceberg-Tabellenformatspezifikation Version 3 ist überall verfügbar, AWS-Regionen wo Amazon EMR-, AWS Glue AWS Glue Data Catalog, und S3-Tabellen eingesetzt werden. Informationen zur Verfügbarkeit in den einzelnen Regionen finden Sie unter AWS-Services Nach Regionen.

Weitere Ressourcen

- Dokumentation zu Apache Iceberg
- Apache Iceberg-Tabellenspezifikation

- Anleitung für die Migration von Tabellendaten von Amazon S3 zu S3-Tabellen
- Tutorial: Erste Schritte mit S3-Tabellen

Migration vorhandener Tabellen nach Iceberg

Dieser Abschnitt konzentriert sich auf die Migration Ihrer vorhandenen Tabellen im Hive-Stil in das Iceberg-Format. Er gilt für Tabellen, die traditionelle Hive-kompatible Formate wie Apache Parquet oder Apache ORC verwenden. Diese Informationen gelten nicht für Tabellen, die bereits moderne Tabellenformate wie Linux Foundation Delta Lake oder Apache Hudi verwenden.

Um Ihre aktuellen Tabellen im Hive-Stil in das Iceberg-Format zu migrieren, können Sie entweder die direkte oder die vollständige Datenmigration verwenden:

- Bei der direkten Migration werden die Metadatendateien von Iceberg zusätzlich zu den vorhandenen Datendateien generiert.
- Bei der vollständigen Datenmigration wird die Iceberg-Metadatenebene erstellt und außerdem vorhandene Datendateien aus der Originaltabelle in die neue Iceberg-Tabelle umgeschrieben.

Die folgenden Abschnitte bieten einen detaillierten Überblick über die einzelnen Migrationsmethoden, einschließlich step-by-step Anweisungen und Überlegungen zur Implementierung. Weitere Informationen zu diesen Migrationsstrategien finden Sie im Abschnitt Tabellenmigration der Iceberg-Dokumentation.

Nachdem Sie sich mit den Einzelheiten der Methoden zur direkten und vollständigen Datenmigration befasst haben, lesen Sie sich die folgenden beiden wichtigen Abschnitte durch, die Ihnen bei der Entscheidungsfindung helfen sollen:

- Bei der Auswahl einer Migrationsstrategie finden Sie eine Reihe von Fragen und Szenarien, anhand derer Sie anhand Ihrer spezifischen Anforderungen und Anwendungsfälle den am besten geeigneten Migrationsansatz ermitteln können.
- Die Zusammenfassung der Migrationsoptionen bietet eine umfassende Tabelle, in der die wichtigsten Merkmale und Überlegungen der verschiedenen Migrationsoptionen verglichen werden. Diese Tabelle dient als Kurzanleitung und bietet einen Funktionsvergleich, der Ihnen hilft, die technischen Kompromisse zwischen den Methoden zu verstehen.

Direkte Migration

Durch die direkte Migration müssen Sie nicht mehr all Ihre Datendateien neu schreiben. Stattdessen werden Iceberg-Metadatendateien generiert und mit Ihren vorhandenen Datendateien verknüpft.

Diese Methode ist in der Regel schneller und kostengünstiger, insbesondere bei großen Datensätzen oder Tabellen mit kompatiblen Dateiformaten wie Parquet, Avro und ORC.

> **Note**
>
> Die direkte Migration kann bei der Migration zu Amazon S3-Tabellen nicht verwendet werden.

Iceberg bietet zwei Hauptoptionen für die Implementierung der In-Place-Migration:

- Verwenden des Snapshot-Verfahrens, um eine neue Iceberg-Tabelle zu erstellen und dabei die Quelltabelle unverändert zu lassen. Weitere Informationen finden Sie unter Snapshot-Tabelle in der Iceberg-Dokumentation.
- Verwenden des Migrationsverfahrens, um eine neue Iceberg-Tabelle als Ersatz für die Quelltabelle zu erstellen. Weitere Informationen finden Sie unter Tabelle migrieren in der Iceberg-Dokumentation. Dieses Verfahren funktioniert zwar mit Hive Metastore (HMS), ist aber derzeit nicht kompatibel mit dem. AWS Glue Data Catalog Das Verfahren zur Replikation der Tabellenmigration im AWS Glue Data Catalog Abschnitt weiter unten in diesem Handbuch bietet eine Problemumgehung, um mit dem Datenkatalog ein ähnliches Ergebnis zu erzielen.

Nachdem Sie die direkte Migration mit einer der Optionen `snapshot` oder durchgeführt haben`migrate`, bleiben einige Datendateien möglicherweise nicht migriert. Dies ist in der Regel der Fall, wenn Autoren während oder nach der Migration weiterhin in die Quelltabelle schreiben. Um diese verbleibenden Dateien in Ihre Iceberg-Tabelle zu integrieren, können Sie die add_files-Prozedur verwenden. Weitere Informationen finden Sie in der Iceberg-Dokumentation unter Dateien hinzufügen.

Nehmen wir an, Sie haben eine Parquet-basierte `products` Tabelle, die in Athena wie folgt erstellt und gefüllt wurde:

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

INSERT INTO mydb.products
```

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

In den folgenden Abschnitten wird erklärt, wie Sie die `migrate` Verfahren `snapshot` und für diese Tabelle verwenden können.

Option 1: Snapshot-Verfahren

Das `snapshot` Verfahren erstellt eine neue Iceberg-Tabelle, die einen anderen Namen hat, aber das Schema und die Partitionierung der Quelltabelle repliziert. Bei diesem Vorgang bleibt die Quelltabelle sowohl während als auch nach der Aktion vollständig unverändert. Dabei wird effektiv eine einfache Kopie der Tabelle erstellt, was besonders nützlich ist, um Szenarien zu testen oder Daten zu untersuchen, ohne das Risiko einzugehen, Änderungen an der ursprünglichen Datenquelle zu riskieren. Dieser Ansatz ermöglicht eine Übergangsphase, in der sowohl die Originaltabelle als auch die Iceberg-Tabelle verfügbar bleiben (siehe die Hinweise am Ende dieses Abschnitts). Wenn die Tests abgeschlossen sind, können Sie Ihre neue Iceberg-Tabelle in die Produktionsumgebung überführen, indem Sie alle Autoren und Leser auf die neue Tabelle umstellen.

Sie können das `snapshot` Verfahren ausführen, indem Sie Spark in jedem Amazon EMR-Bereitstellungsmodell verwenden (z. B. Amazon EMR auf EC2, Amazon EMR auf EKS, EMR Serverless) und. AWS Glue

Gehen Sie wie folgt vor, um die direkte Migration mit dem Spark-Verfahren zu testen: `snapshot`

1. Starten Sie eine Spark-Anwendung und konfigurieren Sie die Spark-Sitzung mit den folgenden Einstellungen:
   - `"spark.sql.extensions":"org.apache.iceberg.spark.extensions.IcebergSparkSessi`
   - `"spark.sql.catalog.spark_catalog":"org.apache.iceberg.spark.SparkSessionCatal`
   - `"spark.sql.catalog.spark_catalog.type":"glue"`
   - `"spark.hadoop.hive.metastore.client.factory.class":"com.amazonaws.glue.catalo`
2. Führen Sie das `snapshot` Verfahren aus, um eine neue Iceberg-Tabelle zu erstellen, die auf die ursprünglichen Tabellendatendateien verweist:

```
spark.sql(f"""
CALL system.snapshot(
```

```
source_table => 'mydb.products',
table => 'mydb.products_iceberg',
location => 's3://amzn-s3-demo-bucket/products_iceberg/'
)
"""
).show(truncate=False)
```

Der Ausgabedatenrahmen enthält die `imported_files_count` (die Anzahl der hinzugefügten Dateien).

3. Überprüfen Sie die neue Tabelle, indem Sie sie abfragen:

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

Hinweise

- Nachdem Sie die Prozedur ausgeführt haben, führt jede Änderung der Datendatei an der Quelltabelle dazu, dass die generierte Tabelle nicht mehr synchron ist. Neue Dateien, die Sie hinzufügen, sind in der Eisberg-Tabelle nicht sichtbar, und Dateien, die Sie entfernt haben, wirken sich auf die Abfragefunktionen in der Eisberg-Tabelle aus. Gehen Sie wie folgt vor, um Synchronisationsprobleme zu vermeiden:
  - Wenn die neue Iceberg-Tabelle für den Produktionseinsatz vorgesehen ist, beenden Sie alle Prozesse, die in die Originaltabelle schreiben, und leiten Sie sie in die neue Tabelle um.
  - Wenn Sie eine Übergangszeit benötigen oder wenn die neue Iceberg-Tabelle zu Testzwecken verwendet wird, finden Sie weiter unten in diesem Abschnitt Anleitungen zur Aufrechterhaltung der Tabellensynchronisierung unter Synchronisieren von Iceberg-Tabellen nach der In-Place-Migration.
- Wenn Sie das `snapshot` Verfahren verwenden, wird die `gc.enabled` Eigenschaft `false` in den Tabelleneigenschaften der erstellten Iceberg-Tabelle auf festgelegt. Diese Einstellung verbietet Aktionen wie `expire_snapshots`, `remove_orphan_files`, oder `DROP TABLE` mit der `PURGE` Option, durch die Datendateien physisch gelöscht würden. Lösch- oder Zusammenführungsvorgänge von Iceberg, die sich nicht direkt auf Quelldateien auswirken, sind weiterhin zulässig.

- Um Ihre neue Iceberg-Tabelle voll funktionsfähig zu machen, ohne Einschränkungen bei Aktionen, die Datendateien physisch löschen, können Sie die `gc.enabled` Tabelleneigenschaft auf ändern. `true` Diese Einstellung ermöglicht jedoch Aktionen, die sich auf Quelldatendateien auswirken, wodurch der Zugriff auf die Originaltabelle beeinträchtigt werden könnte. Ändern Sie die `gc.enabled` Eigenschaft daher nur, wenn Sie die Funktionalität der Originaltabelle nicht mehr beibehalten müssen. Beispiel:

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

Option 2: Migrationsverfahren

Das `migrate` Verfahren erstellt eine neue Iceberg-Tabelle, die denselben Namen, dasselbe Schema und dieselbe Partitionierung wie die Quelltabelle hat. Wenn diese Prozedur ausgeführt wird, sperrt sie die Quelltabelle und benennt sie um in `<table_name>_BACKUP_` (oder in einen benutzerdefinierten Namen, der durch den `backup_table_name` Parameter procedure angegeben wird).

> **Note**
>
> Wenn Sie den `drop_backup` Prozedurparameter auf setzen`true`, wird die Originaltabelle nicht als Backup aufbewahrt.

Folglich erfordert die `migrate` Tabellenprozedur, dass alle Änderungen, die sich auf die Quelltabelle auswirken, gestoppt werden, bevor die Aktion ausgeführt wird. Bevor Sie die `migrate` Prozedur ausführen:

- Stoppen Sie alle Writer, die mit der Quelltabelle interagieren.
- Ändern Sie Leser und Autoren, die Iceberg nicht nativ unterstützen, um die Iceberg-Unterstützung zu aktivieren.

Beispiel:

- Athena arbeitet weiterhin ohne Änderung.

- Spark benötigt:
  - Iceberg Java Archive (JAR) -Dateien, die in den Klassenpfad aufgenommen werden sollen (siehe die Abschnitte Arbeiten mit Iceberg in Amazon EMR und Arbeiten mit Iceberg in den Abschnitten weiter oben in AWS Glue diesem Handbuch).
  - Die folgenden Konfigurationen des Spark-Sitzungskatalogs (werden verwendet, `SparkSessionCatalog` um Iceberg-Unterstützung hinzuzufügen und gleichzeitig die integrierten Katalogfunktionen für Tabellen beizubehalten, die nicht zu Iceberg gehören):
    - `"spark.sql.extensions":"org.apache.iceberg.spark.extensions.IcebergSparkSes`
    - `"spark.sql.catalog.spark_catalog":"org.apache.iceberg.spark.SparkSessionCat`
    - `"spark.sql.catalog.spark_catalog.type":"glue"`
    - `"spark.hadoop.hive.metastore.client.factory.class":"com.amazonaws.glue.cata`

Nachdem Sie das Verfahren ausgeführt haben, können Sie Ihre Writer mit ihrer neuen Iceberg-Konfiguration neu starten.

Derzeit ist das `migrate` Verfahren nicht mit dem kompatibel AWS Glue Data Catalog, da der Datenkatalog den `RENAME` Vorgang nicht unterstützt. Daher empfehlen wir, dieses Verfahren nur zu verwenden, wenn Sie mit Hive Metastore arbeiten. Wenn Sie den Datenkatalog verwenden, finden Sie im nächsten Abschnitt einen alternativen Ansatz.

Sie können das `migrate` Verfahren für alle Amazon EMR-Bereitstellungsmodelle (Amazon EMR auf EC2, Amazon EMR auf EKS, EMR Serverless) und ausführen, erfordert jedoch eine konfigurierte Verbindung zu AWS Glue Hive Metastore. Amazon EMR on EC2 ist die empfohlene Wahl, da es eine integrierte Hive Metastore-Konfiguration bietet, die die Komplexität der Einrichtung minimiert.

Gehen Sie wie folgt vor, um die In-Place-Migration mit dem `migrate` Spark-Verfahren von einem Amazon EMR auf EC2-Cluster zu testen, der mit Hive Metastore konfiguriert ist:

1. Starten Sie eine Spark-Anwendung und konfigurieren Sie die Spark-Sitzung so, dass sie die Iceberg Hive-Katalogimplementierung verwendet. Wenn Sie beispielsweise die `pyspark` CLI verwenden:

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

2. Erstellen Sie eine `products` Tabelle in Hive Metastore. Dies ist die Quelltabelle, die bei einer typischen Migration bereits vorhanden ist.

   a. Erstellen Sie die `products` externe Hive-Tabelle in Hive Metastore, um auf die vorhandenen Daten in Amazon S3 zu verweisen:

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

   b. Fügen Sie die vorhandenen Partitionen mit dem folgenden Befehl hinzu: `MSCK REPAIR TABLE`

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

   c. Stellen Sie sicher, dass die Tabelle Daten enthält, indem Sie eine `SELECT` Abfrage ausführen:

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

   Beispielausgabe:

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

3. Verwenden Sie das Iceberg-Verfahren`migrate`:

```
df_res=spark.sql(f"""
CALL system.migrate(
table => 'default.products'
)
"""
)

df_res.show()
```

Der Ausgabedatenrahmen enthält `migrated_files_count` (die Anzahl der Dateien, die der Iceberg-Tabelle hinzugefügt wurden):

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

4. Vergewissern Sie sich, dass die Backup-Tabelle erstellt wurde:

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

Beispielausgabe:

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

5. Überprüfen Sie den Vorgang, indem Sie die Iceberg-Tabelle abfragen:

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

> **Hinweise**
>
> - Nachdem Sie die Prozedur ausgeführt haben, sind alle aktuellen Prozesse, die die Quelltabelle abfragen oder in sie schreiben, betroffen, wenn sie nicht ordnungsgemäß mit Iceberg-Unterstützung konfiguriert sind. Daher empfehlen wir, dass Sie die folgenden Schritte ausführen:
>
>   1. Stoppen Sie vor der Migration alle Prozesse mithilfe der Quelltabelle.
>   2. Führen Sie die Migration durch.
>   3. Reaktivieren Sie die Prozesse, indem Sie die richtigen Iceberg-Einstellungen verwenden.
>
> - Wenn Datendateien während des Migrationsprozesses geändert werden (neue Dateien werden hinzugefügt oder Dateien entfernt), gerät die generierte Tabelle nicht mehr synchron. Informationen zu Synchronisierungsoptionen finden Sie unter Synchronisieren von Iceberg-Tabellen nach der In-Place-Migration weiter unten in diesem Abschnitt.

Das Verfahren zur Tabellenmigration replizieren in AWS Glue Data Catalog

Sie können das Ergebnis des Migrationsvorgangs replizieren AWS Glue Data Catalog (Sicherung der Originaltabelle und Ersetzen durch eine Iceberg-Tabelle), indem Sie die folgenden Schritte ausführen:

1. Verwenden Sie das Snapshot-Verfahren, um eine neue Iceberg-Tabelle zu erstellen, die auf die Datendateien der Originaltabelle verweist.
2. Sichern Sie die Metadaten der ursprünglichen Tabelle im Datenkatalog:
   a. Verwenden Sie die GetTableAPI, um die Quelltabellendefinition abzurufen.
   b. Verwenden Sie die GetPartitionsAPI, um die Partitionsdefinition der Quelltabelle abzurufen.
   c. Verwenden Sie die CreateTableAPI, um eine Backup-Tabelle im Datenkatalog zu erstellen.
   d. Verwenden Sie die BatchCreatePartitionAPI CreatePartitionoder, um Partitionen in der Backup-Tabelle im Datenkatalog zu registrieren.
3. Ändern Sie die `gc.enabled` Iceberg-Tabelleneigenschaft auf`false`, um vollständige Tabellenoperationen zu aktivieren.
4. Entfernen Sie die ursprüngliche Tabelle.
5. Suchen Sie die Iceberg-Tabellenmetadaten-JSON-Datei im Metadatenordner des Stammverzeichnisses der Tabelle.
6. Registrieren Sie die neue Tabelle im Datenkatalog, indem Sie die Prozedur register_table mit dem ursprünglichen Tabellennamen und dem Speicherort der `metadata.json` Datei verwenden, die von der Prozedur erstellt wurde: `snapshot`

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

Synchronisieren von Iceberg-Tabellen nach der In-Place-Migration

Das `add_files` Verfahren bietet eine flexible Möglichkeit, vorhandene Daten in Iceberg-Tabellen zu integrieren. Insbesondere werden vorhandene Datendateien (wie Parquet-Dateien) registriert, indem auf ihre absoluten Pfade in der Metadatenebene von Iceberg verwiesen wird. Standardmäßig fügt das Verfahren Dateien aus allen Tabellenpartitionen zu einer Iceberg-Tabelle hinzu, Sie können jedoch selektiv Dateien aus bestimmten Partitionen hinzufügen. Dieser selektive Ansatz ist in mehreren Szenarien besonders nützlich:

- Wenn der Quelltabelle nach der ersten Migration neue Partitionen hinzugefügt werden.

- Wenn nach der ersten Migration Datendateien zu vorhandenen Partitionen hinzugefügt oder aus vorhandenen Partitionen entfernt werden. Um geänderte Partitionen erneut hinzuzufügen, muss die Partition jedoch zuerst gelöscht werden. Weitere Informationen dazu finden Sie weiter unten in diesem Abschnitt.

Im Folgenden finden Sie einige Überlegungen zur Verwendung des `add_file` Verfahrens, nachdem die direkte Migration (`snapshotodermigrate`) durchgeführt wurde, um die neue Iceberg-Tabelle mit den Quelldatendateien synchron zu halten:

- Wenn neue Daten zu neuen Partitionen in der Quelltabelle hinzugefügt werden, verwenden Sie das `add_files` Verfahren mit der `partition_filter` Option, diese Ergänzungen selektiv in die Iceberg-Tabelle zu integrieren:

```
spark.sql(f"""
CALL system.add_files(
source_table => 'mydb.products',
table => 'mydb.products_iceberg',
partition_filter => map('category', 'electronics')
).show(truncate=False)
```

  oder:

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

- Die `add_files` Prozedur sucht entweder in der gesamten Quelltabelle oder in bestimmten Partitionen nach Dateien, wenn Sie die `partition_filter` Option angeben, und versucht, alle gefundenen Dateien zur Iceberg-Tabelle hinzuzufügen. Standardmäßig ist die `check_duplicate_files` Prozedureigenschaft auf gesetzt`true`, wodurch verhindert wird, dass die Prozedur ausgeführt wird, wenn in der Iceberg-Tabelle bereits Dateien vorhanden sind. Dies ist wichtig, da es keine integrierte Option zum Überspringen zuvor hinzugefügter Dateien gibt und eine Deaktivierung `check_duplicate_files` dazu führt, dass Dateien zweimal hinzugefügt werden, wodurch Duplikate entstehen. Gehen Sie folgendermaßen vor, wenn der Quelltabelle neue Dateien hinzugefügt werden:

1. Verwenden Sie für neue Partitionen `add_files` with a, `partition_filter` um nur Dateien aus der neuen Partition zu importieren.
2. Löschen Sie bei vorhandenen Partitionen zuerst die Partition aus der Iceberg-Tabelle und führen Sie den Vorgang dann erneut `add_files` für diese Partition aus, wobei Sie den angeben. `partition_filter` Beispiel:

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

# Then on the source table, some new files were generated under the
 category='electronics' partition. Example:
spark.sql("""
INSERT INTO mydb.products
VALUES (1003, 'Tablet', 'electronics')
""")

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

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

> **Note**
>
> Jeder `add_files` Vorgang generiert einen neuen Iceberg-Tabellen-Snapshot mit angehängten Daten.

Auswahl der richtigen Strategie für die direkte Migration

Beachten Sie die Fragen in der folgenden Tabelle, um die beste Strategie für die direkte Migration auszuwählen.

Frage	Empfehlung	Erklärung
Möchten Sie schnell migrieren , ohne Daten neu zu schreiben , und gleichzeitig die Hive- und Iceberg-Tabellen für Tests oder schrittweise Umstellung zugänglich halten?	`snapshot`Verfahren gefolgt von Verfahren `add_files`	Gehen Sie `snapshot` wie folgt vor, um eine neue Iceberg-Tabelle zu erstellen , indem Sie das Schema klonen und auf Datendateien verweisen, ohne die Quelltabe lle zu ändern. Verwenden Sie das `add_files` Verfahren , um Partitionen einzubezi ehen, die nach der Migration hinzugefügt oder geändert wurden. Beachten Sie, dass beim erneuten Hinzufüge n geänderter Partitionen zunächst die Partition gelöscht werden muss.
Verwenden Sie Hive Metastore und möchten Sie Ihre Hive-Tabelle sofort durch eine Iceberg-Tabelle ersetzen, ohne die Daten neu zu schreiben?	`migrate`add_files Verfahren gefolgt von Verfahren	Gehen Sie `migrate` wie folgt vor, um eine Iceberg-Tabelle zu erstellen, die Quelltabelle zu sichern und die Originalt abelle durch die Iceberg-V ersion zu ersetzen.

Frage	Empfehlung	Erklärung
		Hinweis: Diese Option ist mit Hive Metastore kompatibel, aber nicht mit. AWS Glue Data Catalog Gehen Sie `add_files` wie folgt vor, um Partitionen zu integrieren, die nach der Migration hinzugefügt oder geändert wurden. Beachten Sie, dass beim erneuten Hinzufügen geänderter Partitionen zunächst die Partition gelöscht werden muss.

Frage	Empfehlung	Erklärung
Verwenden Sie Ihre Hive-Tabelle AWS Glue Data Catalog und möchten Sie diese sofort durch eine Iceberg-Tabelle ersetzen, ohne die Daten neu zu schreiben?	Anpassung des `migrate` Verfahrens, gefolgt vom Verfahren `add_files`	Verhalten der `migrate` Prozedur replizieren: 1. Wird verwendet`snapshot`, um eine Iceberg-Tabelle zu erstellen. 2. Sichern Sie die ursprüngl ichen Tabellenmetadaten mithilfe AWS Glue APIs von. 3. Aktivieren Sie `gc.enable d` die Iceberg-Tabellenei genschaften. 4. Entfernen Sie die ursprüngl iche Tabelle. 5. Wird verwendet`register_ table` , um einen neuen Tabelleneintrag mit dem ursprünglichen Namen zu erstellen. Hinweis: Diese Option erfordert die manuelle Bearbeitung von AWS Glue API-Aufrufen für die Metadatensicherung. Verwenden Sie das `add_files` Verfahren, um Partitionen zu integrier en, die nach der Migration hinzugefügt oder geändert wurden. Beachten Sie, dass beim erneuten Hinzufüge

Frage	Empfehlung	Erklärung
		n geänderter Partitionen zunächst die Partition gelöscht werden muss.

Vollständige Datenmigration

Bei der vollständigen Datenmigration werden sowohl die Datendateien als auch die Metadaten neu erstellt. Dieser Ansatz dauert länger und erfordert zusätzliche Rechenressourcen im Vergleich zur direkten Migration. Eine vollständige Datenmigration bietet jedoch erhebliche Möglichkeiten zur Verbesserung der Tabellenqualität und zur Optimierung der Datenspeicher- und Zugriffsmuster.

Während der vollständigen Datenmigration können Sie verschiedene nützliche Operationen durchführen, z. B. die Datenvalidierung zur Sicherstellung der Integrität und Richtigkeit, Schemaänderungen, um den aktuellen Anforderungen besser gerecht zu werden, und Anpassungen der Partitionsstrategie zur Verbesserung der Abfrageleistung. Sie können Daten auch neu sortieren, um allgemeine Zugriffsmuster zu optimieren, die versteckte Iceberg-Partitionierung implementieren, um die Abfrageeffizienz zu erhöhen, und bei Bedarf eine Dateiformatkonvertierung (z. B. von CSV nach Parquet) durchführen.

Diese Funktionen machen die vollständige Datenmigration ideal für die Umstellung auf das Iceberg-Format und für die umfassende Verfeinerung und Optimierung Ihrer Datenspeicherstrategie. Obwohl eine vollständige Datenmigration im Vorfeld mehr Zeit und Ressourcen erfordert, können die daraus resultierenden Verbesserungen der Datenqualität, Organisation und Abfrageleistung langfristige Vorteile bieten. Verwenden Sie eine der folgenden Optionen, um eine vollständige Datenmigration zu implementieren:

- Verwenden Sie die `CREATE TABLE ... AS SELECT` (CTAS) -Anweisung in Spark (auf Amazon EMR oder AWS Glue) oder in Athena. Sie können die Partitionsspezifikation und die Tabelleneigenschaften für die neue Iceberg-Tabelle mithilfe der UND-Klauseln festlegen. `PARTITIONED BY` `TBLPROPERTIES` Sie können das Schema und die Partitionierung für die neue Tabelle Ihren Bedürfnissen entsprechend ändern, anstatt sie von der Quelltabelle zu erben.
- Lesen Sie aus der Quelltabelle und schreiben Sie die Daten als neue Iceberg-Tabelle, indem Sie Spark auf Amazon EMR verwenden oder. AWS Glue Weitere Informationen finden Sie in der Iceberg-Dokumentation unter Tabelle erstellen.

Auswählen einer Migrationsstrategie

Bei der Umstellung auf das Iceberg-Format ist die Wahl zwischen direkter Migration und vollständiger Migration entscheidend. Beachten Sie die folgenden Fragen und Empfehlungen, um den für Ihre spezifischen Bedürfnisse am besten geeigneten Ansatz zu ermitteln:

Frage	Empfehlung
Was ist das Datendateiformat (z. B. CSV oder Apache Parquet)?	• Ziehen Sie eine direkte Migration in Betracht, wenn Ihr Tabellendateiformat Parquet, ORC oder Avro ist. • Verwenden Sie für andere Formate wie CSV, JSON usw. die vollständige Datenmigration.
Möchten Sie das Tabellens chema aktualisieren oder konsolidieren?	• Wenn Sie das Tabellenschema mithilfe der systemeigenen Funktionen von Iceberg weiterentwickeln möchten, sollten Sie eine direkte Migration in Betracht ziehen. Sie können beispiels weise Spalten nach der Migration umbenennen. (Das Schema kann in der Iceberg-Metadatenebene geändert werden.) • Wenn Sie ganze Spalten entfernen möchten, weil sie nicht mehr benötigt werden, empfehlen wir Ihnen, die vollständige Datenmigration zu verwenden.
Würde die Tabelle von einer Änderung der Partition sstrategie profitieren?	• Wenn der Partitionierungsansatz von Iceberg Ihren Anforderu ngen entspricht (z. B. werden neue Daten mithilfe des neuen Partitionslayouts gespeichert, während die vorhandenen Partitionen unverändert bleiben), sollten Sie eine direkte Migration in Betracht ziehen. • Wenn Sie versteckte Partitionen in Ihrer Tabelle verwenden möchten, sollten Sie eine vollständige Datenmigration in Betracht ziehen. Weitere Informationen zu versteckten Partitionen finden Sie im Abschnitt Bewährte Methoden.
Würde es für die Tabelle von Vorteil sein, die Strategie für die Sortierreihenfolge hinzuzufügen oder zu ändern?	• Um die Sortierreihenfolge Ihrer Daten hinzuzufügen oder zu ändern, muss der Datensatz neu geschrieben werden. In diesem Fall sollten Sie die vollständige Datenmigration in Betracht ziehen.

Frage	Empfehlung
	• Bei großen Tabellen, bei denen es unerschwinglich teuer ist, alle Tabellenpartitionen neu zu schreiben, sollten Sie die direkte Migration in Betracht ziehen und die Komprimierung (mit aktivierter Sortierung) für die Partitionen ausführen, auf die am häufigsten zugegriffen wird.
Enthält die Tabelle viele kleine Dateien?	• Um kleine Dateien zu größeren Dateien zusammenzuführen, muss der Datensatz neu geschrieben werden. In diesem Fall sollten Sie die vollständige Datenmigration in Betracht ziehen. • Bei großen Tabellen, bei denen es unerschwinglich teuer ist, alle Tabellenpartitionen neu zu schreiben, sollten Sie die direkte Migration in Betracht ziehen und die Komprimierung (mit aktivierter Sortierung) für die Partitionen ausführen, auf die am häufigsten zugegriffen wird.

Zusammenfassung der Migrationsoptionen

In dieser Tabelle sind die wichtigsten Merkmale und Überlegungen für jede Migrationsoption zusammengefasst.

Merkmal	Migration vor Ort Snapshot	Migration vor Ort migrate	Vollständige Datenmigration CTAS oder (TABELLE ERSTELLEN + EINFÜGEN)
Verbesse ungen des Datenlay uts im Rahmen des Migratior			

Merkmal	Migration vor Ort Snapshot	Migration vor Ort migrate	Vollständige Datenmigration CTAS oder (TABELLE ERSTELLEN + EINFÜGEN)
sprozesse			
• Daten neu sortier	 N	 N	 Ja
• Änderr Sie die Partitic ierung (z. B. um die verste e Iceber P artition erung zu verwer)	 N	 N	 Ja
• Änderr Sie das Tabell chema	 N	 N	 Ja

Merkmal	Migration vor Ort Snapshot	Migration vor Ort migrate	Vollständige Datenmigration CTAS oder (TABELLE ERSTELLEN + EINFÜGEN)
• Optimi n Sie die Dateig e	 N	 N	 Ja
• Überpı n Sie das Schem vorhar er Daten, bevor Sie die Daten hinzufi n	 N	 N	 Ja
Unterstü zte Dateiforı ate	Parquet, Avro, ORC	Parkett, Avro, ORC	Parkett, Avro, ORC, JSON, CSV

Merkmal	Migration vor Ort Snapshot	Migration vor Ort migrate	Vollständige Datenmigration CTAS oder (TABELLE ERSTELLEN + EINFÜGEN)
Ersatz der Quelltabelle durch eine Iceberg-Tabelle	 N (erstellt eine neue Tabelle, aber mit zusätzlichen Schritten können Sie die Quelltabelle ersetzen)	 Ja (erstellt eine Backup-Tabelle und ersetzt die Quelltabelle durch eine Iceberg-Tabelle)	 Nein (erstellt eine neue Tabelle)
Auswirkung auf die Quelltabelle			

Merkmal	Migration vor Ort Snapshot	Migration vor Ort migrate	Vollständige Datenmigration CTAS oder (TABELLE ERSTELLEN + EINFÜGEN)
• Operat en zum Lösch von Dateie in der Iceber T abelle (`expi` `apsho` Operat en, Lösch einer Tabell beim Lösch	Korrumpiert die Quelltabelle	Korrumpiert die Backup-Tabelle	Sicher, Quelle nicht betroffen
Auswirku g auf den Eisberg- Tisch			

Merkmal	Migration vor Ort Snapshot	Migration vor Ort migrate	Vollständige Datenmigration CTAS oder (TABELLE ERSTELLEN + EINFÜGEN)
• Auswirkung, wenn Quelltabellendaten entfernt werden	Korrumpiert die Iceberg-Tabelle	Korrumpiert die Eisberg-Tabelle	Keine Auswirkungen auf die Iceberg-Tabelle
• Auswirkung, wenn neue Dateien am Speicherort der Quelltabelle hinzugefügt werden	In der neuen Tabelle nicht sichtbar (Partition muss mit integriert werden`add_files`)	Auf der neuen Tabelle nicht sichtbar (Partition muss mit integriert werden`add_files`)	Auf der neuen Tabelle nicht sichtbar (Ich brauche `INSERT INTO` die neue Tabelle)
Kosten	Niedrig	Niedrig	Höher (vollständige Datenumschreibung)
Geschwindigkeit der Migration	Schnell	Schnell	Langsamer

Merkmal	Migration vor Ort Snapshot	Migration vor Ort migrate	Vollständige Datenmigration CTAS oder (TABELLE ERSTELLEN + EINFÜGEN)
Kann für die Migration zu Amazon S3 S3-Tabellen verwendet werden	 N	 N	 Ja
Erfordert manuelle DDL	 N (Schema und Partitionen werden aus der Quelltabelle kopiert)	 N (Schema und Partitionen werden aus der Quelltabelle kopiert)	Wenn Sie CTAS verwenden, müssen Sie nur die Partitionierung angeben
Beste Verwendung	Schnelle Migration ohne Umschreiben von Daten, sodass Hive und Iceberg für Tests oder schrittweise Umstellung side-by-side verwendet werden können.	Ersetzen einer Hive-Tabelle an Ort und Stelle, ohne Daten neu zu schreiben, wenn eine sofortige Umstellung akzeptabel ist.	Vollständige Iceberg-Optimierung mit Datenumschreibung. Ideal für die Neugestaltung von Partitionen oder Schemas oder für die Verbesserung von Layout und Leistung. Immer empfehlenswert, wenn möglich.

Bewährte Methoden zur Optimierung von Apache Iceberg-Workloads

Iceberg ist ein Tabellenformat, das entwickelt wurde, um das Data Lake-Management zu vereinfachen und die Workload-Leistung zu verbessern. In verschiedenen Anwendungsfällen können unterschiedliche Aspekte wie Kosten, Leseleistung, Schreibleistung oder Datenspeicherung priorisiert werden. Daher bietet Iceberg Konfigurationsoptionen, um diese Kompromisse zu bewältigen. Dieser Abschnitt bietet Einblicke in die Optimierung und Feinabstimmung Ihrer Iceberg-Workloads, um Ihre Anforderungen zu erfüllen.

Themen

Allgemeine bewährte Methoden

Unabhängig von Ihrem Anwendungsfall empfehlen wir Ihnen, diese allgemeinen bewährten Methoden zu befolgen AWS, wenn Sie Apache Iceberg auf verwenden.

- Verwenden Sie Version 2 des Iceberg-Formats.

  Athena verwendet standardmäßig Version 2 des Iceberg-Formats.

  Wenn Sie Spark auf Amazon EMR verwenden oder AWS Glue Iceberg-Tabellen erstellen, geben Sie die Formatversion an, wie in der Iceberg-Dokumentation beschrieben.
- Verwenden Sie das AWS Glue Data Catalog als Ihren Datenkatalog.

  Athena verwendet AWS Glue Data Catalog standardmäßig das.

  Wenn Sie Spark auf Amazon EMR verwenden oder AWS Glue mit Iceberg arbeiten, fügen Sie Ihrer Spark-Sitzung die folgende Konfiguration hinzu, um die zu verwenden. AWS Glue Data Catalog

Weitere Informationen finden Sie im Abschnitt Spark-Konfigurationen für Iceberg weiter AWS Glue oben in diesem Handbuch.

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

- Verwenden Sie den AWS Glue Data Catalog als Sperrmanager.

  Athena verwendet standardmäßig den AWS Glue Data Catalog AS-Lock-Manager für Iceberg-Tabellen.

  Wenn Sie Spark auf Amazon EMR verwenden oder AWS Glue mit Iceberg arbeiten, stellen Sie sicher, dass Sie Ihre Spark-Sitzungskonfiguration so konfigurieren, dass sie den AWS Glue Data Catalog AS-Lock-Manager verwendet. Weitere Informationen finden Sie unter Optimistic Locking in der Iceberg-Dokumentation.

- Verwenden Sie die Zstandard-Komprimierung (ZSTD).

  Der Standard-Komprimierungscodec von Iceberg ist gzip, der mithilfe der Tabelleneigenschaft geändert werden kann. `write.<file_type>.compression-codec` Athena verwendet ZSTD bereits als Standard-Komprimierungscodec für Iceberg-Tabellen.

  Im Allgemeinen empfehlen wir die Verwendung des ZSTD-Komprimierungscodecs, da er ein Gleichgewicht zwischen GZIP und Snappy herstellt und eine gute Leistung bietet, ohne die Komprimierungsrate zu beeinträchtigen. read/write Darüber hinaus können die Komprimierungsstufen an Ihre Bedürfnisse angepasst werden. Weitere Informationen finden Sie unter ZSTD-Komprimierungsstufen in Athena in der Athena-Dokumentation.

  Snappy bietet zwar insgesamt die beste Lese- und Schreibleistung, hat aber ein niedrigeres Komprimierungsverhältnis als GZIP und ZSTD. Wenn Sie der Leistung Priorität einräumen — auch wenn das bedeutet, größere Datenmengen in Amazon S3 zu speichern — ist Snappy möglicherweise die optimale Wahl.

Optimierung der Leseleistung

In diesem Abschnitt werden Tabelleneigenschaften beschrieben, die Sie unabhängig von der Engine anpassen können, um die Leseleistung zu optimieren.

Partitionierung

Wie bei Hive-Tabellen verwendet Iceberg Partitionen als primäre Indizierungsebene, um das Lesen unnötiger Metadaten- und Datendateien zu vermeiden. Spaltenstatistiken werden auch als sekundäre Indizierungsebene berücksichtigt, um die Abfrageplanung weiter zu verbessern, was zu einer besseren Gesamtausführungszeit führt.

Ihre Daten partitionieren

Um die Datenmenge zu reduzieren, die bei der Abfrage von Iceberg-Tabellen gescannt wird, wählen Sie eine ausgewogene Partitionsstrategie, die Ihren erwarteten Lesemustern entspricht:

- Identifizieren Sie Spalten, die häufig in Abfragen verwendet werden. Dies sind ideale Partitionierungskandidaten. Wenn Sie beispielsweise normalerweise Daten von einem bestimmten Tag abfragen, wäre ein natürliches Beispiel für eine Partitionsspalte eine Datumsspalte.
- Wählen Sie eine Partitionsspalte mit niedriger Kardinalität, um zu vermeiden, dass zu viele Partitionen erstellt werden. Zu viele Partitionen können die Anzahl der Dateien in der Tabelle erhöhen, was sich negativ auf die Abfrageleistung auswirken kann. Als Faustregel gilt, dass „zu viele Partitionen“ als ein Szenario definiert werden können, in dem die Datengröße in den meisten Partitionen weniger als das Zwei- bis Fünffache des von `target-file-size-bytes` festgelegten Werts beträgt.

> **Note**
>
> Wenn Sie in der Regel Filter für eine Spalte mit hoher Kardinalität verwenden (z. B. eine `id` Spalte, die Tausende von Werten haben kann), verwenden Sie die Funktion für versteckte Partitionierung von Iceberg mit Bucket-Transformationen, wie im nächsten Abschnitt erklärt.

Verwenden Sie versteckte Partitionierung

Wenn Ihre Abfragen häufig nach einer Ableitung einer Tabellenspalte filtern, verwenden Sie versteckte Partitionen, anstatt explizit neue Spalten zu erstellen, die als Partitionen verwendet werden können. Weitere Informationen zu dieser Funktion finden Sie in der Iceberg-Dokumentation.

Verwenden Sie beispielsweise in einem Datensatz mit einer Zeitstempelspalte (z. B.`2023-01-01 09:00:00`) Partitionstransformationen, um den Datumsanteil aus dem Zeitstempel zu extrahieren

und diese Partitionen im Handumdrehen zu erstellen, anstatt eine neue Spalte mit dem analysierten Datum zu erstellen (z. B.`2023-01-01`).

Die häufigsten Anwendungsfälle für versteckte Partitionierung sind:

- Partitionierung nach Datum oder Uhrzeit, wenn die Daten eine Zeitstempelspalte haben. Iceberg bietet mehrere Transformationen, um die Datums- oder Uhrzeitteile eines Zeitstempels zu extrahieren.
- Partitionierung nach einer Hash-Funktion einer Spalte, wenn die partitionierende Spalte eine hohe Kardinalität aufweist und zu vielen Partitionen führen würde. Die Bucket-Transformation von Iceberg gruppiert mehrere Partitionswerte zu weniger versteckten (Bucket-) Partitionen, indem Hash-Funktionen für die Partitionierungsspalte verwendet werden.

Einen Überblick über alle verfügbaren Partitionstransformationen finden Sie unter Partitionstransformationen in der Iceberg-Dokumentation.

Spalten, die für versteckte Partitionierung verwendet werden, können durch die Verwendung regulärer SQL-Funktionen wie und Teil von Abfrageprädikaten werden. `year()` `month()` Prädikate können auch mit Operatoren wie und kombiniert werden. `BETWEEN` `AND`

> **Note**
>
> Iceberg kann keine Partitionsbereinigung für Funktionen durchführen, die einen anderen Datentyp ergeben, z. B. `substring(event_time, 1, 10) = '2022-01-01'`

Verwenden Sie Partition Evolution

Verwenden Sie die Partitionsentwicklung von Iceberg, wenn die bestehende Partitionsstrategie nicht optimal ist. Wenn Sie beispielsweise stündliche Partitionen wählen, die sich als zu klein herausstellen (jeweils nur ein paar Megabyte), sollten Sie eine Umstellung auf tägliche oder monatliche Partitionen in Betracht ziehen.

Sie können diesen Ansatz verwenden, wenn die beste Partitionsstrategie für eine Tabelle zunächst unklar ist und Sie Ihre Partitionierungsstrategie verfeinern möchten, wenn Sie mehr Erkenntnisse gewinnen. Eine weitere effektive Anwendung der Partitionsentwicklung besteht darin, dass sich die Datenmengen ändern und die aktuelle Partitionierungsstrategie im Laufe der Zeit an Effektivität verliert.

Anweisungen zur Weiterentwicklung von Partitionen finden Sie unter ALTER TABLE SQL-Erweiterungen in der Iceberg-Dokumentation.

Optimieren der Dateigrößen

Zur Optimierung der Abfrageleistung gehört die Minimierung der Anzahl kleiner Dateien in Ihren Tabellen. Für eine gute Abfrageleistung empfehlen wir generell, Parquet- und ORC-Dateien größer als 100 MB zu verwenden.

Die Dateigröße wirkt sich auch auf die Abfrageplanung für Iceberg-Tabellen aus. Mit zunehmender Anzahl von Dateien in einer Tabelle nimmt auch die Größe der Metadatendateien zu. Größere Metadatendateien können zu einer langsameren Abfrageplanung führen. Wenn die Tabellengröße zunimmt, sollten Sie daher die Dateigröße erhöhen, um die exponentielle Erweiterung der Metadaten zu verringern.

Verwenden Sie die folgenden bewährten Methoden, um Dateien mit der richtigen Größe in Iceberg-Tabellen zu erstellen.

Legen Sie die Größe der Zieldatei und der Zeilengruppe fest

Iceberg bietet die folgenden wichtigen Konfigurationsparameter zur Optimierung des Layouts der Datendatei. Wir empfehlen, dass Sie diese Parameter verwenden, um die Zieldateigröße und die Zeilengruppen- oder Strike-Größe festzulegen.

Parameter	Standardwert	Kommentar
`write.target-file-size-bytes`	512 MB	Dieser Parameter gibt die maximale Dateigröße an, die Iceberg erstellen wird. Bestimmte Dateien können jedoch mit einer kleineren Größe als diesem Limit geschrieben werden.
`write.parquet.row-group-size-bytes`	128 MB	Sowohl Parquet als auch ORC speichern Daten in Blöcken, sodass Engines bei einigen Vorgängen vermeiden können, die gesamte Datei zu lesen.

Parameter	Standardwert	Kommentar
`write.orc.stripe-size-bytes`	64 MB	
`write.distribution-mode`	Keine, für Iceberg Version 1.1 und niedriger Hash, beginnend mit Iceberg Version 1.2	Iceberg fordert Spark auf, Daten zwischen seinen Aufgaben zu sortieren, bevor sie in den Speicher geschrieben werden.

- Basierend auf Ihrer erwarteten Tabellengröße sollten Sie sich an diese allgemeinen Richtlinien halten:
  - Kleine Tabellen (bis zu einigen Gigabyte) — Reduzieren Sie die Zieldateigröße auf 128 MB. Reduzieren Sie auch die Zeilengruppen- oder Stripe-Größe (z. B. auf 8 oder 16 MB).
  - Mittlere bis große Tabellen (von einigen Gigabyte bis zu Hunderten von Gigabyte) — Die Standardwerte sind ein guter Ausgangspunkt für diese Tabellen. Wenn Ihre Abfragen sehr selektiv sind, passen Sie die Zeilengruppen- oder Stripe-Größe an (z. B. auf 16 MB).
  - Sehr große Tabellen (Hunderte von Gigabyte oder Terabyte) — Erhöhen Sie die Zieldateigröße auf 1024 MB oder mehr und erwägen Sie, die Zeilengruppen- oder Stripe-Größe zu erhöhen, wenn Ihre Abfragen normalerweise große Datenmengen abrufen.

- Um sicherzustellen, dass Spark-Anwendungen, die in Iceberg-Tabellen schreiben, Dateien mit der entsprechenden Größe erstellen, setzen Sie die `write.distribution-mode` Eigenschaft entweder auf oder. `hash` `range` Eine ausführliche Erklärung des Unterschieds zwischen diesen Modi finden Sie in der Iceberg-Dokumentation unter Writing Distribution Modes.

Dies sind allgemeine Richtlinien. Wir empfehlen Ihnen, Tests durchzuführen, um die am besten geeigneten Werte für Ihre spezifischen Tabellen und Workloads zu ermitteln.

Führen Sie eine regelmäßige Komprimierung durch

Die Konfigurationen in der vorherigen Tabelle legen eine maximale Dateigröße fest, die Schreibaufgaben erzeugen können, garantieren jedoch nicht, dass Dateien diese Größe haben. Führen Sie die Komprimierung regelmäßig durch, um kleine Dateien zu größeren Dateien zusammenzufassen, um die richtigen Dateigrößen sicherzustellen. Eine ausführliche Anleitung

zum Ausführen der Komprimierung finden Sie weiter unten in diesem Handbuch unter Iceberg-Komprimierung.

Optimieren Sie die Spaltenstatistiken

Iceberg verwendet Spaltenstatistiken, um Dateien zu bereinigen. Dadurch wird die Abfrageleistung verbessert, da die Datenmenge, die durch Abfragen gescannt wird, reduziert wird. Um von Spaltenstatistiken zu profitieren, stellen Sie sicher, dass Iceberg Statistiken für alle Spalten sammelt, die häufig in Abfragefiltern verwendet werden.

Standardmäßig sammelt Iceberg Statistiken nur für die ersten 100 Spalten in jeder Tabelle, wie in der Tabelleneigenschaft definiert. `write.metadata.metrics.max-inferred-column-defaults` Wenn Ihre Tabelle mehr als 100 Spalten hat und Ihre Abfragen häufig auf Spalten außerhalb der ersten 100 Spalten verweisen (z. B. wenn Sie Abfragen haben, die nach Spalte 132 filtern), stellen Sie sicher, dass Iceberg Statistiken zu diesen Spalten sammelt. Um dies zu erreichen, gibt es zwei Möglichkeiten:

- Wenn Sie die Iceberg-Tabelle erstellen, ordnen Sie die Spalten neu an, sodass die Spalten, die Sie für Abfragen benötigen, in den angegebenen Spaltenbereich fallen `write.metadata.metrics.max-inferred-column-defaults` (Standard ist 100).

  Hinweis: Wenn Sie keine Statistiken für 100 Spalten benötigen, können Sie die `write.metadata.metrics.max-inferred-column-defaults` Konfiguration auf einen gewünschten Wert (z. B. 20) anpassen und die Spalten neu anordnen, sodass die Spalten, die Sie zum Lesen und Schreiben von Abfragen benötigen, in die ersten 20 Spalten auf der linken Seite des Datensatzes fallen.
- Wenn Sie in Abfragefiltern nur wenige Spalten verwenden, können Sie die allgemeine Eigenschaft für die Erfassung von Metriken deaktivieren und selektiv einzelne Spalten auswählen, für die Statistiken erfasst werden sollen, wie in diesem Beispiel gezeigt:

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

Hinweis: Spaltenstatistiken sind am effektivsten, wenn die Daten nach diesen Spalten sortiert sind. Weitere Informationen finden Sie im Abschnitt „Sortierreihenfolge festlegen“ weiter unten in diesem Handbuch.

Wählen Sie die richtige Aktualisierungsstrategie

Verwenden Sie eine copy-on-write Strategie zur Optimierung der Leseleistung, wenn langsamere Schreibvorgänge für Ihren Anwendungsfall akzeptabel sind. Dies ist die von Iceberg verwendete Standardstrategie.

Copy-on-write führt zu einer besseren Leseleistung, da Dateien leseoptimiert direkt in den Speicher geschrieben werden. Im Vergleich merge-on-read zu dauert jeder Schreibvorgang jedoch länger und verbraucht mehr Rechenressourcen. Dies stellt einen klassischen Kompromiss zwischen Lese- und Schreiblatenz dar. In der Regel copy-on-write ist dies ideal für Anwendungsfälle, in denen die meisten Aktualisierungen in denselben Tabellenpartitionen gespeichert sind (z. B. für tägliche Batchladevorgänge).

Copy-on-write Konfigurationen (`write.update.mode`write.delete.mode, und`write.merge.mode`) können auf Tabellenebene oder unabhängig voneinander auf der Anwendungsseite festgelegt werden.

Verwenden Sie die STD-Komprimierung

Sie können den von Iceberg verwendeten Komprimierungscodec mithilfe der Tabelleneigenschaft ändern. `write.<file_type>.compression-codec` Wir empfehlen, den ZSTD-Komprimierungscodec zu verwenden, um die Gesamtleistung von Tabellen zu verbessern.

Standardmäßig verwenden die Iceberg-Versionen 1.3 und früher die GZIP-Komprimierung, die im Vergleich zu ZSTD eine langsamere read/write Leistung bietet.

Hinweis: Einige Engines verwenden möglicherweise andere Standardwerte. Dies ist bei Iceberg-Tabellen der Fall, die mit Athena oder Amazon EMR Version 7.x erstellt wurden.

Legen Sie die Sortierreihenfolge fest

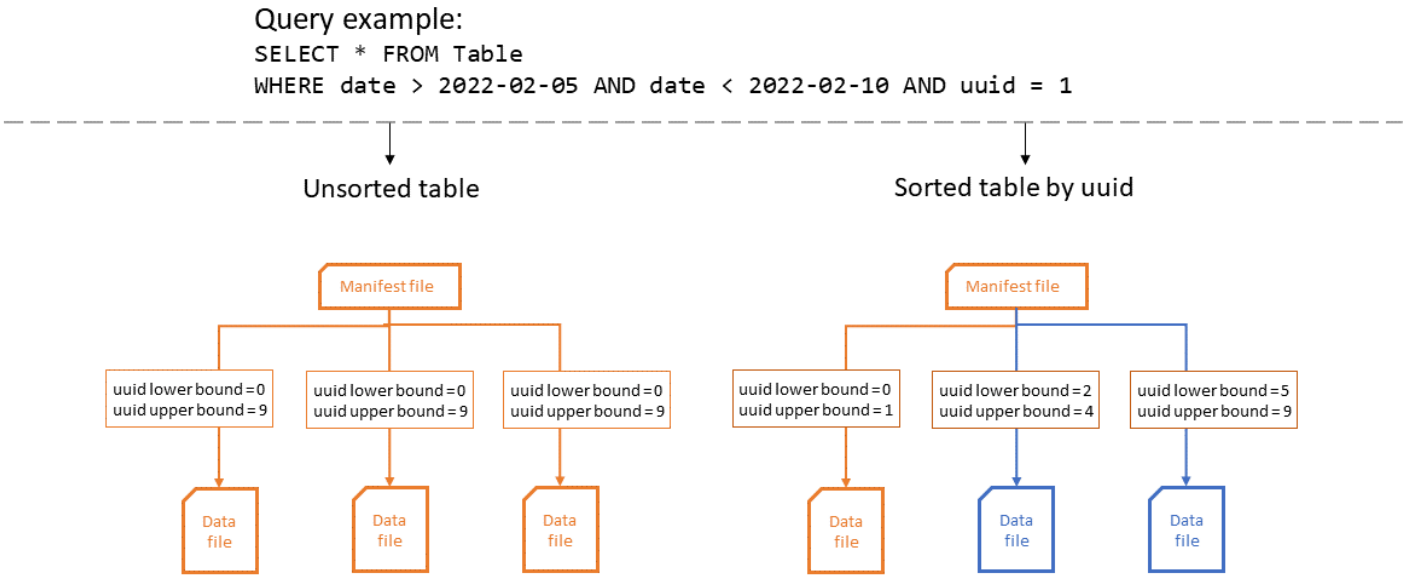
Um die Leseleistung bei Iceberg-Tabellen zu verbessern, empfehlen wir, dass Sie Ihre Tabelle nach einer oder mehreren Spalten sortieren, die häufig in Abfragefiltern verwendet werden. Durch Sortierung in Kombination mit den Spaltenstatistiken von Iceberg kann das Bereinigen von Dateien erheblich effizienter gestaltet werden, was zu schnelleren Lesevorgängen führt. Durch die Sortierung wird auch die Anzahl der Amazon S3 S3-Anfragen für Abfragen reduziert, die die Sortierspalten in Abfragefiltern verwenden.

Sie können eine hierarchische Sortierreihenfolge auf Tabellenebene festlegen, indem Sie eine DDL-Anweisung (Data Definition Language) mit Spark ausführen. Die verfügbaren Optionen finden Sie in

der Iceberg-Dokumentation. Nachdem Sie die Sortierreihenfolge festgelegt haben, wenden Autoren diese Sortierung auf nachfolgende Datenschreibvorgänge in der Iceberg-Tabelle an.

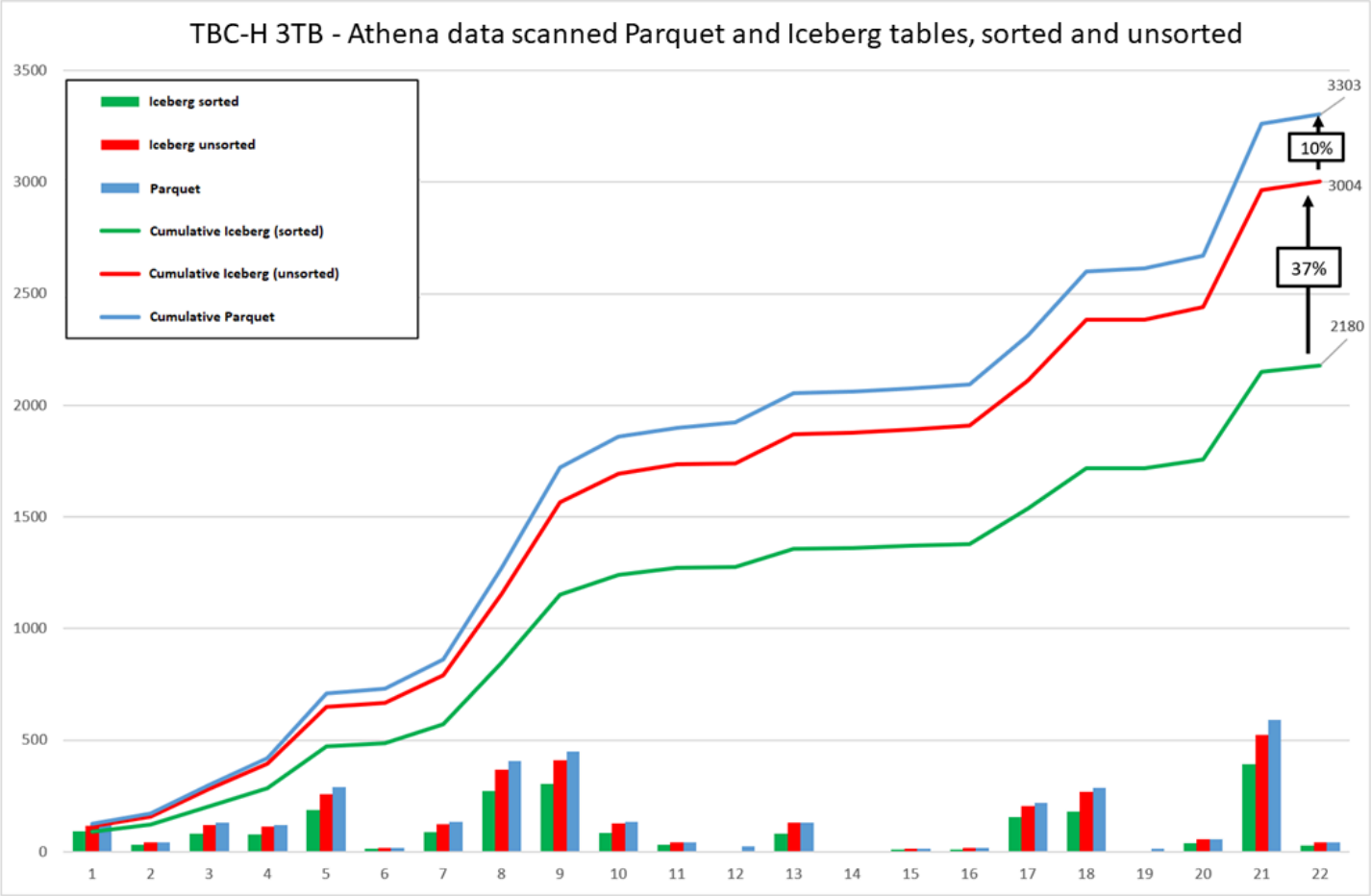
In Tabellen, die nach Datum (`yyyy-mm-dd`) partitioniert sind, nach denen die meisten Abfragen filternuuid, können Sie beispielsweise die DDL-Option verwenden, `Write Distributed By Partition Locally Ordered` um sicherzustellen, dass Spark Dateien mit nicht überlappenden Bereichen schreibt.

Das folgende Diagramm zeigt, wie sich die Effizienz von Spaltenstatistiken verbessert, wenn Tabellen sortiert werden. In diesem Beispiel muss die sortierte Tabelle nur eine einzige Datei öffnen und profitiert optimal von der Partition und Datei von Iceberg. In der unsortierten Tabelle `uuid` kann jede Datei potenziell in jeder beliebigen Datendatei vorkommen, sodass die Abfrage alle Datendateien öffnen muss.

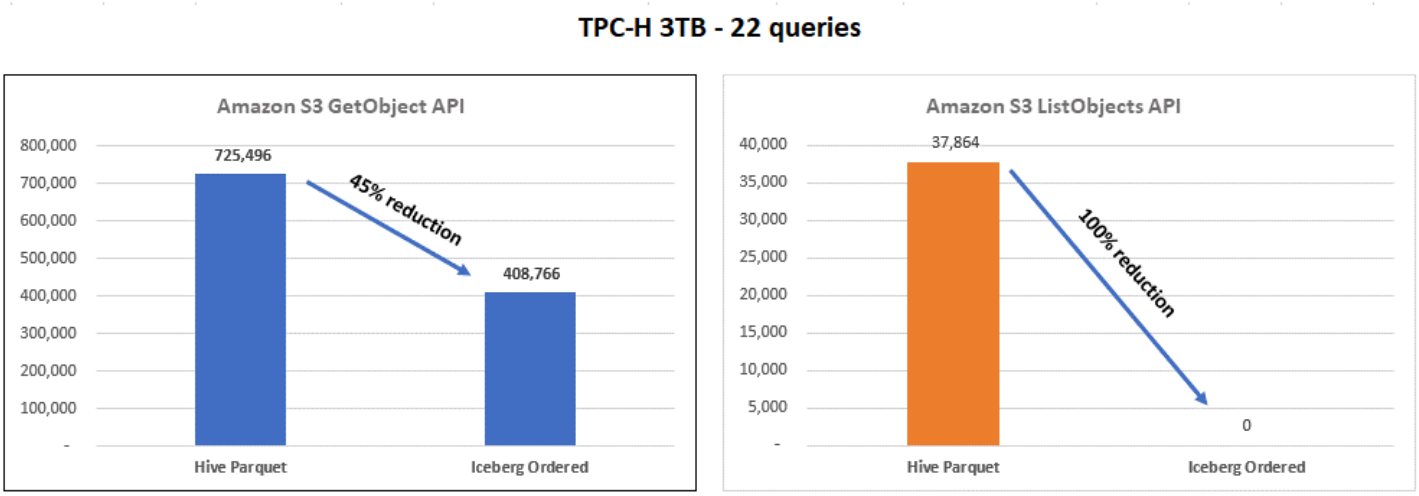


Das Ändern der Sortierreihenfolge hat keine Auswirkungen auf bestehende Datendateien. Sie können die Iceberg-Komprimierung verwenden, um die Sortierreihenfolge auf diese anzuwenden.

Die Verwendung von nach Iceberg sortierten Tabellen kann die Kosten für Ihre Arbeitslast senken, wie in der folgenden Grafik dargestellt.



Diese Grafiken fassen die Ergebnisse der Durchführung des TPC-H-Benchmarks für Hive-Tabellen (Parquet) im Vergleich zu sortierten Iceberg-Tabellen zusammen. Bei anderen Datensätzen oder Workloads können die Ergebnisse jedoch anders ausfallen.



Optimierung der Schreibleistung

In diesem Abschnitt werden Tabelleneigenschaften beschrieben, die Sie anpassen können, um die Schreibleistung auf Iceberg-Tabellen unabhängig von der Engine zu optimieren.

Stellen Sie den Tabellenverteilungsmodus ein

Iceberg bietet mehrere Schreibverteilungsmodi, die definieren, wie Schreibdaten auf Spark-Aufgaben verteilt werden. Einen Überblick über die verfügbaren Modi finden Sie unter Writing Distribution Modes in der Iceberg-Dokumentation.

Für Anwendungsfälle, in denen der Schreibgeschwindigkeit Priorität eingeräumt wird, insbesondere bei Streaming-Workloads, legen Sie die Einstellung auf fest. `write.distribution-mode` `none` Dadurch wird sichergestellt, dass Iceberg kein zusätzliches Spark-Shuffling anfordert und dass Daten geschrieben werden, sobald sie in Spark-Aufgaben verfügbar sind. Dieser Modus eignet sich besonders für Spark Structured Streaming-Anwendungen.

> **Note**
>
> Wenn Sie den Schreibverteilungsmodus auf einstellen, werden in der `none` Regel zahlreiche kleine Dateien erzeugt, was die Leseleistung beeinträchtigt. Wir empfehlen eine regelmäßige Komprimierung, um diese kleinen Dateien aus Gründen der Abfrageleistung in Dateien mit der richtigen Größe zu konsolidieren.

Wählen Sie die richtige Aktualisierungsstrategie

Verwenden Sie eine merge-on-read Strategie zur Optimierung der Schreibleistung, wenn langsamere Lesevorgänge mit den neuesten Daten für Ihren Anwendungsfall akzeptabel sind.

Wenn Sie Iceberg verwenden merge-on-read, schreibt Iceberg Aktualisierungen und Löschungen als separate kleine Dateien in den Speicher. Wenn die Tabelle gelesen wird, muss der Leser diese Änderungen mit den Basisdateien zusammenführen, um die neueste Ansicht der Daten zurückzugeben. Dies führt zu Leistungseinbußen bei Lesevorgängen, beschleunigt aber das Schreiben von Aktualisierungen und Löschungen. In der Regel merge-on-read ist es ideal für das Streaming von Workloads mit Updates oder Jobs mit wenigen Updates, die auf viele Tabellenpartitionen verteilt sind.

Sie können merge-on-read Konfigurationen (`write.update.modewrite.delete.mode`, und`write.merge.mode`) auf Tabellenebene oder unabhängig voneinander auf der Anwendungsseite festlegen.

Die Verwendung merge-on-read erfordert eine regelmäßige Komprimierung, um zu verhindern, dass sich die Leseleistung im Laufe der Zeit verschlechtert. Durch die Komprimierung werden Aktualisierungen und Löschungen mit vorhandenen Datendateien abgeglichen, um einen neuen Satz von Datendateien zu erstellen, wodurch die Leistungseinbußen beim Lesen vermieden werden. Standardmäßig werden bei der Komprimierung von Iceberg gelöschte Dateien nicht zusammengeführt, es sei denn, Sie ändern die Standardeinstellung der `delete-file-threshold` Eigenschaft auf einen kleineren Wert (siehe die Iceberg-Dokumentation). Weitere Informationen zur Komprimierung finden Sie im Abschnitt Eisberg-Komprimierung weiter unten in diesem Handbuch.

Wählen Sie das richtige Dateiformat

Iceberg unterstützt das Schreiben von Daten in den Formaten Parquet, ORC und Avro. Parquet ist das Standardformat. Parquet und ORC sind spaltenförmige Formate, die eine hervorragende Leseleistung bieten, aber im Allgemeinen langsamer zu schreiben sind. Dies stellt den typischen Kompromiss zwischen Lese- und Schreibleistung dar.

Wenn die Schreibgeschwindigkeit für Ihren Anwendungsfall wichtig ist, z. B. bei Streaming-Workloads, sollten Sie erwägen, im Avro-Format zu schreiben, indem Sie `Avro` in den `write-format` Writer-Optionen auf setzen. Da Avro ein zeilenbasiertes Format ist, bietet es schnellere Schreibzeiten auf Kosten einer langsameren Leseleistung.

Um die Leseleistung zu verbessern, sollten Sie regelmäßig komprimieren, um kleine Avro-Dateien zusammenzuführen und in größere Parquet-Dateien umzuwandeln. Das Ergebnis des Verdichtungsvorgangs wird durch die `write.format.default` Tabelleneinstellung bestimmt. Das Standardformat für Iceberg ist Parquet. Wenn Sie also in Avro schreiben und dann die Komprimierung ausführen, wandelt Iceberg die Avro-Dateien in Parquet-Dateien um. Hier ein Beispiel:

```
spark.sql(f"""
    CREATE TABLE IF NOT EXISTS glue_catalog.{DB_NAME}.{TABLE_NAME} (
        Col_1 float,
        <<<…other columns…>>
        ts timestamp)
    USING iceberg
    PARTITIONED BY (days(ts))
```

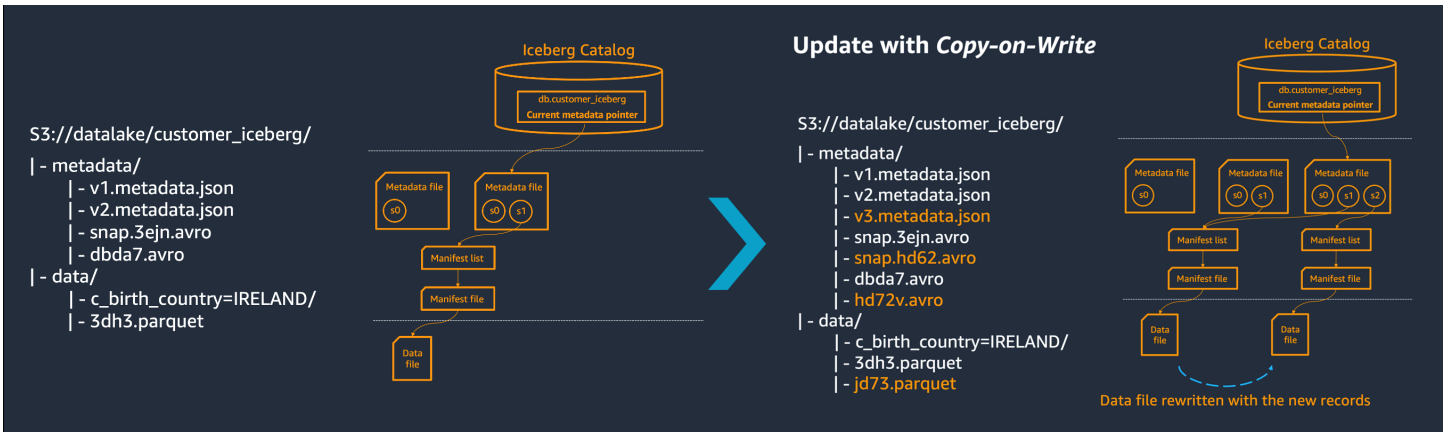
```
    OPTIONS (
      'format-version'='2',
      write.format.default'=parquet)
""")

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

    .start()
```

Optimierung des Speichers

Durch das Aktualisieren oder Löschen von Daten in einer Iceberg-Tabelle erhöht sich die Anzahl der Kopien Ihrer Daten, wie in der folgenden Abbildung dargestellt. Das Gleiche gilt für die laufende Komprimierung: Sie erhöht die Anzahl der Datenkopien in Amazon S3. Das liegt daran, dass Iceberg die allen Tabellen zugrunde liegenden Dateien als unveränderlich behandelt.



Folgen Sie den bewährten Methoden in diesem Abschnitt, um die Speicherkosten zu verwalten.

Aktivieren Sie S3 Intelligent-Tiering

Verwenden Sie die Amazon S3 S3-Speicherklasse Intelligent-Tiering, um Daten automatisch auf die kostengünstigste Zugriffsebene zu verschieben, wenn sich die Zugriffsmuster ändern. Diese Option hat weder Betriebsaufwand noch Auswirkungen auf die Leistung.

Hinweis: Verwenden Sie nicht die optionalen Stufen (wie Archive Access und Deep Archive Access) in S3 Intelligent-Tiering mit Iceberg-Tabellen. Informationen zur Archivierung von Daten finden Sie in den Richtlinien im nächsten Abschnitt.

Sie können Amazon S3 S3-Lebenszyklusregeln auch verwenden, um Ihre eigenen Regeln für das Verschieben von Objekten in eine andere Amazon S3 S3-Speicherklasse wie S3 Standard-IA oder S3 One Zone-IA festzulegen (siehe Unterstützte Übergänge und zugehörige Einschränkungen in der Amazon S3 S3-Dokumentation).

Archivieren oder löschen Sie historische Schnappschüsse

Für jede festgeschriebene Transaktion (Einfügen, Aktualisieren, Zusammenführen, Komprimieren) zu einer Iceberg-Tabelle wird eine neue Version oder ein Snapshot der Tabelle erstellt. Im Laufe der Zeit häufen sich die Anzahl der Versionen und die Anzahl der Metadatendateien in Amazon S3 an.

Das Speichern von Snapshots einer Tabelle ist für Funktionen wie Snapshot-Isolation, Tabellen-Rollback und Zeitreiseabfragen erforderlich. Die Speicherkosten steigen jedoch mit der Anzahl der Versionen, die Sie behalten.

In der folgenden Tabelle werden die Entwurfsmuster beschrieben, die Sie implementieren können, um die Kosten auf der Grundlage Ihrer Datenaufbewahrungsanforderungen zu verwalten.

Entwurfsmuster	Lösung	Anwendungsfälle
Lösche alte Schnappschüsse	• Verwenden Sie die VACUUM-Anweisung in Athena, um alte Schnappschüsse zu entfernen. Für diesen Vorgang fallen keine Rechenkosten an. • Alternativ können Sie Spark auf Amazon EMR verwenden oder AWS Glue Snapshots entfernen. Weitere Informationen finden Sie unter expire_sn	Bei diesem Ansatz werden Snapshots gelöscht, die nicht mehr benötigt werden, um die Speicherkosten zu senken. Sie können je nach Ihren Datenaufbewahrungsanforderungen konfigurieren, wie viele Snapshots aufbewahrt werden sollen oder wie lange. Mit dieser Option werden die Snapshots dauerhaft gelöscht. Ein Rollback oder

Entwurfsmuster	Lösung	Anwendungsfälle
	apshots in der Iceberg-D okumentation.	eine Zeitreise zu abgelaufenen Snapshots sind nicht möglich.
Legen Sie Aufbewahrungsricht linien für bestimmte Snapshots fest	1. Verwenden Sie Tags, um bestimmte Snapshots zu markieren und eine Aufbewahrungsrichtlinie in Iceberg zu definiere n. Weitere Informationen finden Sie unter Historisc he Tags in der Iceberg-D okumentation. Sie können beispiels weise einen Snapshot pro Monat für ein Jahr aufbewahren, indem Sie die folgende SQL-Anweisung in Spark auf Amazon EMR verwenden: `ALTER TABLE glue_cata log.db.table CREATE TAG 'EOM-01' AS OF VERSION 30 RETAIN 365 DAYS` 2. Verwenden Sie Spark auf Amazon EMR oder AWS Glue um die verbleibenden Snapshots ohne Tags zu entfernen.	Dieses Muster ist hilfreich bei der Einhaltung geschäftlicher oder rechtlicher Anforderu ngen, nach denen Sie den Status einer Tabelle zu einem bestimmten Zeitpunkt in der Vergangenheit anzeigen müssen. Indem Sie Aufbewahr ungsrichtlinien für bestimmte markierte Snapshots festlegen , können Sie andere (nicht markierte) Snapshots entfernen, die erstellt wurden. Auf diese Weise können Sie die Anforderungen an die Datenaufbewahrung erfüllen, ohne jeden einzelnen erstellte n Snapshot behalten zu müssen.

Entwurfsmuster	Lösung	Anwendungsfälle
Archivieren Sie alte Schnappschüsse	1. Verwenden Sie Amazon S3 S3-Tags, um Objekte mit Spark zu markieren . (Amazon S3 S3-Tags unterscheiden sich von Iceberg-Tags. Weitere Informationen finden Sie in der Iceberg-Dokumentation.) Beispiel: `spark.sql.catalog.my_catalog.s3.delete-enabled=false and \ spark.sql.catalog.my_catalog.s3.delete.tags.my_key=to_archive` 2. Verwenden Sie Spark auf Amazon EMR oder AWS Glue um Snapshots zu entfernen. Wenn Sie die Einstellungen im Beispiel verwenden, kennzeichnet dieses Verfahren Objekte und trennt sie von den Metadaten der Iceberg-Tabelle, anstatt sie aus Amazon S3 zu löschen. 3. Verwenden Sie S3-Lebenszyklusregeln, um Objekte, die als gekennzeichnet sind`to_archive` , in eine der S3 Glacier-Speicherklassen zu übertragen.	Dieses Muster ermöglicht es Ihnen, alle Tabellenversionen und Snapshots zu geringeren Kosten aufzubewahren. Sie können keine Zeitreise unternehmen oder zu archivierten Snapshots zurückkehren, ohne diese Versionen zuerst als neue Tabellen wiederherzustellen. Dies ist in der Regel für Prüfungszwecke akzeptabel. Sie können diesen Ansatz mit dem vorherigen Entwurfsmuster kombinieren und Aufbewahrungsrichtlinien für bestimmte Snapshots festlegen.

Entwurfsmuster	Lösung	Anwendungsfälle
	4. Um archivierte Daten abzufragen: • Stellen Sie die archivierten Objekte wieder her (dieser Schritt ist nicht erforderlich, wenn Objekte in die Amazon Glacier Instant Retrieval-Speicherklasse übertragen wurden). • Verwenden Sie die Prozedur register_table in Iceberg, um den Snapshot als Tabelle im Katalog zu registrieren. Eine ausführliche Anleitung finden Sie im AWS Blogbeitrag Verbessern Sie die betriebliche Effizienz von Apache Iceberg-Tabellen, die auf Amazon S3 S3-Datenseen basieren.	

Löschen Sie verwaiste Dateien

In bestimmten Situationen können Iceberg-Anwendungen fehlschlagen, bevor Sie Ihre Transaktionen festschreiben. Dadurch verbleiben Datendateien in Amazon S3. Da kein Commit vorgenommen wurde, werden diese Dateien keiner Tabelle zugeordnet, sodass Sie sie möglicherweise asynchron bereinigen müssen.

Um diese Löschungen zu handhaben, können Sie die VACUUM-Anweisung in Amazon Athena verwenden. Diese Anweisung entfernt Schnappschüsse und löscht auch verwaiste Dateien. Dies ist sehr kosteneffizient, da Athena die Rechenkosten für diesen Vorgang nicht in Rechnung stellt. Außerdem müssen Sie keine zusätzlichen Operationen planen, wenn Sie die `VACUUM` Anweisung verwenden.

Alternativ können Sie Spark auf Amazon EMR verwenden oder AWS Glue das `remove_orphan_files` Verfahren ausführen. Dieser Vorgang ist mit Rechenkosten verbunden und muss unabhängig geplant werden. Weitere Informationen finden Sie in der Iceberg-Dokumentation.

Pflege von Tabellen mithilfe von Komprimierung

Iceberg enthält Funktionen, mit denen Sie Tabellenverwaltungsoperationen durchführen können, nachdem Sie Daten in die Tabelle geschrieben haben. Einige Wartungsvorgänge konzentrieren sich auf die Optimierung von Metadatendateien, während andere die Clusterung der Daten in den Dateien verbessern, sodass Abfrage-Engines die für die Beantwortung von Benutzeranfragen erforderlichen Informationen effizient finden können. Dieser Abschnitt konzentriert sich auf Optimierungen im Zusammenhang mit der Komprimierung.

Verdichtung von Eisbergen

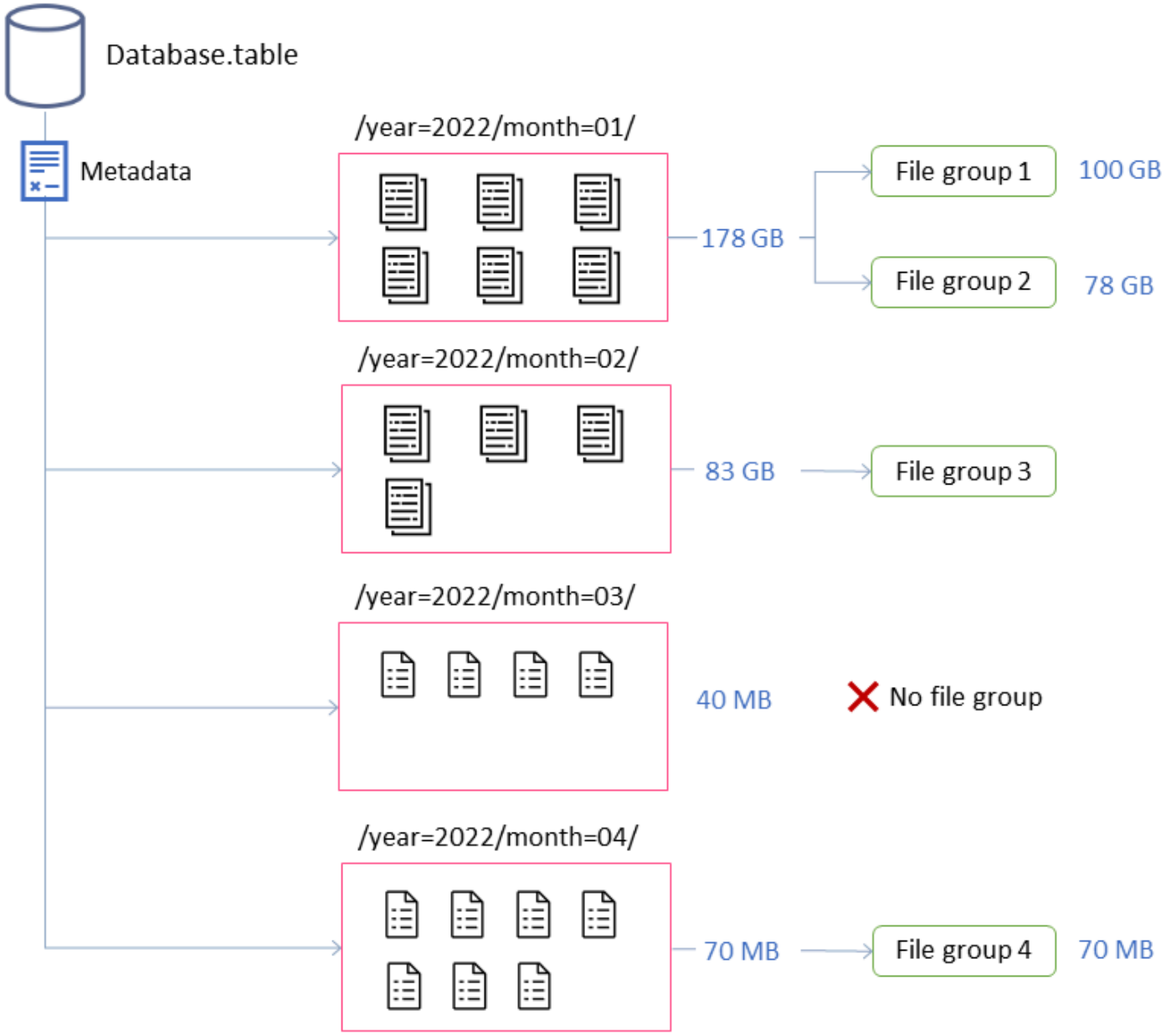
In Iceberg können Sie die Komprimierung verwenden, um vier Aufgaben auszuführen:

- Kombinieren kleiner Dateien zu größeren Dateien, die in der Regel über 100 MB groß sind. Diese Technik wird als Mülltonnenverpackung bezeichnet.
- Zusammenführen von gelöschten Dateien mit Datendateien. Löschdateien werden durch Aktualisierungen oder Löschungen generiert, die diesen Ansatz verwenden. merge-on-read
- (Neu-) Sortierung der Daten gemäß Abfragemustern. Daten können ohne jegliche Sortierreihenfolge oder mit einer Sortierreihenfolge geschrieben werden, die für Schreibvorgänge und Aktualisierungen geeignet ist.
- Clustering der Daten mithilfe von raumfüllenden Kurven zur Optimierung für unterschiedliche Abfragemuster, insbesondere für die Sortierung in Z-Reihenfolge.

Auf AWS, können Sie Tabellenkomprimierungs- und Wartungsvorgänge für Iceberg über Amazon Athena oder mithilfe von Spark in Amazon EMR oder ausführen. AWS Glue

Wenn Sie die Komprimierung mithilfe der Prozedur rewrite_data_files ausführen, können Sie mehrere Regler einstellen, um das Verdichtungsverhalten zu steuern. Das folgende Diagramm zeigt das

Standardverhalten beim Packen von Behältern. Das Verständnis der Kompaktierung von Behältern ist für das Verständnis der Implementierungen der hierarchischen Sortierung und der Sortierung in Z-Reihenfolge von entscheidender Bedeutung, da es sich dabei um Erweiterungen der Schnittstelle zum Packen von Behältern handelt, die auf ähnliche Weise funktionieren. Der Hauptunterschied besteht in dem zusätzlichen Schritt, der zum Sortieren oder Clustern der Daten erforderlich ist.



In diesem Beispiel besteht die Iceberg-Tabelle aus vier Partitionen. Jede Partition hat eine andere Größe und eine unterschiedliche Anzahl von Dateien. Wenn Sie eine Spark-Anwendung starten, um die Komprimierung auszuführen, erstellt die Anwendung insgesamt vier Dateigruppen zur Verarbeitung. Eine Dateigruppe ist eine Iceberg-Abstraktion, die eine Sammlung von Dateien

darstellt, die von einem einzelnen Spark-Job verarbeitet werden. Das heißt, die Spark-Anwendung, die die Komprimierung ausführt, erstellt vier Spark-Jobs zur Verarbeitung der Daten.

Optimieren des Verdichtungsverhaltens

Die folgenden wichtigen Eigenschaften steuern, wie Datendateien für die Komprimierung ausgewählt werden:

- MAX_FILE_GROUP_SIZE_BYTES legt das Datenlimit für eine einzelne Dateigruppe (Spark-Job) standardmäßig auf 100 GB fest. Diese Eigenschaft ist besonders wichtig für Tabellen ohne Partitionen oder Tabellen mit Partitionen, die sich über Hunderte von Gigabyte erstrecken. Wenn Sie dieses Limit festlegen, können Sie die Abläufe unterteilen, um die Arbeit zu planen und Fortschritte zu erzielen, und gleichzeitig eine Erschöpfung der Ressourcen im Cluster verhindern.

  Hinweis: Jede Dateigruppe ist separat sortiert. Wenn Sie also eine Sortierung auf Partitionsebene durchführen möchten, müssen Sie dieses Limit an die Partitionsgröße anpassen.
- MIN_FILE_SIZE_BYTES oder MIN_FILE_SIZE_DEFAULT_RATIO sind standardmäßig auf 75 Prozent der auf Tabellenebene festgelegten Zieldateigröße eingestellt. Wenn eine Tabelle beispielsweise eine Zielgröße von 512 MB hat, ist jede Datei, die kleiner als 384 MB ist, in der Gruppe der zu komprimierenden Dateien enthalten.
- MAX_FILE_SIZE_BYTES oder MAX_FILE_SIZE_DEFAULT_RATIO sind standardmäßig auf 180 Prozent der Zieldateigröße eingestellt. Wie bei den beiden Eigenschaften, die Mindestdateigrößen festlegen, werden diese Eigenschaften verwendet, um Kandidatendateien für den Komprimierungsauftrag zu identifizieren.
- MIN_INPUT_FILES gibt die Mindestanzahl der zu komprimierenden Dateien an, wenn die Größe einer Tabellenpartition kleiner als die Zieldateigröße ist. Der Wert dieser Eigenschaft wird verwendet, um zu bestimmen, ob es sich lohnt, die Dateien anhand der Anzahl der Dateien zu komprimieren (der Standardwert ist 5).
- DELETE_FILE_THRESHOLD gibt die Mindestanzahl von Löschvorgängen für eine Datei an, bevor sie in die Komprimierung aufgenommen wird. Sofern Sie nichts anderes angeben, werden bei der Komprimierung keine Löschdateien mit Datendateien kombiniert. Um diese Funktion zu aktivieren, müssen Sie mithilfe dieser Eigenschaft einen Schwellenwert festlegen. Dieser Schwellenwert ist spezifisch für einzelne Datendateien. Wenn Sie ihn also auf 3 setzen, wird eine Datendatei nur dann neu geschrieben, wenn drei oder mehr Löschdateien darauf verweisen.

Diese Eigenschaften geben Aufschluss über die Bildung der Dateigruppen im vorherigen Diagramm.

Die beschriftete Partition `month=01` umfasst beispielsweise zwei Dateigruppen, da sie die maximale Größenbeschränkung von 100 GB überschreitet. Im Gegensatz dazu enthält die `month=02` Partition eine einzelne Dateigruppe, da sie weniger als 100 GB groß ist. Die `month=03` Partition erfüllt nicht die standardmäßige Mindestanforderung für Eingabedateien von fünf Dateien. Infolgedessen wird sie nicht komprimiert. Schließlich werden die Dateien komprimiert, obwohl die `month=04` Partition nicht genügend Daten enthält, um eine einzelne Datei mit der gewünschten Größe zu bilden, da die Partition mehr als fünf kleine Dateien enthält.

Sie können diese Parameter für Spark festlegen, der auf Amazon EMR oder AWS Glue ausgeführt wird. Für Amazon Athena können Sie ähnliche Eigenschaften verwalten, indem Sie die Tabelleneigenschaften verwenden, die mit dem Präfix beginnen`optimize_`).

Verdichtung mit Spark auf Amazon EMR ausführen oder AWS Glue

In diesem Abschnitt wird beschrieben, wie Sie einen Spark-Cluster richtig dimensionieren, um das Komprimierungsprogramm von Iceberg auszuführen. Im folgenden Beispiel wird Amazon EMR Serverless verwendet, aber Sie können dieselbe Methode in Amazon EMR on EC2 oder EKS oder in verwenden. AWS Glue

Sie können die Korrelation zwischen Dateigruppen und Spark-Jobs nutzen, um die Cluster-Ressourcen zu planen. Um die Dateigruppen sequentiell zu verarbeiten und dabei die maximale Größe von 100 GB pro Dateigruppe zu berücksichtigen, können Sie die folgenden Spark-Eigenschaften festlegen:

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

Wenn Sie die Komprimierung beschleunigen möchten, können Sie horizontal skalieren, indem Sie die Anzahl der Dateigruppen erhöhen, die parallel komprimiert werden. Sie können Amazon EMR auch mithilfe manueller oder dynamischer Skalierung skalieren.

- Manuelles Skalieren (z. B. um den Faktor 4)
  - `MAX_CONCURRENT_FILE_GROUP_REWRITES`= `4` (unser Faktor)
  - `spark.executor.instances`= `5` (im Beispiel verwendeter Wert) x `4` (unser Faktor) = `20`
  - `spark.dynamicAllocation.enabled` = `FALSE`
- Dynamische Skalierung

- `spark.dynamicAllocation.enabled= TRUE` (Standard, keine Aktion erforderlich)
- MAX_CONCURRENT_FILE_GROUP_REWRITES = `N` (richten Sie diesen Wert an`spark.dynamicAllocation.maxExecutors`, der standardmäßig 100 ist; basierend auf den Executor-Konfigurationen im Beispiel können Sie ihn auf 20 setzen) `N`

Dies sind Richtlinien zur Unterstützung der Clustergröße. Sie sollten jedoch auch die Leistung Ihrer Spark-Jobs überwachen, um die besten Einstellungen für Ihre Workloads zu finden.

Verdichtung mit Amazon Athena ausführen

Athena bietet eine Implementierung des Verdichtungsprogramms von Iceberg als verwaltete Funktion über die OPTIMIZE-Anweisung an. Sie können diese Anweisung verwenden, um die Komprimierung auszuführen, ohne die Infrastruktur auswerten zu müssen.

Diese Anweisung gruppiert kleine Dateien mithilfe des Bin-Packing-Algorithmus zu größeren Dateien und führt gelöschte Dateien mit vorhandenen Datendateien zusammen. Um die Daten mithilfe von hierarchischer Sortierung oder Sortierung in Z-Reihenfolge zu clustern, verwenden Sie Spark auf Amazon EMR oder. AWS Glue

Sie können das Standardverhalten der `OPTIMIZE` Anweisung bei der Tabellenerstellung ändern, indem Sie Tabelleneigenschaften in der `CREATE TABLE` Anweisung übergeben, oder nach der Tabellenerstellung, indem Sie die Anweisung verwenden. `ALTER TABLE` Standardwerte finden Sie in der Athena-Dokumentation.

Empfehlungen für die Ausführung der Komprimierung

Anwendungsfall	Empfehlung
Die Komprimierung von Bin Packing wird nach einem Zeitplan ausgeführt	• Verwenden Sie die `OPTIMIZE` Anweisung in Athena, wenn Sie nicht wissen, wie viele kleine Dateien Ihre Tabelle enthält. Das Athena-Preismodell basiert auf den gescannten Daten. Wenn also keine Dateien komprimiert werden müssen, fallen mit diesen Vorgängen keine Kosten an. Um zu vermeiden, dass es bei Athena-Tabellen zu

Anwendungsfall	Empfehlung
	Timeouts kommt, führen Sie das Programm `OPTIMIZE` auf Basis aus. per-table-partition • Verwenden Sie Amazon EMR oder AWS Glue mit dynamischer Skalierung, wenn Sie erwarten, dass große Mengen kleiner Dateien komprimiert werden.
Die Bin-Packing-Komprimierung wird auf der Grundlage von Ereignissen ausgeführt	• Verwenden Sie Amazon EMR oder AWS Glue mit dynamischer Skalierung, wenn Sie erwarten, dass große Mengen kleiner Dateien komprimiert werden.
Komprimierung zum Sortieren von Daten ausführen	• Verwenden Sie Amazon EMR oder AWS Glue, da das Sortieren ein teurer Vorgang ist und möglicherweise Daten auf die Festplatte übertragen werden müssen.
Führen Sie die Komprimierung aus, um die Daten mithilfe der Sortierung in Z-Reihenfolge zu clustern	• Verwenden Sie Amazon EMR oder AWS Glue, da die Sortierung in Z-Reihenfolge ein sehr teurer Vorgang ist und möglicherweise Daten auf die Festplatte übertragen werden müssen.

Anwendungsfall	Empfehlung
Die Komprimierung wird auf Partitionen ausgeführt, die aufgrund spät eingehender Daten möglicherweise von anderen Anwendungen aktualisiert werden	• Verwenden Sie Amazon EMR oder AWS Glue. Aktivieren Sie die Iceberg-Eigenschaft PARTIAL_PROGRESS_ENABLED. Wenn Sie diese Option verwenden, teilt Iceberg die Komprimierungsausgabe in mehrere Commits auf. Wenn es zu einer Kollision kommt (d. h. wenn die Datendatei während der Komprimierung aktualisiert wird), reduziert diese Einstellung die Kosten für Wiederholungsversuche, indem sie auf den Commit beschränkt wird, der die betroffene Datei enthält. Andernfalls müssen Sie möglicherweise alle Dateien neu komprimieren.
Ausführen der Komprimierung auf kalten Partitionen (Datenpartitionen, die keine aktiven Schreibvorgänge mehr empfangen)	• Verwenden Sie Amazon EMR oder AWS Glue. Geben Sie in der `rewrite_data_files` Prozedur ein `where` Prädikat an, das aktiv geschriebene Partitionen ausschließt. Diese Strategie verhindert Datenkonflikte zwischen Autoren und Verdichtungsjobs und lässt nur Metadatenkonflikte übrig, die Iceberg automatisch lösen kann.

Verwenden von Iceberg-Workloads in Amazon S3

In diesem Abschnitt werden Iceberg-Eigenschaften beschrieben, mit denen Sie die Interaktion von Iceberg mit Amazon S3 optimieren können.

Vermeiden Sie Hot-Partitionierung (HTTP 503-Fehler)

Einige Data Lake-Anwendungen, die auf Amazon S3 ausgeführt werden, verarbeiten Millionen oder Milliarden von Objekten und verarbeiten Petabyte an Daten. Dies kann dazu führen, dass Präfixe ein hohes Datenvolumen empfangen und diese in der Regel anhand von HTTP-503-Fehlern (Service

nicht verfügbar) erkannt werden. Verwenden Sie die folgenden Iceberg-Eigenschaften, um dieses Problem zu vermeiden:

- Auf `hash` oder `range` so einstellen`write.distribution-mode`, dass Iceberg große Dateien schreibt, was zu weniger Amazon S3 S3-Anfragen führt. Dies ist die bevorzugte Konfiguration und sollte für die meisten Fälle geeignet sein.
- Wenn Sie aufgrund eines immensen Datenvolumens in Ihren Workloads weiterhin 503-Fehler feststellen, können Sie dies `true` in `write.object-storage.enabled` Iceberg einstellen. Dadurch wird Iceberg angewiesen, Objektnamen zu hashen und die Last auf mehrere, zufällige Amazon S3 S3-Präfixe zu verteilen.

Weitere Informationen zu diesen Eigenschaften finden Sie unter Eigenschaften schreiben in der Iceberg-Dokumentation.

Verwenden Sie die Wartungsoperationen von Iceberg, um ungenutzte Daten freizugeben

Um Iceberg-Tabellen zu verwalten, können Sie die Iceberg-Kern-API, Iceberg-Clients (wie Spark) oder verwaltete Dienste wie Amazon Athena verwenden. Um alte oder unbenutzte Dateien aus Amazon S3 zu löschen, empfehlen wir, nur Iceberg native APIs zu verwenden, um Schnappschüsse, alte Metadatendateien und verwaiste Dateien zu entfernen.

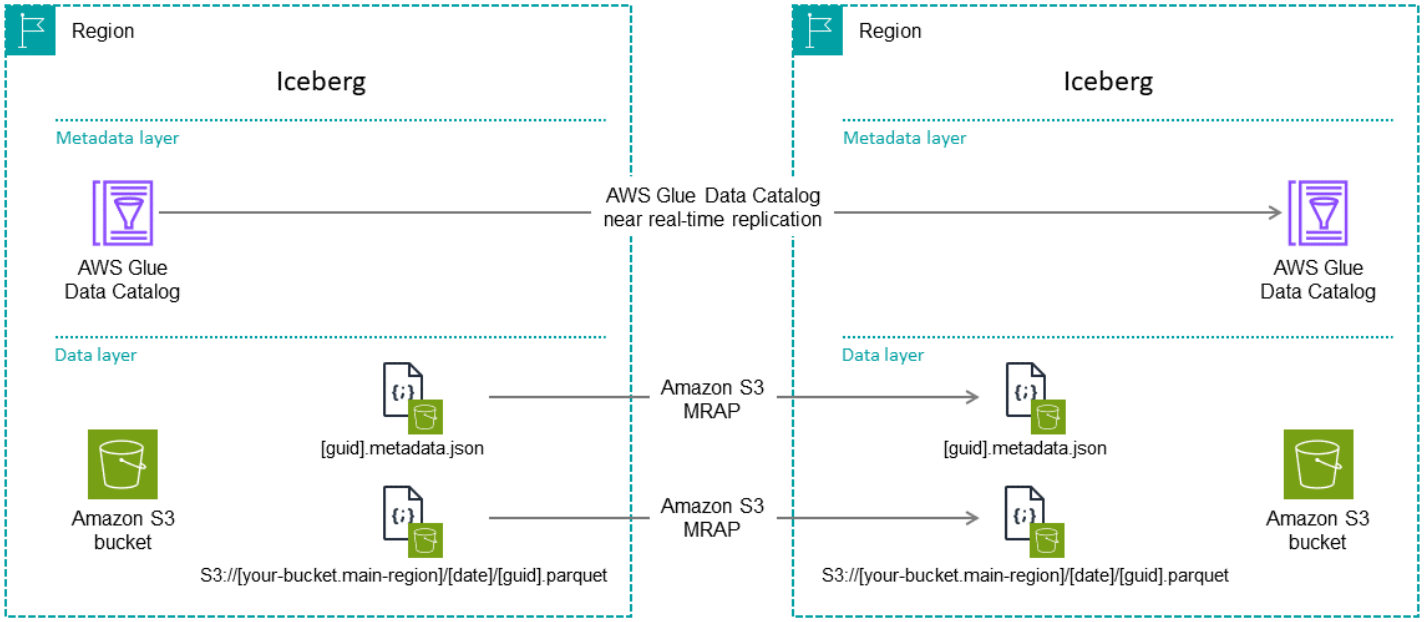
Die Verwendung von Amazon S3 APIs über Boto3, das Amazon S3 S3-SDK oder das AWS Command Line Interface (AWS CLI) oder die Verwendung anderer, nicht von Iceberg stammender Methoden zum Überschreiben oder Entfernen von Amazon S3 S3-Dateien für eine Iceberg-Tabelle führt zu Tabellenbeschädigungen und Abfragefehlern.

Replizieren Sie Daten zwischen AWS-Regionen

Wenn Sie Iceberg-Tabellen in Amazon S3 speichern, können Sie die in Amazon S3 integrierten Funktionen wie Cross-Region Replication (CRR) und Multi-Region Access Points (MRAP) verwenden, um Daten über mehrere zu replizieren. AWS-Regionen MRAP bietet einen globalen Endpunkt für Anwendungen, um auf S3-Buckets zuzugreifen, die sich in mehreren befinden. AWS-Regionen Iceberg unterstützt keine relativen Pfade, aber Sie können MRAP verwenden, um Amazon S3 S3-Operationen durchzuführen, indem Sie Buckets Access Points zuordnen. MRAP lässt sich auch nahtlos in den regionsübergreifenden Replikationsprozess von Amazon S3 integrieren, was zu einer Verzögerung von bis zu 15 Minuten führt. Sie müssen sowohl Daten- als auch Metadatendateien replizieren.

> **Important**
>
> Derzeit funktioniert die Iceberg-Integration mit MRAP nur mit Apache Spark. Wenn Sie ein Failover auf die Sekundärseite durchführen müssen AWS-Region, müssen Sie planen, Benutzeranfragen an eine Spark-SQL-Umgebung (wie Amazon EMR) in der Failover-Region umzuleiten.

Die CRR- und MRAP-Funktionen helfen Ihnen beim Aufbau einer regionsübergreifenden Replikationslösung für Iceberg-Tabellen, wie in der folgenden Abbildung dargestellt.



So richten Sie diese regionsübergreifende Replikationsarchitektur ein:

1. Erstellen Sie Tabellen mithilfe des MRAP-Speicherorts. Dadurch wird sichergestellt, dass Iceberg-Metadatendateien auf den MRAP-Speicherort und nicht auf den physischen Bucket-Speicherort verweisen.
2. Replizieren Sie Iceberg-Dateien mithilfe von Amazon S3 MRAP. MRAP unterstützt Datenreplikation mit einem Service Level Agreement (SLA) von 15 Minuten. Iceberg verhindert, dass Lesevorgänge bei der Replikation zu Inkonsistenzen führen.
3. Stellen Sie die Tabellen in der sekundären Region AWS Glue Data Catalog zur Verfügung. Sie können aus zwei Optionen wählen:
   - Richten Sie eine Pipeline für die Replikation von Iceberg-Tabellenmetadaten mithilfe AWS Glue Data Catalog der Replikation ein. Dieses Tool ist im Replikationsrepository GitHub Glue Catalog

und Lake Formation Permissions verfügbar. Dieser ereignisgesteuerte Mechanismus repliziert Tabellen in der Zielregion auf der Grundlage von Ereignisprotokollen.

- Registrieren Sie die Tabellen in der sekundären Region, wenn ein Failover erforderlich ist. Für diese Option können Sie das vorherige Hilfsprogramm oder die Iceberg-Prozedur register_table verwenden und auf die neueste Datei verweisen. `metadata.json`

Überwachung von Apache Iceberg-Workloads

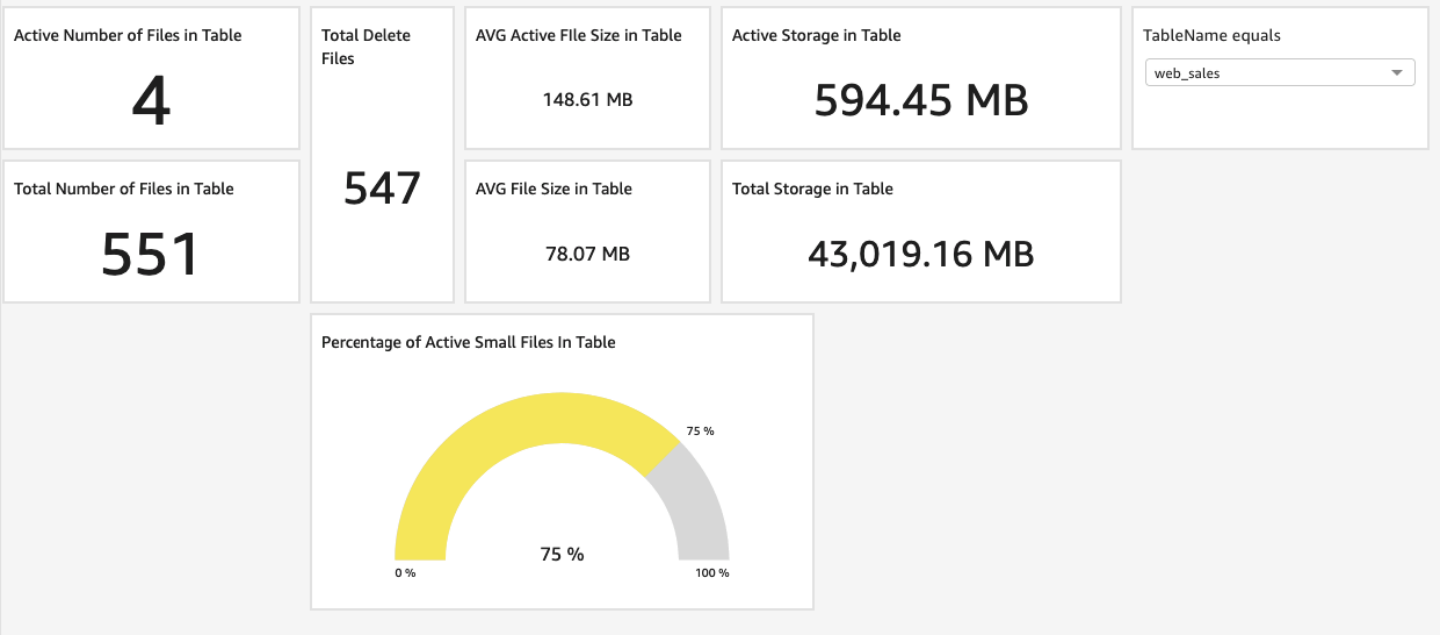
Um Iceberg-Workloads zu überwachen, haben Sie zwei Möglichkeiten: die Analyse von Metadatentabellen oder die Verwendung von Metrik-Reportern. Metrik-Reporter wurden in Iceberg Version 1.2 eingeführt und sind nur für REST- und JDBC-Kataloge verfügbar.

Wenn Sie Iceberg-Tabellen verwenden AWS Glue Data Catalog, können Sie Einblicke in den Zustand Ihrer Iceberg-Tabellen gewinnen, indem Sie zusätzlich zu den von Iceberg bereitgestellten Metadatentabellen eine Überwachung einrichten.

Die Überwachung ist für das Leistungsmanagement und die Fehlerbehebung von entscheidender Bedeutung. Wenn beispielsweise eine Partition in einer Iceberg-Tabelle einen bestimmten Prozentsatz kleiner Dateien erreicht, kann Ihr Workload einen Komprimierungsjob starten, um die Dateien zu größeren Dateien zu konsolidieren. Dadurch wird verhindert, dass Abfragen über ein akzeptables Maß hinaus verlangsamt werden.

Überwachung auf Tabellenebene

Der folgende Bildschirm zeigt ein Dashboard zur Tabellenüberwachung, das in Amazon Quick Sight erstellt wurde. Dieses Dashboard fragt Iceberg-Metadatentabellen mithilfe von Spark SQL ab und erfasst detaillierte Metriken wie die Anzahl der aktiven Dateien und den Gesamtspeicher. Diese Informationen werden dann für betriebliche Zwecke in AWS Glue Tabellen gespeichert. Schließlich wird ein Quick Sight-Dashboard, wie in der folgenden Abbildung gezeigt, mithilfe von Amazon Athena erstellt. Diese Informationen helfen Ihnen dabei, spezifische Probleme in Ihren Systemen zu identifizieren und zu beheben.



Das Quick Sight-Beispiel-Dashboard erfasst die folgenden wichtigen Leistungsindikatoren (KPIs) für eine Iceberg-Tabelle:

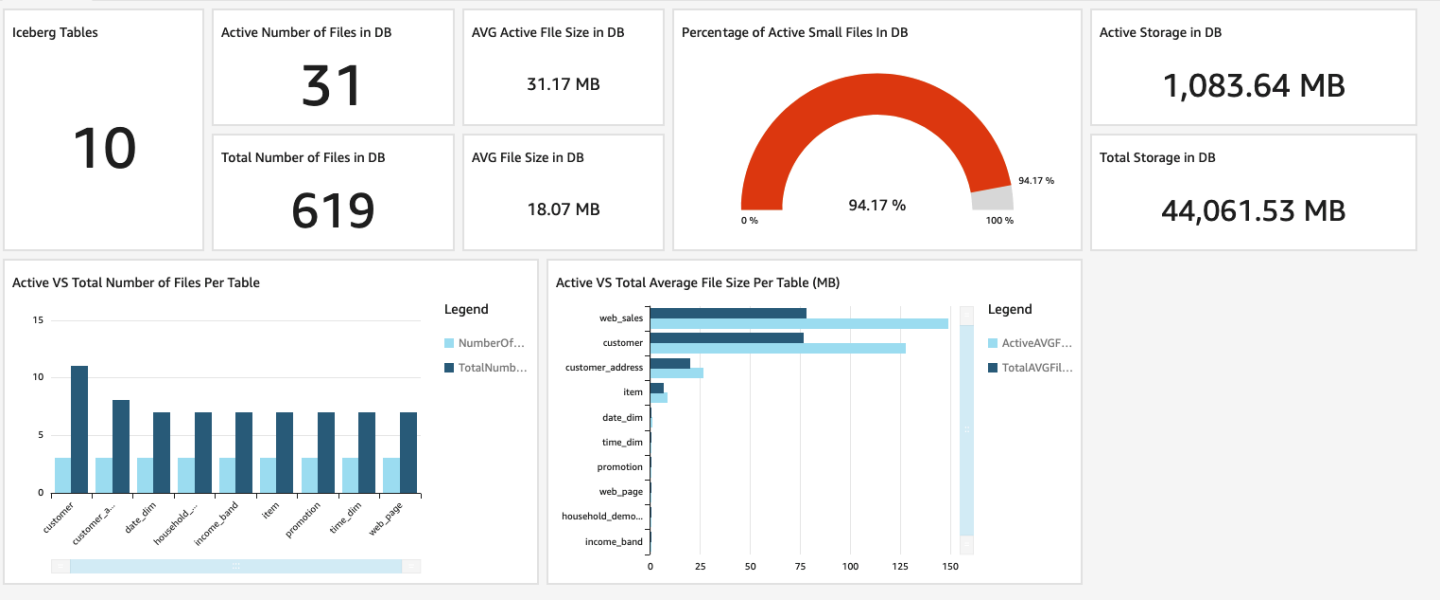
KPI	Beschreibung	Query
Anzahl der Dateien	Die Anzahl der Dateien in der Iceberg-Tabelle (für alle Snapshots)	`select count(*) from <catalog.database.table_name>.all_files`
Anzahl der aktiven Dateien	Die Anzahl der aktiven Dateien im letzten Snapshot der Iceberg-Tabelle	`select count(*) from <catalog.database.table_name>.files`
Durchschnittliche Dateigröße	Die durchschnittliche Dateigröße in Megabyte für alle Dateien in der Iceberg-T abelle	`select avg(file_size_in_bytes)/1000000 from <catalog.database.table_name>.all_files`
Durchschnittliche Größe der aktiven Datei	Die durchschnittliche Dateigröße in Megabyte für die	`select avg(file_size_in_bytes)/1000000`

KPI	Beschreibung	Query
	aktiven Dateien in der Iceberg-Tabelle	`from <catalog.database.table_name>.files`
Prozentsatz kleiner Dateien	Der Prozentsatz der aktiven Dateien, die kleiner als 100 MB sind	`select cast(sum( case when file_size_in_bytes < 100000000 then 1 else 0 end)*100/count(*) as decimal(10,2)) from <catalog.database.table_name>.files`
Gesamtspeichergröße	Die Gesamtgröße aller Dateien in der Tabelle, ausgenommen verwaiste Dateien und Amazon S3 S3-Objektversionen (falls aktiviert)	`select sum(file_size_in_bytes)/1000000 from <catalog.database.table_name>.all_files`
Gesamtgröße des aktiven Speichers	Die Gesamtgröße aller Dateien in den aktuellen Snapshots einer bestimmten Tabelle	`select sum(file_size_in_bytes)/1000000 from <catalog.database.table_name>.files`

Weitere Informationen zum Erstellen von Dashboards finden Sie in der Quick Sight-Dokumentation.

Überwachung auf Datenbankebene

Das folgende Beispiel zeigt ein Überwachungs-Dashboard, das in Quick Sight erstellt wurde, um einen Überblick über die Datenbankebene KPIs für eine Sammlung von Iceberg-Tabellen zu bieten.



Dieses Dashboard sammelt Folgendes: KPIs

KPI	Beschreibung	Query
Anzahl der Dateien	Die Anzahl der Dateien in der Iceberg-Datenbank (für alle Snapshots)	Dieses Dashboard verwendet die Abfragen auf Tabellene bene, die im vorherigen Abschnitt bereitgestellt wurden, und konsolidiert die Ergebnisse.
Anzahl der aktiven Dateien	Die Anzahl der aktiven Dateien in der Iceberg-D atenbank (basierend auf den letzten Snapshots der Iceberg-Tabellen)	
Durchschnittliche Dateigröße	Die durchschnittliche Dateigröße in Megabyte für alle Dateien in der Iceberg-D atenbank	
Durchschnittliche Größe der aktiven Datei	Die durchschnittliche Dateigröße in Megabyte für alle aktiven Dateien in der Iceberg-Datenbank	

KPI	Beschreibung	Query
Prozentsatz kleiner Dateien	Der Prozentsatz der aktiven Dateien, die kleiner als 100 MB sind, in der Iceberg-D atenbank	
Gesamtgröße des Speichers	Die Gesamtgröße aller Dateien in der Datenbank , ausgenommen verwaiste Dateien und Amazon S3 S3-Objektversionen (falls aktiviert)	
Gesamtgröße des aktiven Speichers	Die Gesamtgröße aller Dateien in den aktuellen Snapshots aller Tabellen in der Datenbank	

Präventive Wartung

Durch die Einrichtung der in den vorherigen Abschnitten erläuterten Überwachungsfunktionen können Sie die Tabellenpflege aus einem präventiven und nicht aus einem reaktiven Blickwinkel betrachten. Sie können beispielsweise die Metriken auf Tabellen- und Datenbankebene verwenden, um Aktionen wie die folgenden zu planen:

- Verwenden Sie die Bin-Packing-Komprimierung, um kleine Dateien zu gruppieren, wenn eine Tabelle N kleine Dateien erreicht.
- Verwenden Sie die Bin-Packing-Komprimierung, um gelöschte Dateien zusammenzuführen, wenn eine Tabelle N Löschdateien in einer bestimmten Partition erreicht.
- Entfernen Sie kleine Dateien, die bereits komprimiert wurden, indem Sie Snapshots entfernen, wenn der Gesamtspeicher X-mal höher ist als der aktive Speicher.

Verwaltung und Zugriffskontrolle für Apache Iceberg auf AWS

Apache Iceberg lässt sich integrieren AWS Lake Formation , um die Datenverwaltung zu vereinfachen. Diese Integration ermöglicht es Data Lake-Administratoren, Iceberg-Tabellen Zugriffsberechtigungen auf Zellebene zuzuweisen. Ein Beispiel für das Abfragen von Iceberg-Tabellen mithilfe von Amazon Athena und AWS Lake Formation finden Sie im AWS Blogbeitrag Interagieren Sie mit Apache Iceberg-Tabellen mithilfe von Amazon Athena und kontoübergreifenden, fein abgestuften Berechtigungen mithilfe von. AWS Lake Formation

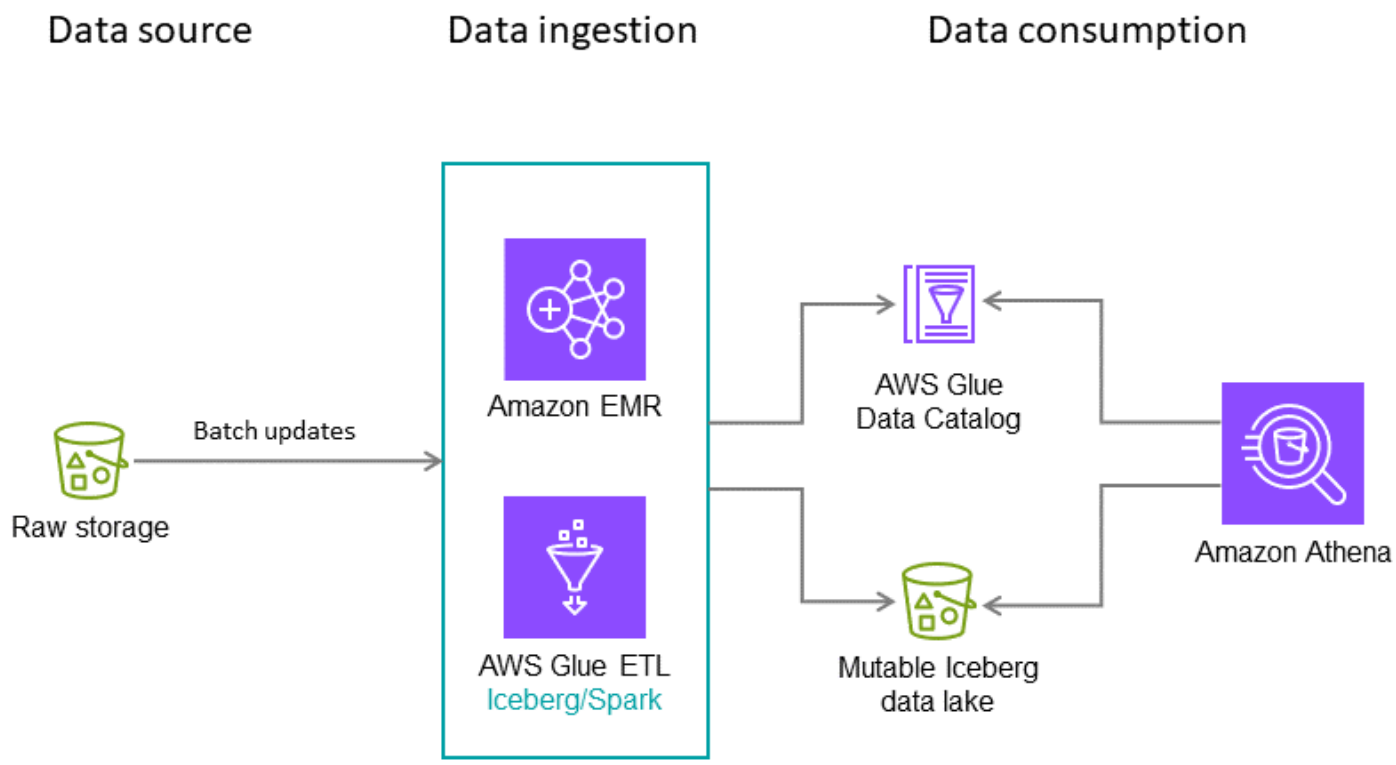
Referenzarchitekturen für Apache Iceberg auf AWS

Dieser Abschnitt enthält Beispiele für die Anwendung von Best Practices in verschiedenen Anwendungsfällen wie Batch-Ingestion und einem Data Lake, der Batch- und Streaming-Datenaufnahme kombiniert.

Nächtliche Batch-Erfassung

Nehmen wir für diesen hypothetischen Anwendungsfall an, dass Ihre Iceberg-Tabelle jede Nacht Kreditkartentransaktionen aufnimmt. Jeder Stapel enthält nur inkrementelle Aktualisierungen, die mit der Zieltabelle zusammengeführt werden müssen. Mehrmals pro Jahr werden vollständige historische Daten empfangen. Für dieses Szenario empfehlen wir die folgende Architektur und Konfigurationen.

Hinweis: Dies ist nur ein Beispiel. Die optimale Konfiguration hängt von Ihren Daten und Anforderungen ab.



Empfehlungen:

- Dateigröße: 128 MB, da Apache Spark-Aufgaben Daten in 128-MB-Blöcken verarbeiten.

- Schreibtyp:. copy-on-write Wie bereits weiter oben in diesem Handbuch beschrieben, trägt dieser Ansatz dazu bei, dass Daten leseoptimiert geschrieben werden.
- Partitionsvariablen:. year/month/day In unserem hypothetischen Anwendungsfall fragen wir am häufigsten aktuelle Daten ab, obwohl wir gelegentlich vollständige Tabellenscans für die Daten der letzten zwei Jahre durchführen. Das Ziel der Partitionierung besteht darin, schnelle Lesevorgänge auf der Grundlage der Anforderungen des Anwendungsfalls zu ermöglichen.
- Sortierreihenfolge: Zeitstempel
- Datenkatalog: AWS Glue Data Catalog

Data Lake, der Batch-Erfassung und Erfassung nahezu in Echtzeit kombiniert

Sie können einen Data Lake auf Amazon S3 bereitstellen, der Batch- und Streaming-Daten über Konten und Regionen hinweg gemeinsam nutzt. Ein Architekturdiagramm und weitere Informationen finden Sie im AWS Blogbeitrag Build a transactional data lake using Apache Iceberg and Cross-account data shares using AWS Lake Formation Amazon Athena. AWS Glue

Ressourcen

- Verwendung des Iceberg-Frameworks in AWS Glue(Dokumentation)AWS Glue
- Iceberg (Amazon EMR-Dokumentation)
- Verwenden von Apache Iceberg-Tabellen (Amazon Athena Athena-Dokumentation)
- Amazon S3 S3-Dokumentation
- Schnelle Dokumentation
- Replikation von Glue Catalog und Lake Formation Permissions (GitHub Repository)
- Dokumentation zu Apache Iceberg
- Apache Spark-Dokumentation

Mitwirkende

Die folgenden Personen haben diesen Leitfaden AWS verfasst, mitverfasst und rezensiert.

Mitwirkende

- Stefano Sandona, Lösungsarchitekt, Big Data
- Imtiaz (Taz) Sayed, Lösungsarchitekt, Technischer Leiter, Analytik
- Shana Schipers, Lösungsarchitektin für Big Data
- Prashant Singh, Softwareentwicklungsingenieur, Amazon EMR
- Arun A K, Lösungsarchitekt für Big Data und ETL
- Francisco Morillo, Lösungsarchitekt, Streaming
- Suthan Phillips, Analytics-Architekt, Amazon EMR
- Sercan Karaoglu, Lösungsarchitekt
- Yonatan Dolan, Analysespezialist
- Guy Bachar, Lösungsarchitekt
- Sofia Zilberman, Lösungsarchitektin, Streaming
- Dan Stair, spezialisierter Lösungsarchitekt
- Sakti Mishra, Lösungsarchitekt
- Ron Ortloff, Hauptproduktmanager, Amazon S3
- David Zhang, Lösungsarchitekt, Analytik

Rezensenten

- Rick Sears, Geschäftsführer, Amazon EMR
- Linda OConnor, Amazon EMR-Spezialistin
- Ian Meyers, Direktor von Amazon EMR
- Vinita Ananth, Direktorin, Amazon EMR-Produktmanagement
- Jason Berkowitz, Produktmanager, AWS Lake Formation
- Mahesh Mishra, Produktmanager, Amazon Redshift
- Vladimir Zlatkin, Manager, Lösungsarchitektur, Big Data
- Karthik Prabhakar, Analytics-Architekt, Amazon EMR

- Vijay Jain, Produktmanager
- Anupriti Warade, Produktmanager, Amazon S3
- Ajit Tandale, Lösungsarchitekt, Daten
- Gwen Chen, Produktmarketing-Managerin
- Noritaka Sekiyama, Hauptarchitekt, Big Data

Dokumentverlauf

In der folgenden Tabelle werden wichtige Änderungen in diesem Leitfaden beschrieben. Um Benachrichtigungen über zukünftige Aktualisierungen zu erhalten, können Sie einen RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Neuer Abschnitt	Es wurden Informationen zur Arbeit mit der Iceberg-Tabellenformatspezifikation Version 3 hinzugefügt.	26. November 2025
Korrektur	Die Informationen zu den `gc.enabled` Einstellungen im Abschnitt Snapshot-Verfahren wurden korrigiert.	24. Oktober 2025
Ergänzungen	Es wurden neue Abschnitte zur Arbeit mit Iceberg-Tabellen mithilfe von Trino und PyIceberg hinzugefügt und der Abschnitt zur Migration von Tabellen nach Iceberg erweitert.	12. August 2025
Aktualisierungen	Die Informationen im gesamten Handbuch wurden erweitert und präzisiert, sodass sie die neuesten Versionen von AWS Glue Amazon EMR und Apache Iceberg widerspiegeln.	14. Juli 2025
Ergänzungen	Es wurde ein neuer Abschnitt über die Arbeit mit Iceberg-T	20. Februar 2025

	abellen mithilfe von Amazon Data Firehose hinzugefügt.	
Erste Veröffentlichung	—	30. April 2024

AWS Glossar zu präskriptiven Leitlinien

Die folgenden Begriffe werden häufig in Strategien, Leitfäden und Mustern von AWS Prescriptive Guidance verwendet. Um Einträge vorzuschlagen, verwenden Sie bitte den Link Feedback geben am Ende des Glossars.

Zahlen

7 Rs

Sieben gängige Migrationsstrategien für die Verlagerung von Anwendungen in die Cloud. Diese Strategien bauen auf den 5 Rs auf, die Gartner 2011 identifiziert hat, und bestehen aus folgenden Elementen:

- Faktorwechsel/Architekturwechsel – Verschieben Sie eine Anwendung und ändern Sie ihre Architektur, indem Sie alle Vorteile cloudnativer Feature nutzen, um Agilität, Leistung und Skalierbarkeit zu verbessern. Dies beinhaltet in der Regel die Portierung des Betriebssystems und der Datenbank. Beispiel: Migrieren Sie Ihre lokale Oracle-Datenbank auf die Amazon Aurora PostgreSQL-kompatible Edition.
- Plattformwechsel (Lift and Reshape) – Verschieben Sie eine Anwendung in die Cloud und führen Sie ein gewisses Maß an Optimierung ein, um die Cloud-Funktionen zu nutzen. Beispiel: Migrieren Sie Ihre lokale Oracle-Datenbank zu Amazon Relational Database Service (Amazon RDS) für Oracle in der. AWS Cloud
- Neukauf (Drop and Shop) – Wechseln Sie zu einem anderen Produkt, indem Sie typischerweise von einer herkömmlichen Lizenz zu einem SaaS-Modell wechseln. Beispiel: Migrieren Sie Ihr CRM-System (Customer Relationship Management) zu Salesforce.com.
- Hostwechsel (Lift and Shift) – Verschieben Sie eine Anwendung in die Cloud, ohne Änderungen vorzunehmen, um die Cloud-Funktionen zu nutzen. Beispiel: Migrieren Sie Ihre lokale Oracle-Datenbank zu Oracle auf einer EC2-Instanz in der. AWS Cloud
- Verschieben (Lift and Shift auf Hypervisor-Ebene) – Verlagern Sie die Infrastruktur in die Cloud, ohne neue Hardware kaufen, Anwendungen umschreiben oder Ihre bestehenden Abläufe ändern zu müssen. Sie migrieren Server von einer lokalen Plattform zu einem Cloud-Dienst für dieselbe Plattform. Beispiel: Migrieren Sie eine Microsoft Hyper-V Anwendung zu AWS.
- Beibehaltung (Wiederaufgreifen) – Bewahren Sie Anwendungen in Ihrer Quellumgebung auf. Dazu können Anwendungen gehören, die einen umfangreichen Faktorwechsel erfordern und

die Sie auf einen späteren Zeitpunkt verschieben möchten, sowie ältere Anwendungen, die Sie beibehalten möchten, da es keine geschäftliche Rechtfertigung für ihre Migration gibt.

- Außerbetriebnahme – Dekommissionierung oder Entfernung von Anwendungen, die in Ihrer Quellumgebung nicht mehr benötigt werden.

A

ABAC

Siehe attributbasierte Zugriffskontrolle.

abstrahierte Dienste

Siehe Managed Services.

ACID

Siehe Atomarität, Konsistenz, Isolierung und Haltbarkeit.

Aktiv-Aktiv-Migration

Eine Datenbankmigrationsmethode, bei der die Quell- und Zieldatenbanken synchron gehalten werden (mithilfe eines bidirektionalen Replikationstools oder dualer Schreibvorgänge) und beide Datenbanken Transaktionen von miteinander verbundenen Anwendungen während der Migration verarbeiten. Diese Methode unterstützt die Migration in kleinen, kontrollierten Batches, anstatt einen einmaligen Cutover zu erfordern. Es ist flexibler, erfordert aber mehr Arbeit als eine aktiv-passive Migration.

Aktiv-Passiv-Migration

Eine Datenbankmigrationsmethode, bei der die Quell- und Zieldatenbanken synchron gehalten werden, aber nur die Quelldatenbank verarbeitet Transaktionen von verbindenden Anwendungen, während Daten in die Zieldatenbank repliziert werden. Die Zieldatenbank akzeptiert während der Migration keine Transaktionen.

Aggregatfunktion

Eine SQL-Funktion, die mit einer Gruppe von Zeilen arbeitet und einen einzelnen Rückgabewert für die Gruppe berechnet. Beispiele für Aggregatfunktionen sind `SUM` und`MAX`.

AI

Siehe künstliche Intelligenz.

AIOps

Siehe Operationen im Bereich künstliche Intelligenz.

Anonymisierung

Der Prozess des dauerhaften Löschens personenbezogener Daten in einem Datensatz. Anonymisierung kann zum Schutz der Privatsphäre beitragen. Anonymisierte Daten gelten nicht mehr als personenbezogene Daten.

Anti-Muster

Eine häufig verwendete Lösung für ein wiederkehrendes Problem, bei dem die Lösung kontraproduktiv, ineffektiv oder weniger wirksam als eine Alternative ist.

Anwendungssteuerung

Ein Sicherheitsansatz, bei dem nur zugelassene Anwendungen verwendet werden können, um ein System vor Schadsoftware zu schützen.

Anwendungsportfolio

Eine Sammlung detaillierter Informationen zu jeder Anwendung, die von einer Organisation verwendet wird, einschließlich der Kosten für die Erstellung und Wartung der Anwendung und ihres Geschäftswerts. Diese Informationen sind entscheidend für den Prozess der Portfoliofindung und -analyse und hilft bei der Identifizierung und Priorisierung der Anwendungen, die migriert, modernisiert und optimiert werden sollen.

künstliche Intelligenz (KI)

Das Gebiet der Datenverarbeitungswissenschaft, das sich der Nutzung von Computertechnologien zur Ausführung kognitiver Funktionen widmet, die typischerweise mit Menschen in Verbindung gebracht werden, wie Lernen, Problemlösen und Erkennen von Mustern. Weitere Informationen finden Sie unter Was ist künstliche Intelligenz?

Operationen mit künstlicher Intelligenz (AIOps)

Der Prozess des Einsatzes von Techniken des Machine Learning zur Lösung betrieblicher Probleme, zur Reduzierung betrieblicher Zwischenfälle und menschlicher Eingriffe sowie zur Steigerung der Servicequalität. Weitere Informationen zur Verwendung in der AWS Migrationsstrategie finden Sie im Operations Integration Guide. AIOps

Asymmetrische Verschlüsselung

Ein Verschlüsselungsalgorithmus, der ein Schlüsselpaar, einen öffentlichen Schlüssel für die Verschlüsselung und einen privaten Schlüssel für die Entschlüsselung verwendet. Sie können den

öffentlichen Schlüssel teilen, da er nicht für die Entschlüsselung verwendet wird. Der Zugriff auf den privaten Schlüssel sollte jedoch stark eingeschränkt sein.

Atomizität, Konsistenz, Isolierung, Haltbarkeit (ACID)

Eine Reihe von Softwareeigenschaften, die die Datenvalidität und betriebliche Zuverlässigkeit einer Datenbank auch bei Fehlern, Stromausfällen oder anderen Problemen gewährleisten.

Attributbasierte Zugriffskontrolle (ABAC)

Die Praxis, detaillierte Berechtigungen auf der Grundlage von Benutzerattributen wie Abteilung, Aufgabenrolle und Teamname zu erstellen. Weitere Informationen finden Sie unter ABAC AWS in der AWS Identity and Access Management (IAM-) Dokumentation.

autoritative Datenquelle

Ein Ort, an dem Sie die primäre Version der Daten speichern, die als die zuverlässigste Informationsquelle angesehen wird. Sie können Daten aus der maßgeblichen Datenquelle an andere Speicherorte kopieren, um die Daten zu verarbeiten oder zu ändern, z. B. zu anonymisieren, zu redigieren oder zu pseudonymisieren.

Availability Zone

Ein bestimmter Standort innerhalb einer AWS-Region , der vor Ausfällen in anderen Availability Zones geschützt ist und kostengünstige Netzwerkkonnektivität mit niedriger Latenz zu anderen Availability Zones in derselben Region bietet.

AWS Framework für die Einführung der Cloud (AWS CAF)

Ein Framework mit Richtlinien und bewährten Verfahren, das Unternehmen bei der Entwicklung eines effizienten und effektiven Plans für die erfolgreiche Umstellung auf die Cloud unterstützt. AWS AWS CAF unterteilt die Leitlinien in sechs Schwerpunktbereiche, die als Perspektiven bezeichnet werden: Unternehmen, Mitarbeiter, Unternehmensführung, Plattform, Sicherheit und Betrieb. Die Perspektiven Geschäft, Mitarbeiter und Unternehmensführung konzentrieren sich auf Geschäftskompetenzen und -prozesse, während sich die Perspektiven Plattform, Sicherheit und Betriebsabläufe auf technische Fähigkeiten und Prozesse konzentrieren. Die Personalperspektive zielt beispielsweise auf Stakeholder ab, die sich mit Personalwesen (HR), Personalfunktionen und Personalmanagement befassen. Aus dieser Perspektive bietet AWS CAF Leitlinien für Personalentwicklung, Schulung und Kommunikation, um das Unternehmen auf eine erfolgreiche Cloud-Einführung vorzubereiten. Weitere Informationen finden Sie auf der AWS -CAF-Webseite und dem AWS -CAF-Whitepaper.

AWS Workload-Qualifizierungsrahmen (AWS WQF)

Ein Tool, das Workloads bei der Datenbankmigration bewertet, Migrationsstrategien empfiehlt und Arbeitsschätzungen bereitstellt. AWS WQF ist in () enthalten. AWS Schema Conversion Tool AWS SCT Es analysiert Datenbankschemas und Codeobjekte, Anwendungscode, Abhängigkeiten und Leistungsmerkmale und stellt Bewertungsberichte bereit.

B

schlechter Bot

Ein Bot, der Einzelpersonen oder Organisationen stören oder ihnen Schaden zufügen soll.

BCP

Siehe Planung der Geschäftskontinuität.

Verhaltensdiagramm

Eine einheitliche, interaktive Ansicht des Ressourcenverhaltens und der Interaktionen im Laufe der Zeit. Sie können ein Verhaltensdiagramm mit Amazon Detective verwenden, um fehlgeschlagene Anmeldeversuche, verdächtige API-Aufrufe und ähnliche Vorgänge zu untersuchen. Weitere Informationen finden Sie unter Daten in einem Verhaltensdiagramm in der Detective-Dokumentation.

Big-Endian-System

Ein System, welches das höchstwertige Byte zuerst speichert. Siehe auch Endianness.

Binäre Klassifikation

Ein Prozess, der ein binäres Ergebnis vorhersagt (eine von zwei möglichen Klassen). Beispielsweise könnte Ihr ML-Modell möglicherweise Probleme wie „Handelt es sich bei dieser E-Mail um Spam oder nicht?“ vorhersagen müssen oder „Ist dieses Produkt ein Buch oder ein Auto?“

Bloom-Filter

Eine probabilistische, speichereffiziente Datenstruktur, mit der getestet wird, ob ein Element Teil einer Menge ist.

Blau/Grün-Bereitstellung

Eine Bereitstellungsstrategie, bei der Sie zwei separate, aber identische Umgebungen erstellen. Sie führen die aktuelle Anwendungsversion in einer Umgebung (blau) und die neue

Anwendungsversion in der anderen Umgebung (grün) aus. Mit dieser Strategie können Sie schnell und mit minimalen Auswirkungen ein Rollback durchführen.

Bot

Eine Softwareanwendung, die automatisierte Aufgaben über das Internet ausführt und menschliche Aktivitäten oder Interaktionen simuliert. Manche Bots sind nützlich oder nützlich, wie z. B. Webcrawler, die Informationen im Internet indexieren. Einige andere Bots, sogenannte bösartige Bots, sollen Einzelpersonen oder Organisationen stören oder ihnen Schaden zufügen.

Botnetz

Netzwerke von Bots, die mit Malware infiziert sind und unter der Kontrolle einer einzigen Partei stehen, die als Bot-Herder oder Bot-Operator bezeichnet wird. Botnetze sind der bekannteste Mechanismus zur Skalierung von Bots und ihrer Wirkung.

branch

Ein containerisierter Bereich eines Code-Repositorys. Der erste Zweig, der in einem Repository erstellt wurde, ist der Hauptzweig. Sie können einen neuen Zweig aus einem vorhandenen Zweig erstellen und dann Feature entwickeln oder Fehler in dem neuen Zweig beheben. Ein Zweig, den Sie erstellen, um ein Feature zu erstellen, wird allgemein als Feature-Zweig bezeichnet. Wenn das Feature zur Veröffentlichung bereit ist, führen Sie den Feature-Zweig wieder mit dem Hauptzweig zusammen. Weitere Informationen finden Sie unter Über Branches (GitHub Dokumentation).

Zugang durch Glasbruch

Unter außergewöhnlichen Umständen und im Rahmen eines genehmigten Verfahrens ist dies eine schnelle Methode für einen Benutzer, auf einen Bereich zuzugreifen AWS-Konto , für den er normalerweise keine Zugriffsrechte besitzt. Weitere Informationen finden Sie unter dem Indikator Implementation break-glass procedures in den AWS Well-Architected-Leitlinien.

Brownfield-Strategie

Die bestehende Infrastruktur in Ihrer Umgebung. Wenn Sie eine Brownfield-Strategie für eine Systemarchitektur anwenden, richten Sie sich bei der Gestaltung der Architektur nach den Einschränkungen der aktuellen Systeme und Infrastruktur. Wenn Sie die bestehende Infrastruktur erweitern, könnten Sie Brownfield- und Greenfield-Strategien mischen.

Puffer-Cache

Der Speicherbereich, in dem die am häufigsten abgerufenen Daten gespeichert werden.

Geschäftsfähigkeit

Was ein Unternehmen tut, um Wert zu generieren (z. B. Vertrieb, Kundenservice oder Marketing). Microservices-Architekturen und Entwicklungsentscheidungen können von den Geschäftskapazitäten beeinflusst werden. Weitere Informationen finden Sie im Abschnitt Organisiert nach Geschäftskapazitäten des Whitepapers Ausführen von containerisierten Microservices in AWS.

Planung der Geschäftskontinuität (BCP)

Ein Plan, der die potenziellen Auswirkungen eines störenden Ereignisses, wie z. B. einer groß angelegten Migration, auf den Betrieb berücksichtigt und es einem Unternehmen ermöglicht, den Betrieb schnell wieder aufzunehmen.

C

CAF

Weitere Informationen finden Sie unter Framework AWS für die Cloud-Einführung.

Bereitstellung auf Kanaren

Die langsame und schrittweise Veröffentlichung einer Version für Endbenutzer. Wenn Sie sich sicher sind, stellen Sie die neue Version bereit und ersetzen die aktuelle Version vollständig.

CCoE

Weitere Informationen finden Sie im Cloud Center of Excellence.

CDC

Siehe Erfassung von Änderungsdaten.

Erfassung von Datenänderungen (CDC)

Der Prozess der Nachverfolgung von Änderungen an einer Datenquelle, z. B. einer Datenbanktabelle, und der Aufzeichnung von Metadaten zu der Änderung. Sie können CDC für verschiedene Zwecke verwenden, z. B. für die Prüfung oder Replikation von Änderungen in einem Zielsystem, um die Synchronisation aufrechtzuerhalten.

Chaos-Technik

Absichtliches Einführen von Ausfällen oder Störungsereignissen, um die Widerstandsfähigkeit eines Systems zu testen. Sie können AWS Fault Injection Service (AWS FIS) verwenden, um Experimente durchzuführen, die Ihre AWS Workloads stress, und deren Reaktion zu bewerten.

CI/CD

Siehe Continuous Integration und Continuous Delivery.

Klassifizierung

Ein Kategorisierungsprozess, der bei der Erstellung von Vorhersagen hilft. ML-Modelle für Klassifikationsprobleme sagen einen diskreten Wert voraus. Diskrete Werte unterscheiden sich immer voneinander. Beispielsweise muss ein Modell möglicherweise auswerten, ob auf einem Bild ein Auto zu sehen ist oder nicht.

clientseitige Verschlüsselung

Lokale Verschlüsselung von Daten, bevor das Ziel sie AWS-Service empfängt.

Cloud-Exzellenzzentrum (CCoE)

Ein multidisziplinäres Team, das die Cloud-Einführung in der gesamten Organisation vorantreibt, einschließlich der Entwicklung bewährter Cloud-Methoden, der Mobilisierung von Ressourcen, der Festlegung von Migrationszeitplänen und der Begleitung der Organisation durch groß angelegte Transformationen. Weitere Informationen finden Sie in den CCoE-Beiträgen im AWS Cloud Enterprise Strategy Blog.

Cloud Computing

Die Cloud-Technologie, die typischerweise für die Ferndatenspeicherung und das IoT-Gerätemanagement verwendet wird. Cloud Computing ist häufig mit Edge-Computing-Technologie verbunden.

Cloud-Betriebsmodell

In einer IT-Organisation das Betriebsmodell, das zum Aufbau, zur Weiterentwicklung und Optimierung einer oder mehrerer Cloud-Umgebungen verwendet wird. Weitere Informationen finden Sie unter Aufbau Ihres Cloud-Betriebsmodells.

Phasen der Einführung der Cloud

Die vier Phasen, die Unternehmen bei der Migration in der Regel durchlaufen AWS Cloud:

- Projekt – Durchführung einiger Cloud-bezogener Projekte zu Machbarkeitsnachweisen und zu Lernzwecken
- Fundament — Tätigen Sie grundlegende Investitionen, um Ihre Cloud-Einführung zu skalieren (z. B. Einrichtung einer landing zone, Definition eines CCo E, Einrichtung eines Betriebsmodells)

- Migration – Migrieren einzelner Anwendungen
- Neuentwicklung – Optimierung von Produkten und Services und Innovation in der Cloud

Diese Phasen wurden von Stephen Orban im Blogbeitrag The Journey Toward Cloud-First & the Stages of Adoption im AWS Cloud Enterprise Strategy-Blog definiert. Informationen darüber, wie sie mit der AWS Migrationsstrategie zusammenhängen, finden Sie im Leitfaden zur Vorbereitung der Migration.

CMDB

Siehe Datenbank für das Konfigurationsmanagement.

Code-Repository

Ein Ort, an dem Quellcode und andere Komponenten wie Dokumentation, Beispiele und Skripts gespeichert und im Rahmen von Versionskontrollprozessen aktualisiert werden. Zu den gängigen Cloud-Repositorys gehören GitHub oderBitbucket Cloud. Jede Version des Codes wird Zweig genannt. In einer Microservice-Struktur ist jedes Repository einer einzelnen Funktionalität gewidmet. Eine einzelne CI/CD-Pipeline kann mehrere Repositorien verwenden.

Kalter Cache

Ein Puffer-Cache, der leer oder nicht gut gefüllt ist oder veraltete oder irrelevante Daten enthält. Dies beeinträchtigt die Leistung, da die Datenbank-Instance aus dem Hauptspeicher oder der Festplatte lesen muss, was langsamer ist als das Lesen aus dem Puffercache.

Kalte Daten

Daten, auf die selten zugegriffen wird und die in der Regel historisch sind. Bei der Abfrage dieser Art von Daten sind langsame Abfragen in der Regel akzeptabel. Durch die Verlagerung dieser Daten auf leistungsschwächere und kostengünstigere Speicherstufen oder -klassen können Kosten gesenkt werden.

Computer Vision (CV)

Ein Bereich der KI, der maschinelles Lernen nutzt, um Informationen aus visuellen Formaten wie digitalen Bildern und Videos zu analysieren und zu extrahieren. Amazon SageMaker AI bietet beispielsweise Bildverarbeitungsalgorithmen für CV.

Drift in der Konfiguration

Bei einer Arbeitslast eine Änderung der Konfiguration gegenüber dem erwarteten Zustand. Dies kann dazu führen, dass der Workload nicht mehr richtlinienkonform wird, und zwar in der Regel schrittweise und unbeabsichtigt.

Verwaltung der Datenbankkonfiguration (CMDB)

Ein Repository, das Informationen über eine Datenbank und ihre IT-Umgebung speichert und verwaltet, inklusive Hardware- und Softwarekomponenten und deren Konfigurationen. In der Regel verwenden Sie Daten aus einer CMDB in der Phase der Portfolioerkennung und -analyse der Migration.

Konformitätspaket

Eine Sammlung von AWS Config Regeln und Abhilfemaßnahmen, die Sie zusammenstellen können, um Ihre Konformitäts- und Sicherheitsprüfungen individuell anzupassen. Mithilfe einer YAML-Vorlage können Sie ein Conformance Pack als einzelne Entität in einer AWS-Konto AND-Region oder unternehmensweit bereitstellen. Weitere Informationen finden Sie in der Dokumentation unter Conformance Packs. AWS Config

Kontinuierliche Bereitstellung und kontinuierliche Integration (CI/CD)

Der Prozess der Automatisierung der Quell-, Build-, Test-, Staging- und Produktionsphasen des Softwareveröffentlichungsprozesses. CI/CD wird allgemein als Pipeline beschrieben. CI/CD kann Ihnen helfen, Prozesse zu automatisieren, die Produktivität zu steigern, die Codequalität zu verbessern und schneller zu liefern. Weitere Informationen finden Sie unter Vorteile der kontinuierlichen Auslieferung. CD kann auch für kontinuierliche Bereitstellung stehen. Weitere Informationen finden Sie unterKontinuierliche Auslieferung im Vergleich zu kontinuierlicher Bereitstellung.

CV

Siehe Computer Vision.

D

Daten im Ruhezustand

Daten, die in Ihrem Netzwerk stationär sind, z. B. Daten, die sich im Speicher befinden.

Datenklassifizierung

Ein Prozess zur Identifizierung und Kategorisierung der Daten in Ihrem Netzwerk auf der Grundlage ihrer Kritikalität und Sensitivität. Sie ist eine wichtige Komponente jeder Strategie für das Management von Cybersecurity-Risiken, da sie Ihnen hilft, die geeigneten Schutz- und Aufbewahrungskontrollen für die Daten zu bestimmen. Die Datenklassifizierung ist ein Bestandteil

der Sicherheitssäule im AWS Well-Architected Framework. Weitere Informationen finden Sie unter Datenklassifizierung.

Datendrift

Eine signifikante Abweichung zwischen den Produktionsdaten und den Daten, die zum Trainieren eines ML-Modells verwendet wurden, oder eine signifikante Änderung der Eingabedaten im Laufe der Zeit. Datendrift kann die Gesamtqualität, Genauigkeit und Fairness von ML-Modellvorhersagen beeinträchtigen.

Daten während der Übertragung

Daten, die sich aktiv durch Ihr Netzwerk bewegen, z. B. zwischen Netzwerkressourcen.

Datennetz

Ein architektonisches Framework, das verteilte, dezentrale Dateneigentum mit zentraler Verwaltung und Steuerung ermöglicht.

Datenminimierung

Das Prinzip, nur die Daten zu sammeln und zu verarbeiten, die unbedingt erforderlich sind. Durch Datenminimierung im AWS Cloud können Datenschutzrisiken, Kosten und der CO2-Fußabdruck Ihrer Analysen reduziert werden.

Datenperimeter

Eine Reihe präventiver Schutzmaßnahmen in Ihrer AWS Umgebung, die sicherstellen, dass nur vertrauenswürdige Identitäten auf vertrauenswürdige Ressourcen von erwarteten Netzwerken zugreifen. Weitere Informationen finden Sie unter Aufbau eines Datenperimeters auf. AWS

Vorverarbeitung der Daten

Rohdaten in ein Format umzuwandeln, das von Ihrem ML-Modell problemlos verarbeitet werden kann. Die Vorverarbeitung von Daten kann bedeuten, dass bestimmte Spalten oder Zeilen entfernt und fehlende, inkonsistente oder doppelte Werte behoben werden.

Herkunft der Daten

Der Prozess der Nachverfolgung des Ursprungs und der Geschichte von Daten während ihres gesamten Lebenszyklus, z. B. wie die Daten generiert, übertragen und gespeichert wurden.

betroffene Person

Eine Person, deren Daten gesammelt und verarbeitet werden.

Data Warehouse

Ein Datenverwaltungssystem, das Business Intelligence wie Analysen unterstützt. Data Warehouses enthalten in der Regel große Mengen historischer Daten und werden in der Regel für Abfragen und Analysen verwendet.

Datenbankdefinitionssprache (DDL)

Anweisungen oder Befehle zum Erstellen oder Ändern der Struktur von Tabellen und Objekten in einer Datenbank.

Datenbankmanipulationssprache (DML)

Anweisungen oder Befehle zum Ändern (Einfügen, Aktualisieren und Löschen) von Informationen in einer Datenbank.

DDL

Siehe Datenbankdefinitionssprache.

Deep-Ensemble

Mehrere Deep-Learning-Modelle zur Vorhersage kombinieren. Sie können Deep-Ensembles verwenden, um eine genauere Vorhersage zu erhalten oder um die Unsicherheit von Vorhersagen abzuschätzen.

Deep Learning

Ein ML-Teilbereich, der mehrere Schichten künstlicher neuronaler Netzwerke verwendet, um die Zuordnung zwischen Eingabedaten und Zielvariablen von Interesse zu ermitteln.

defense-in-depth

Ein Ansatz zur Informationssicherheit, bei dem eine Reihe von Sicherheitsmechanismen und -kontrollen sorgfältig in einem Computernetzwerk verteilt werden, um die Vertraulichkeit, Integrität und Verfügbarkeit des Netzwerks und der darin enthaltenen Daten zu schützen. Wenn Sie diese Strategie anwenden AWS, fügen Sie mehrere Steuerelemente auf verschiedenen Ebenen der AWS Organizations Struktur hinzu, um die Ressourcen zu schützen. Ein defense-in-depth Ansatz könnte beispielsweise Multi-Faktor-Authentifizierung, Netzwerksegmentierung und Verschlüsselung kombinieren.

delegierter Administrator

In AWS Organizations kann ein kompatibler Dienst ein AWS Mitgliedskonto registrieren, um die Konten der Organisation und die Berechtigungen für diesen Dienst zu verwalten. Dieses Konto

wird als delegierter Administrator für diesen Service bezeichnet. Weitere Informationen und eine Liste kompatibler Services finden Sie unter Services, die mit AWS Organizations funktionieren in der AWS Organizations -Dokumentation.

Einsatz

Der Prozess, bei dem eine Anwendung, neue Feature oder Codekorrekturen in der Zielumgebung verfügbar gemacht werden. Die Bereitstellung umfasst das Implementieren von Änderungen an einer Codebasis und das anschließende Erstellen und Ausführen dieser Codebasis in den Anwendungsumgebungen.

Entwicklungsumgebung

Siehe Umgebung.

Detektivische Kontrolle

Eine Sicherheitskontrolle, die darauf ausgelegt ist, ein Ereignis zu erkennen, zu protokollieren und zu warnen, nachdem ein Ereignis eingetreten ist. Diese Kontrollen stellen eine zweite Verteidigungslinie dar und warnen Sie vor Sicherheitsereignissen, bei denen die vorhandenen präventiven Kontrollen umgangen wurden. Weitere Informationen finden Sie unter Detektivische Kontrolle in Implementierung von Sicherheitskontrollen in AWS.

Abbildung des Wertstroms in der Entwicklung (DVSM)

Ein Prozess zur Identifizierung und Priorisierung von Einschränkungen, die sich negativ auf Geschwindigkeit und Qualität im Lebenszyklus der Softwareentwicklung auswirken. DVSM erweitert den Prozess der Wertstromanalyse, der ursprünglich für Lean-Manufacturing-Praktiken konzipiert wurde. Es konzentriert sich auf die Schritte und Teams, die erforderlich sind, um durch den Softwareentwicklungsprozess Mehrwert zu schaffen und zu steigern.

digitaler Zwilling

Eine virtuelle Darstellung eines realen Systems, z. B. eines Gebäudes, einer Fabrik, einer Industrieanlage oder einer Produktionslinie. Digitale Zwillinge unterstützen vorausschauende Wartung, Fernüberwachung und Produktionsoptimierung.

Maßtabelle

In einem Sternschema eine kleinere Tabelle, die Datenattribute zu quantitativen Daten in einer Faktentabelle enthält. Bei Attributen von Dimensionstabellen handelt es sich in der Regel um Textfelder oder diskrete Zahlen, die sich wie Text verhalten. Diese Attribute werden häufig zum Einschränken von Abfragen, zum Filtern und zur Kennzeichnung von Ergebnismengen verwendet.

Katastrophe

Ein Ereignis, das verhindert, dass ein Workload oder ein System seine Geschäftsziele an seinem primären Einsatzort erfüllt. Diese Ereignisse können Naturkatastrophen, technische Ausfälle oder das Ergebnis menschlichen Handelns sein, wie z. B. unbeabsichtigte Fehlkonfigurationen oder ein Malware-Angriff.

Notfallwiederherstellung (DR)

Die Strategie und der Prozess, mit denen Sie Ausfallzeiten und Datenverluste aufgrund einer Katastrophe minimieren. Weitere Informationen finden Sie unter Disaster Recovery von Workloads unter AWS: Wiederherstellung in der Cloud im AWS Well-Architected Framework.

DML

Siehe Sprache zur Datenbankmanipulation.

Domainorientiertes Design

Ein Ansatz zur Entwicklung eines komplexen Softwaresystems, bei dem seine Komponenten mit sich entwickelnden Domains oder Kerngeschäftszielen verknüpft werden, denen jede Komponente dient. Dieses Konzept wurde von Eric Evans in seinem Buch Domaingesteuertes Design: Bewältigen der Komplexität im Herzen der Software (Boston: Addison-Wesley Professional, 2003) vorgestellt. Informationen darüber, wie Sie domaingesteuertes Design mit dem Strangler-Fig-Muster verwenden können, finden Sie unter Schrittweises Modernisieren älterer Microsoft ASP.NET (ASMX)-Webservices mithilfe von Containern und Amazon API Gateway.

DR

Siehe Disaster Recovery.

Erkennung von Driften

Verfolgung von Abweichungen von einer Basiskonfiguration. Sie können es beispielsweise verwenden, AWS CloudFormation um Abweichungen bei den Systemressourcen zu erkennen, oder Sie können AWS Control Tower damit Änderungen in Ihrer landing zone erkennen, die sich auf die Einhaltung von Governance-Anforderungen auswirken könnten.

DVSM

Siehe Abbildung des Wertstroms in der Entwicklung.

E

EDA

Siehe explorative Datenanalyse.

EDI

Siehe elektronischer Datenaustausch.

Edge-Computing

Die Technologie, die die Rechenleistung für intelligente Geräte an den Rändern eines IoT-Netzwerks erhöht. Im Vergleich zu Cloud Computing kann Edge Computing die Kommunikationslatenz reduzieren und die Reaktionszeit verbessern.

elektronischer Datenaustausch (EDI)

Der automatisierte Austausch von Geschäftsdokumenten zwischen Organisationen. Weitere Informationen finden Sie unter Was ist elektronischer Datenaustausch.

Verschlüsselung

Ein Rechenprozess, der Klartextdaten, die für Menschen lesbar sind, in Chiffretext umwandelt.

Verschlüsselungsschlüssel

Eine kryptografische Zeichenfolge aus zufälligen Bits, die von einem Verschlüsselungsalgorithmus generiert wird. Schlüssel können unterschiedlich lang sein, und jeder Schlüssel ist so konzipiert, dass er unvorhersehbar und einzigartig ist.

Endianismus

Die Reihenfolge, in der Bytes im Computerspeicher gespeichert werden. Big-Endian-Systeme speichern das höchstwertige Byte zuerst. Little-Endian-Systeme speichern das niedrigwertigste Byte zuerst.

Endpunkt

Siehe Service-Endpunkt.

Endpunkt-Services

Ein Service, den Sie in einer Virtual Private Cloud (VPC) hosten können, um ihn mit anderen Benutzern zu teilen. Sie können einen Endpunktdienst mit anderen AWS-Konten oder AWS Identity and Access Management (IAM AWS PrivateLink -) Prinzipalen erstellen und diesen

Berechtigungen gewähren. Diese Konten oder Prinzipale können sich privat mit Ihrem Endpunktservice verbinden, indem sie Schnittstellen-VPC-Endpunkte erstellen. Weitere Informationen finden Sie unter Einen Endpunkt-Service erstellen in der Amazon Virtual Private Cloud (Amazon VPC)-Dokumentation.

Unternehmensressourcenplanung (ERP)

Ein System, das wichtige Geschäftsprozesse (wie Buchhaltung, MES und Projektmanagement) für ein Unternehmen automatisiert und verwaltet.

Envelope-Verschlüsselung

Der Prozess der Verschlüsselung eines Verschlüsselungsschlüssels mit einem anderen Verschlüsselungsschlüssel. Weitere Informationen finden Sie unter Envelope-Verschlüsselung in der AWS Key Management Service (AWS KMS) -Dokumentation.

Umgebung

Eine Instance einer laufenden Anwendung. Die folgenden Arten von Umgebungen sind beim Cloud-Computing üblich:

- Entwicklungsumgebung – Eine Instance einer laufenden Anwendung, die nur dem Kernteam zur Verfügung steht, das für die Wartung der Anwendung verantwortlich ist. Entwicklungsumgebungen werden verwendet, um Änderungen zu testen, bevor sie in höhere Umgebungen übertragen werden. Diese Art von Umgebung wird manchmal als Testumgebung bezeichnet.
- Niedrigere Umgebungen – Alle Entwicklungsumgebungen für eine Anwendung, z. B. solche, die für erste Builds und Tests verwendet wurden.
- Produktionsumgebung – Eine Instance einer laufenden Anwendung, auf die Endbenutzer zugreifen können. In einer CI/CD Pipeline ist die Produktionsumgebung die letzte Bereitstellungsumgebung.
- Höhere Umgebungen – Alle Umgebungen, auf die auch andere Benutzer als das Kernentwicklungsteam zugreifen können. Dies kann eine Produktionsumgebung, Vorproduktionsumgebungen und Umgebungen für Benutzerakzeptanztests umfassen.

Epics

In der agilen Methodik sind dies funktionale Kategorien, die Ihnen helfen, Ihre Arbeit zu organisieren und zu priorisieren. Epics bieten eine allgemeine Beschreibung der Anforderungen und Implementierungsaufgaben. Zu den Sicherheitsepen AWS von CAF gehören beispielsweise Identitäts- und Zugriffsmanagement, Detektivkontrollen, Infrastruktursicherheit, Datenschutz und

Reaktion auf Vorfälle. Weitere Informationen zu Epics in der AWS -Migrationsstrategie finden Sie im Leitfaden zur Programm-Implementierung.

ERP

Siehe Enterprise Resource Planning.

Explorative Datenanalyse (EDA)

Der Prozess der Analyse eines Datensatzes, um seine Hauptmerkmale zu verstehen. Sie sammeln oder aggregieren Daten und führen dann erste Untersuchungen durch, um Muster zu finden, Anomalien zu erkennen und Annahmen zu überprüfen. EDA wird durchgeführt, indem zusammenfassende Statistiken berechnet und Datenvisualisierungen erstellt werden.

F

Faktentabelle

Die zentrale Tabelle in einem Sternschema. Sie speichert quantitative Daten über den Geschäftsbetrieb. In der Regel enthält eine Faktentabelle zwei Arten von Spalten: Spalten, die Kennzahlen enthalten, und Spalten, die einen Fremdschlüssel für eine Dimensionstabelle enthalten.

schnell scheitern

Eine Philosophie, die häufige und inkrementelle Tests verwendet, um den Entwicklungslebenszyklus zu verkürzen. Dies ist ein wichtiger Bestandteil eines agilen Ansatzes.

Grenze zur Fehlerisolierung

Dabei handelt es sich um eine Grenze AWS Cloud, z. B. eine Availability Zone AWS-Region, eine Steuerungsebene oder eine Datenebene, die die Auswirkungen eines Fehlers begrenzt und die Widerstandsfähigkeit von Workloads verbessert. Weitere Informationen finden Sie unter Grenzen zur AWS Fehlerisolierung.

Feature-Zweig

Siehe Zweig.

Features

Die Eingabedaten, die Sie verwenden, um eine Vorhersage zu treffen. In einem Fertigungskontext könnten Feature beispielsweise Bilder sein, die regelmäßig von der Fertigungslinie aus aufgenommen werden.

Bedeutung der Feature

Wie wichtig ein Feature für die Vorhersagen eines Modells ist. Dies wird in der Regel als numerischer Wert ausgedrückt, der mit verschiedenen Techniken wie Shapley Additive Explanations (SHAP) und integrierten Gradienten berechnet werden kann. Weitere Informationen finden Sie unter Interpretierbarkeit von Modellen für maschinelles Lernen mit AWS.

Featuretransformation

Daten für den ML-Prozess optimieren, einschließlich der Anreicherung von Daten mit zusätzlichen Quellen, der Skalierung von Werten oder der Extraktion mehrerer Informationssätze aus einem einzigen Datenfeld. Das ermöglicht dem ML-Modell, von den Daten profitieren. Wenn Sie beispielsweise das Datum „27.05.2021 00:15:37“ in „2021“, „Mai“, „Donnerstag“ und „15“ aufschlüsseln, können Sie dem Lernalgorithmus helfen, nuancierte Muster zu erlernen, die mit verschiedenen Datenkomponenten verknüpft sind.

Eingabeaufforderung mit wenigen Klicks

Bereitstellung einer kleinen Anzahl von Beispielen, die die Aufgabe und das gewünschte Ergebnis veranschaulichen, bevor das LLM aufgefordert wird, eine ähnliche Aufgabe auszuführen. Bei dieser Technik handelt es sich um eine Anwendung des kontextbezogenen Lernens, bei der Modelle anhand von Beispielen (Aufnahmen) lernen, die in Eingabeaufforderungen eingebettet sind. Bei Aufgaben, die spezifische Formatierungs-, Argumentations- oder Fachkenntnisse erfordern, kann die Eingabeaufforderung mit wenigen Handgriffen effektiv sein. Siehe auch Zero-Shot Prompting.

FGAC

Siehe detaillierte Zugriffskontrolle.

Feinkörnige Zugriffskontrolle (FGAC)

Die Verwendung mehrerer Bedingungen, um eine Zugriffsanfrage zuzulassen oder abzulehnen.

Flash-Cut-Migration

Eine Datenbankmigrationsmethode, bei der eine kontinuierliche Datenreplikation durch Erfassung von Änderungsdaten verwendet wird, um Daten in kürzester Zeit zu migrieren, anstatt einen schrittweisen Ansatz zu verwenden. Ziel ist es, Ausfallzeiten auf ein Minimum zu beschränken.

FM

Siehe Fundamentmodell.

Fundamentmodell (FM)

Ein großes neuronales Deep-Learning-Netzwerk, das mit riesigen Datensätzen generalisierter und unbeschrifteter Daten trainiert wurde. FMs sind in der Lage, eine Vielzahl allgemeiner Aufgaben zu erfüllen, z. B. Sprache zu verstehen, Text und Bilder zu generieren und Konversationen in natürlicher Sprache zu führen. Weitere Informationen finden Sie unter Was sind Foundation-Modelle.

G

Generative KI

Eine Untergruppe von KI-Modellen, die mit großen Datenmengen trainiert wurden und mit einer einfachen Textaufforderung neue Inhalte und Artefakte wie Bilder, Videos, Text und Audio erstellen können. Weitere Informationen finden Sie unter Was ist Generative KI.

Geoblocking

Siehe geografische Einschränkungen.

Geografische Einschränkungen (Geoblocking)

Bei Amazon eine Option CloudFront, um zu verhindern, dass Benutzer in bestimmten Ländern auf Inhaltsverteilungen zugreifen. Sie können eine Zulassungsliste oder eine Sperrliste verwenden, um zugelassene und gesperrte Länder anzugeben. Weitere Informationen finden Sie in der Dokumentation unter Beschränkung der geografischen Verteilung Ihrer Inhalte. CloudFront

Gitflow-Workflow

Ein Ansatz, bei dem niedrigere und höhere Umgebungen unterschiedliche Zweige in einem Quellcode-Repository verwenden. Der Gitflow-Workflow gilt als veraltet, und der Trunk-basierte Workflow ist der moderne, bevorzugte Ansatz.

goldenes Bild

Ein Snapshot eines Systems oder einer Software, der als Vorlage für die Bereitstellung neuer Instanzen dieses Systems oder dieser Software verwendet wird. In der Fertigung kann ein Golden Image beispielsweise zur Bereitstellung von Software auf mehreren Geräten verwendet werden und trägt zur Verbesserung der Geschwindigkeit, Skalierbarkeit und Produktivität bei der Geräteherstellung bei.

Greenfield-Strategie

Das Fehlen vorhandener Infrastruktur in einer neuen Umgebung. Bei der Einführung einer Neuausrichtung einer Systemarchitektur können Sie alle neuen Technologien ohne Einschränkung der Kompatibilität mit der vorhandenen Infrastruktur auswählen, auch bekannt als Brownfield. Wenn Sie die bestehende Infrastruktur erweitern, könnten Sie Brownfield- und Greenfield-Strategien mischen.

Integritätsschutz

Eine allgemeine Regel, die dazu beiträgt, Ressourcen, Richtlinien und die Einhaltung von Vorschriften in allen Unternehmenseinheiten zu regeln (OUs). Präventiver Integritätsschutz setzt Richtlinien durch, um die Einhaltung von Standards zu gewährleisten. Sie werden mithilfe von Service-Kontrollrichtlinien und IAM-Berechtigungsgrenzen implementiert. Detektivischer Integritätsschutz erkennt Richtlinienverstöße und Compliance-Probleme und generiert Warnmeldungen zur Abhilfe. Sie werden mithilfe von AWS Config, AWS Security Hub CSPM, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector und benutzerdefinierten AWS Lambda Prüfungen implementiert.

H

HEKTAR

Siehe Hochverfügbarkeit.

Heterogene Datenbankmigration

Migrieren Sie Ihre Quelldatenbank in eine Zieldatenbank, die eine andere Datenbank-Engine verwendet (z. B. Oracle zu Amazon Aurora). Eine heterogene Migration ist in der Regel Teil einer Neuarchitektur, und die Konvertierung des Schemas kann eine komplexe Aufgabe sein. AWS bietet AWS SCT, welches bei Schemakonvertierungen hilft.

hohe Verfügbarkeit (HA)

Die Fähigkeit eines Workloads, im Falle von Herausforderungen oder Katastrophen kontinuierlich und ohne Eingreifen zu arbeiten. HA-Systeme sind so konzipiert, dass sie automatisch ein Failover durchführen, gleichbleibend hohe Leistung bieten und unterschiedliche Lasten und Ausfälle mit minimalen Leistungseinbußen bewältigen.

historische Modernisierung

Ein Ansatz zur Modernisierung und Aufrüstung von Betriebstechnologiesystemen (OT), um den Bedürfnissen der Fertigungsindustrie besser gerecht zu werden. Ein Historian ist eine Art von Datenbank, die verwendet wird, um Daten aus verschiedenen Quellen in einer Fabrik zu sammeln und zu speichern.

Daten zurückhalten

Ein Teil historischer, beschrifteter Daten, der aus einem Datensatz zurückgehalten wird, der zum Trainieren eines Modells für maschinelles Lernen verwendet wird. Sie können Holdout-Daten verwenden, um die Modellleistung zu bewerten, indem Sie die Modellvorhersagen mit den Holdout-Daten vergleichen.

Homogene Datenbankmigration

Migrieren Sie Ihre Quelldatenbank zu einer Zieldatenbank, die dieselbe Datenbank-Engine verwendet (z. B. Microsoft SQL Server zu Amazon RDS für SQL Server). Eine homogene Migration ist in der Regel Teil eines Hostwechsels oder eines Plattformwechsels. Sie können native Datenbankserviceprogramme verwenden, um das Schema zu migrieren.

heiße Daten

Daten, auf die häufig zugegriffen wird, z. B. Echtzeitdaten oder aktuelle Translationsdaten. Für diese Daten ist in der Regel eine leistungsstarke Speicherebene oder -klasse erforderlich, um schnelle Abfrageantworten zu ermöglichen.

Hotfix

Eine dringende Lösung für ein kritisches Problem in einer Produktionsumgebung. Aufgrund seiner Dringlichkeit wird ein Hotfix normalerweise außerhalb des typischen DevOps Release-Workflows erstellt.

Hypercare-Phase

Unmittelbar nach dem Cutover, der Zeitraum, in dem ein Migrationsteam die migrierten Anwendungen in der Cloud verwaltet und überwacht, um etwaige Probleme zu beheben. In der Regel dauert dieser Zeitraum 1–4 Tage. Am Ende der Hypercare-Phase überträgt das Migrationsteam in der Regel die Verantwortung für die Anwendungen an das Cloud-Betriebsteam.

I

IaC

Sehen Sie Infrastruktur als Code.

Identitätsbasierte Richtlinie

Eine Richtlinie, die einem oder mehreren IAM-Prinzipalen zugeordnet ist und deren Berechtigungen innerhalb der AWS Cloud Umgebung definiert.

Leerlaufanwendung

Eine Anwendung mit einer durchschnittlichen CPU- und Arbeitsspeicherauslastung zwischen 5 und 20 Prozent über einen Zeitraum von 90 Tagen. In einem Migrationsprojekt ist es üblich, diese Anwendungen außer Betrieb zu nehmen oder sie On-Premises beizubehalten.

IIoT

Siehe Industrielles Internet der Dinge.

unveränderliche Infrastruktur

Ein Modell, das eine neue Infrastruktur für Produktionsworkloads bereitstellt, anstatt die bestehende Infrastruktur zu aktualisieren, zu patchen oder zu modifizieren. Unveränderliche Infrastrukturen sind von Natur aus konsistenter, zuverlässiger und vorhersehbarer als veränderliche Infrastrukturen. Weitere Informationen finden Sie in der Best Practice Deploy using immutable infrastructure im AWS Well-Architected Framework.

Eingehende (ingress) VPC

In einer Architektur AWS mit mehreren Konten ist dies eine VPC, die Netzwerkverbindungen von außerhalb einer Anwendung akzeptiert, überprüft und weiterleitet. Die AWS Security Reference Architecture empfiehlt, Ihr Netzwerkkonto mit eingehendem und ausgehendem Datenverkehr und Inspektion einzurichten, VPCs um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet im weiteren Sinne zu schützen.

Inkrementelle Migration

Eine Cutover-Strategie, bei der Sie Ihre Anwendung in kleinen Teilen migrieren, anstatt eine einziges vollständiges Cutover durchzuführen. Beispielsweise könnten Sie zunächst nur einige Microservices oder Benutzer auf das neue System umstellen. Nachdem Sie sich vergewissert haben, dass alles ordnungsgemäß funktioniert, können Sie weitere Microservices oder Benutzer

schrittweise verschieben, bis Sie Ihr Legacy-System außer Betrieb nehmen können. Diese Strategie reduziert die mit großen Migrationen verbundenen Risiken.

Industrie 4.0

Ein Begriff, der 2016 von Klaus Schwab eingeführt wurde und sich auf die Modernisierung von Fertigungsprozessen durch Fortschritte in den Bereichen Konnektivität, Echtzeitdaten, Automatisierung, Analytik und KI/ML bezieht.

Infrastruktur

Alle Ressourcen und Komponenten, die in der Umgebung einer Anwendung enthalten sind.

Infrastructure as Code (IaC)

Der Prozess der Bereitstellung und Verwaltung der Infrastruktur einer Anwendung mithilfe einer Reihe von Konfigurationsdateien. IaC soll Ihnen helfen, das Infrastrukturmanagement zu zentralisieren, Ressourcen zu standardisieren und schnell zu skalieren, sodass neue Umgebungen wiederholbar, zuverlässig und konsistent sind.

industrielles Internet der Dinge (T) IIo

Einsatz von mit dem Internet verbundenen Sensoren und Geräten in Industriesektoren wie Fertigung, Energie, Automobilindustrie, Gesundheitswesen, Biowissenschaften und Landwirtschaft. Weitere Informationen finden Sie unter Aufbau einer digitalen Transformationsstrategie für das industrielle Internet der Dinge (IIoT).

Inspektions-VPC

In einer Architektur AWS mit mehreren Konten eine zentralisierte VPC, die Inspektionen des Netzwerkverkehrs zwischen VPCs (in demselben oder unterschiedlichen AWS-Regionen), dem Internet und lokalen Netzwerken verwaltet. In der AWS Security Reference Architecture wird empfohlen, Ihr Netzwerkkonto mit eingehendem und ausgehendem Datenverkehr sowie Inspektionen einzurichten, VPCs um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet im weiteren Sinne zu schützen.

Internet of Things (IoT)

Das Netzwerk verbundener physischer Objekte mit eingebetteten Sensoren oder Prozessoren, das über das Internet oder über ein lokales Kommunikationsnetzwerk mit anderen Geräten und Systemen kommuniziert. Weitere Informationen finden Sie unter Was ist IoT?

Interpretierbarkeit

Ein Merkmal eines Modells für Machine Learning, das beschreibt, inwieweit ein Mensch verstehen kann, wie die Vorhersagen des Modells von seinen Eingaben abhängen. Weitere Informationen finden Sie unter Interpretierbarkeit des Modells für maschinelles Lernen mit. AWS

IoT

Siehe Internet der Dinge.

IT information library (ITIL, IT-Informationsbibliothek)

Eine Reihe von bewährten Methoden für die Bereitstellung von IT-Services und die Abstimmung dieser Services auf die Geschäftsanforderungen. ITIL bietet die Grundlage für ITSM.

T service management (ITSM, IT-Servicemanagement)

Aktivitäten im Zusammenhang mit der Gestaltung, Implementierung, Verwaltung und Unterstützung von IT-Services für eine Organisation. Informationen zur Integration von Cloud-Vorgängen mit ITSM-Tools finden Sie im Leitfaden zur Betriebsintegration.

BIS

Siehe IT-Informationsbibliothek.

ITSM

Siehe IT-Servicemanagement.

L

Labelbasierte Zugangskontrolle (LBAC)

Eine Implementierung der Mandatory Access Control (MAC), bei der den Benutzern und den Daten selbst jeweils explizit ein Sicherheitslabelwert zugewiesen wird. Die Schnittmenge zwischen der Benutzersicherheitsbeschriftung und der Datensicherheitsbeschriftung bestimmt, welche Zeilen und Spalten für den Benutzer sichtbar sind.

Landing Zone

Eine landing zone ist eine gut strukturierte AWS Umgebung mit mehreren Konten, die skalierbar und sicher ist. Dies ist ein Ausgangspunkt, von dem aus Ihre Organisationen Workloads und Anwendungen schnell und mit Vertrauen in ihre Sicherheits- und Infrastrukturumgebung starten

und bereitstellen können. Weitere Informationen zu Landing Zones finden Sie unter Einrichtung einer sicheren und skalierbaren AWS -Umgebung mit mehreren Konten..

großes Sprachmodell (LLM)

Ein Deep-Learning-KI-Modell, das anhand einer riesigen Datenmenge vorab trainiert wurde. Ein LLM kann mehrere Aufgaben ausführen, z. B. Fragen beantworten, Dokumente zusammenfassen, Text in andere Sprachen übersetzen und Sätze vervollständigen. Weitere Informationen finden Sie unter Was sind. LLMs

Große Migration

Eine Migration von 300 oder mehr Servern.

SCHWARZ

Siehe Labelbasierte Zugriffskontrolle.

Geringste Berechtigung

Die bewährte Sicherheitsmethode, bei der nur die für die Durchführung einer Aufgabe erforderlichen Mindestberechtigungen erteilt werden. Weitere Informationen finden Sie unter Geringste Berechtigungen anwenden in der IAM-Dokumentation.

Lift and Shift

Siehe 7 Rs.

Little-Endian-System

Ein System, welches das niedrigwertigste Byte zuerst speichert. Siehe auch Endianness.

LLM

Siehe großes Sprachmodell.

Niedrigere Umgebungen

Siehe Umgebung.

M

Machine Learning (ML)

Eine Art künstlicher Intelligenz, die Algorithmen und Techniken zur Mustererkennung und zum Lernen verwendet. ML analysiert aufgezeichnete Daten, wie z. B. Daten aus dem Internet der

Dinge (IoT), und lernt daraus, um ein statistisches Modell auf der Grundlage von Mustern zu erstellen. Weitere Informationen finden Sie unter Machine Learning.

Hauptzweig

Siehe Filiale.

Malware

Software, die entwickelt wurde, um die Computersicherheit oder den Datenschutz zu gefährden. Malware kann Computersysteme stören, vertrauliche Informationen durchsickern lassen oder sich unbefugten Zugriff verschaffen. Beispiele für Malware sind Viren, Würmer, Ransomware, Trojaner, Spyware und Keylogger.

verwaltete Dienste

AWS-Services für die die Infrastrukturebene, das Betriebssystem und die Plattformen AWS betrieben werden, und Sie greifen auf die Endgeräte zu, um Daten zu speichern und abzurufen. Amazon Simple Storage Service (Amazon S3) und Amazon DynamoDB sind Beispiele für Managed Services. Diese werden auch als abstrakte Dienste bezeichnet.

Manufacturing Execution System (MES)

Ein Softwaresystem zur Verfolgung, Überwachung, Dokumentation und Steuerung von Produktionsprozessen, bei denen Rohstoffe in der Fertigung zu fertigen Produkten umgewandelt werden.

MAP

Siehe Migration Acceleration Program.

Mechanismus

Ein vollständiger Prozess, bei dem Sie ein Tool erstellen, die Akzeptanz des Tools vorantreiben und anschließend die Ergebnisse überprüfen, um Anpassungen vorzunehmen. Ein Mechanismus ist ein Zyklus, der sich im Laufe seiner Tätigkeit selbst verstärkt und verbessert. Weitere Informationen finden Sie unter Aufbau von Mechanismen im AWS Well-Architected Framework.

Mitgliedskonto

Alle AWS-Konten außer dem Verwaltungskonto, die Teil einer Organisation in sind. AWS Organizations Ein Konto kann jeweils nur Mitglied einer Organisation sein.

MES

Siehe Manufacturing Execution System.

Message Queuing-Telemetrietransport (MQTT)

Ein leichtes machine-to-machine (M2M) -Kommunikationsprotokoll, das auf dem Publish/Subscribe-Muster für IoT-Geräte mit beschränkten Ressourcen basiert.

Microservice

Ein kleiner, unabhängiger Dienst, der über genau definierte Kanäle kommuniziert APIs und in der Regel kleinen, eigenständigen Teams gehört. Ein Versicherungssystem kann beispielsweise Microservices beinhalten, die Geschäftsfunktionen wie Vertrieb oder Marketing oder Subdomains wie Einkauf, Schadenersatz oder Analytik zugeordnet sind. Zu den Vorteilen von Microservices gehören Agilität, flexible Skalierung, einfache Bereitstellung, wiederverwendbarer Code und Ausfallsicherheit. Weitere Informationen finden Sie unter Integration von Microservices mithilfe serverloser Dienste. AWS

Microservices-Architekturen

Ein Ansatz zur Erstellung einer Anwendung mit unabhängigen Komponenten, die jeden Anwendungsprozess als Microservice ausführen. Diese Microservices kommunizieren mithilfe von Lightweight über eine klar definierte Schnittstelle. APIs Jeder Microservice in dieser Architektur kann aktualisiert, bereitgestellt und skaliert werden, um den Bedarf an bestimmten Funktionen einer Anwendung zu decken. Weitere Informationen finden Sie unter Implementierung von Microservices auf. AWS

Migration Acceleration Program (MAP)

Ein AWS Programm, das Beratung, Unterstützung, Schulungen und Services bietet, um Unternehmen dabei zu unterstützen, eine solide betriebliche Grundlage für die Umstellung auf die Cloud zu schaffen und die anfänglichen Kosten von Migrationen auszugleichen. MAP umfasst eine Migrationsmethode für die methodische Durchführung von Legacy-Migrationen sowie eine Reihe von Tools zur Automatisierung und Beschleunigung gängiger Migrationsszenarien.

Migration in großem Maßstab

Der Prozess, bei dem der Großteil des Anwendungsportfolios in Wellen in die Cloud verlagert wird, wobei in jeder Welle mehr Anwendungen schneller migriert werden. In dieser Phase werden die bewährten Verfahren und Erkenntnisse aus den früheren Phasen zur Implementierung einer Migrationsfabrik von Teams, Tools und Prozessen zur Optimierung der Migration von Workloads durch Automatisierung und agile Bereitstellung verwendet. Dies ist die dritte Phase der AWS - Migrationsstrategie.

Migrationsfabrik

Funktionsübergreifende Teams, die die Migration von Workloads durch automatisierte, agile Ansätze optimieren. Zu den Teams in der Migrationsabteilung gehören in der Regel Betriebsabläufe, Geschäftsanalysten und Eigentümer, Migrationsingenieure, Entwickler und DevOps Experten, die in Sprints arbeiten. Zwischen 20 und 50 Prozent eines Unternehmensanwendungsportfolios bestehen aus sich wiederholenden Mustern, die durch einen Fabrik-Ansatz optimiert werden können. Weitere Informationen finden Sie in Diskussion über Migrationsfabriken und den Leitfaden zur Cloud-Migration-Fabrik in diesem Inhaltssatz.

Migrationsmetadaten

Die Informationen über die Anwendung und den Server, die für den Abschluss der Migration benötigt werden. Für jedes Migrationsmuster ist ein anderer Satz von Migrationsmetadaten erforderlich. Beispiele für Migrationsmetadaten sind das Zielsubnetz, die Sicherheitsgruppe und AWS das Konto.

Migrationsmuster

Eine wiederholbare Migrationsaufgabe, in der die Migrationsstrategie, das Migrationsziel und die verwendete Migrationsanwendung oder der verwendete Migrationsservice detailliert beschrieben werden. Beispiel: Rehost-Migration zu Amazon EC2 mit AWS Application Migration Service.

Migration Portfolio Assessment (MPA)

Ein Online-Tool, das Informationen zur Validierung des Geschäftsszenarios für die Migration auf das bereitstellt. AWS Cloud MPA bietet eine detaillierte Portfoliobewertung (richtige Servergröße, Preisgestaltung, Gesamtbetriebskostenanalyse, Migrationskostenanalyse) sowie Migrationsplanung (Anwendungsdatenanalyse und Datenerfassung, Anwendungsgruppierung, Migrationspriorisierung und Wellenplanung). Das MPA-Tool (Anmeldung erforderlich) steht allen AWS Beratern und APN-Partnerberatern kostenlos zur Verfügung.

Migration Readiness Assessment (MRA)

Der Prozess, bei dem mithilfe des AWS CAF Erkenntnisse über den Cloud-Bereitschaftsstatus eines Unternehmens gewonnen, Stärken und Schwächen identifiziert und ein Aktionsplan zur Schließung festgestellter Lücken erstellt wird. Weitere Informationen finden Sie im Benutzerhandbuch für Migration Readiness. MRA ist die erste Phase der AWS - Migrationsstrategie.

Migrationsstrategie

Der Ansatz, der verwendet wurde, um einen Workload auf den AWS Cloud zu migrieren. Weitere Informationen finden Sie im Eintrag 7 Rs in diesem Glossar und unter Mobilisieren Sie Ihr Unternehmen, um umfangreiche Migrationen zu beschleunigen.

ML

Siehe maschinelles Lernen.

Modernisierung

Umwandlung einer veralteten (veralteten oder monolithischen) Anwendung und ihrer Infrastruktur in ein agiles, elastisches und hochverfügbares System in der Cloud, um Kosten zu senken, die Effizienz zu steigern und Innovationen zu nutzen. Weitere Informationen finden Sie unter Strategie zur Modernisierung von Anwendungen in der AWS Cloud.

Bewertung der Modernisierungsfähigkeit

Eine Bewertung, anhand derer festgestellt werden kann, ob die Anwendungen einer Organisation für die Modernisierung bereit sind, Vorteile, Risiken und Abhängigkeiten identifiziert und ermittelt wird, wie gut die Organisation den zukünftigen Status dieser Anwendungen unterstützen kann. Das Ergebnis der Bewertung ist eine Vorlage der Zielarchitektur, eine Roadmap, in der die Entwicklungsphasen und Meilensteine des Modernisierungsprozesses detailliert beschrieben werden, sowie ein Aktionsplan zur Behebung festgestellter Lücken. Weitere Informationen finden Sie unter Evaluierung der Modernisierungsbereitschaft von Anwendungen in der AWS Cloud.

Monolithische Anwendungen (Monolithen)

Anwendungen, die als ein einziger Service mit eng gekoppelten Prozessen ausgeführt werden. Monolithische Anwendungen haben verschiedene Nachteile. Wenn ein Anwendungs-Feature stark nachgefragt wird, muss die gesamte Architektur skaliert werden. Das Hinzufügen oder Verbessern der Feature einer monolithischen Anwendung wird ebenfalls komplexer, wenn die Codebasis wächst. Um diese Probleme zu beheben, können Sie eine Microservices-Architektur verwenden. Weitere Informationen finden Sie unter Zerlegen von Monolithen in Microservices.

MPA

Siehe Bewertung des Migrationsportfolios.

MQTT

Siehe Message Queuing-Telemetrietransport.

Mehrklassen-Klassifizierung

Ein Prozess, der dabei hilft, Vorhersagen für mehrere Klassen zu generieren (wobei eines von mehr als zwei Ergebnissen vorhergesagt wird). Ein ML-Modell könnte beispielsweise fragen: „Ist dieses Produkt ein Buch, ein Auto oder ein Telefon?“ oder „Welche Kategorie von Produkten ist für diesen Kunden am interessantesten?“

veränderbare Infrastruktur

Ein Modell, das die bestehende Infrastruktur für Produktionsworkloads aktualisiert und modifiziert. Für eine verbesserte Konsistenz, Zuverlässigkeit und Vorhersagbarkeit empfiehlt das AWS Well-Architected Framework die Verwendung einer unveränderlichen Infrastruktur als bewährte Methode.

O

OAC

Siehe Origin Access Control.

EICHE

Siehe Zugriffsidentität von Origin.

COM

Siehe organisatorisches Change-Management.

Offline-Migration

Eine Migrationsmethode, bei der der Quell-Workload während des Migrationsprozesses heruntergefahren wird. Diese Methode ist mit längeren Ausfallzeiten verbunden und wird in der Regel für kleine, unkritische Workloads verwendet.

OI

Siehe Betriebsintegration.

OLA

Siehe Vereinbarung auf operativer Ebene.

Online-Migration

Eine Migrationsmethode, bei der der Quell-Workload auf das Zielsystem kopiert wird, ohne offline genommen zu werden. Anwendungen, die mit dem Workload verbunden sind, können während

der Migration weiterhin funktionieren. Diese Methode beinhaltet keine bis minimale Ausfallzeit und wird in der Regel für kritische Produktionsworkloads verwendet.

OPC-UA

Siehe Open Process Communications — Unified Architecture.

Offene Prozesskommunikation — Einheitliche Architektur (OPC-UA)

Ein machine-to-machine (M2M) -Kommunikationsprotokoll für die industrielle Automatisierung. OPC-UA bietet einen Interoperabilitätsstandard mit Datenverschlüsselungs-, Authentifizierungs- und Autorisierungsschemata.

Vereinbarung auf Betriebsebene (OLA)

Eine Vereinbarung, in der klargestellt wird, welche funktionalen IT-Gruppen sich gegenseitig versprechen zu liefern, um ein Service Level Agreement (SLA) zu unterstützen.

Überprüfung der Betriebsbereitschaft (ORR)

Eine Checkliste mit Fragen und zugehörigen bewährten Methoden, die Ihnen helfen, Vorfälle und mögliche Ausfälle zu verstehen, zu bewerten, zu verhindern oder deren Umfang zu reduzieren. Weitere Informationen finden Sie unter Operational Readiness Reviews (ORR) im AWS Well-Architected Framework.

Betriebstechnologie (OT)

Hardware- und Softwaresysteme, die mit der physischen Umgebung zusammenarbeiten, um industrielle Abläufe, Ausrüstung und Infrastruktur zu steuern. In der Fertigung ist die Integration von OT- und Informationstechnologie (IT) -Systemen ein zentraler Schwerpunkt der Industrie 4.0-Transformationen.

Betriebsintegration (OI)

Der Prozess der Modernisierung von Abläufen in der Cloud, der Bereitschaftsplanung, Automatisierung und Integration umfasst. Weitere Informationen finden Sie im Leitfaden zur Betriebsintegration.

Organisationspfad

Ein Pfad, der von erstellt wird und in AWS CloudTrail dem alle Ereignisse für alle AWS-Konten in einer Organisation protokolliert werden. AWS Organizations Diese Spur wird in jedem AWS-Konto , der Teil der Organisation ist, erstellt und verfolgt die Aktivität in jedem Konto. Weitere Informationen finden Sie in der CloudTrail Dokumentation unter Einen Trail für eine Organisation erstellen.

Organisatorisches Veränderungsmanagement (OCM)

Ein Framework für das Management wichtiger, disruptiver Geschäftstransformationen aus Sicht der Mitarbeiter, der Kultur und der Führung. OCM hilft Organisationen dabei, sich auf neue Systeme und Strategien vorzubereiten und auf diese umzustellen, indem es die Akzeptanz von Veränderungen beschleunigt, Übergangsprobleme angeht und kulturelle und organisatorische Veränderungen vorantreibt. In der AWS Migrationsstrategie wird dieses Framework aufgrund der Geschwindigkeit des Wandels, der bei Projekten zur Cloud-Einführung erforderlich ist, als Mitarbeiterbeschleunigung bezeichnet. Weitere Informationen finden Sie im OCM-Handbuch.

Ursprungszugriffskontrolle (OAC)

In CloudFront, eine erweiterte Option zur Zugriffsbeschränkung, um Ihre Amazon Simple Storage Service (Amazon S3) -Inhalte zu sichern. OAC unterstützt alle S3-Buckets insgesamt AWS-Regionen, serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) sowie dynamische PUT und `DELETE` Anfragen an den S3-Bucket.

Ursprungszugriffsidentität (OAI)

In CloudFront, eine Option zur Zugriffsbeschränkung, um Ihre Amazon S3 S3-Inhalte zu sichern. Wenn Sie OAI verwenden, CloudFront erstellt es einen Principal, mit dem sich Amazon S3 authentifizieren kann. Authentifizierte Principals können nur über eine bestimmte Distribution auf Inhalte in einem S3-Bucket zugreifen. CloudFront Siehe auch OAC, das eine detailliertere und verbesserte Zugriffskontrolle bietet.

ORR

Weitere Informationen finden Sie unter Überprüfung der Betriebsbereitschaft.

NICHT

Siehe Betriebstechnologie.

Ausgehende (egress) VPC

In einer Architektur AWS mit mehreren Konten eine VPC, die Netzwerkverbindungen verarbeitet, die von einer Anwendung aus initiiert werden. Die AWS Security Reference Architecture empfiehlt die Einrichtung Ihres Netzwerkkontos mit eingehendem und ausgehendem Datenverkehr sowie Inspektion, VPCs um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet im weiteren Sinne zu schützen.

P

Berechtigungsgrenze

Eine IAM-Verwaltungsrichtlinie, die den IAM-Prinzipalen zugeordnet ist, um die maximalen Berechtigungen festzulegen, die der Benutzer oder die Rolle haben kann. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM-Entitys in der IAM-Dokumentation.

persönlich identifizierbare Informationen (PII)

Informationen, die, wenn sie direkt betrachtet oder mit anderen verwandten Daten kombiniert werden, verwendet werden können, um vernünftige Rückschlüsse auf die Identität einer Person zu ziehen. Beispiele für personenbezogene Daten sind Namen, Adressen und Kontaktinformationen.

Personenbezogene Daten

Siehe persönlich identifizierbare Informationen.

Playbook

Eine Reihe vordefinierter Schritte, die die mit Migrationen verbundenen Aufgaben erfassen, z. B. die Bereitstellung zentraler Betriebsfunktionen in der Cloud. Ein Playbook kann die Form von Skripten, automatisierten Runbooks oder einer Zusammenfassung der Prozesse oder Schritte annehmen, die für den Betrieb Ihrer modernisierten Umgebung erforderlich sind.

PLC

Siehe programmierbare Logiksteuerung.

PLM

Siehe Produktlebenszyklusmanagement.

policy

Ein Objekt, das Berechtigungen definieren (siehe identitätsbasierte Richtlinie), Zugriffsbedingungen spezifizieren (siehe ressourcenbasierte Richtlinie) oder die maximalen Berechtigungen für alle Konten in einer Organisation definieren kann AWS Organizations (siehe Dienststeuerungsrichtlinie).

Polyglotte Beharrlichkeit

Unabhängige Auswahl der Datenspeichertechnologie eines Microservices auf der Grundlage von Datenzugriffsmustern und anderen Anforderungen. Wenn Ihre Microservices über dieselbe Datenspeichertechnologie verfügen, kann dies zu Implementierungsproblemen oder zu

Leistungseinbußen führen. Microservices lassen sich leichter implementieren und erzielen eine bessere Leistung und Skalierbarkeit, wenn sie den Datenspeicher verwenden, der ihren Anforderungen am besten entspricht.

Portfoliobewertung

Ein Prozess, bei dem das Anwendungsportfolio ermittelt, analysiert und priorisiert wird, um die Migration zu planen. Weitere Informationen finden Sie in Bewerten der Migrationsbereitschaft.

predicate

Eine Abfragebedingung, die `true` oder zurückgibt`false`, was üblicherweise in einer Klausel vorkommt. `WHERE`

Prädikat Pushdown

Eine Technik zur Optimierung von Datenbankabfragen, bei der die Daten in der Abfrage vor der Übertragung gefiltert werden. Dadurch wird die Datenmenge reduziert, die aus der relationalen Datenbank abgerufen und verarbeitet werden muss, und die Abfrageleistung wird verbessert.

Präventive Kontrolle

Eine Sicherheitskontrolle, die verhindern soll, dass ein Ereignis eintritt. Diese Kontrollen stellen eine erste Verteidigungslinie dar, um unbefugten Zugriff oder unerwünschte Änderungen an Ihrem Netzwerk zu verhindern. Weitere Informationen finden Sie unter Präventive Kontrolle in Implementierung von Sicherheitskontrollen in AWS.

Prinzipal

Eine Entität AWS , die Aktionen ausführen und auf Ressourcen zugreifen kann. Diese Entität ist in der Regel ein Root-Benutzer für eine AWS-Konto, eine IAM-Rolle oder einen Benutzer. Weitere Informationen finden Sie unter Prinzipal in Rollenbegriffe und -konzepte in der IAM-Dokumentation.

Datenschutz von Natur aus

Ein systemtechnischer Ansatz, der den Datenschutz während des gesamten Entwicklungsprozesses berücksichtigt.

Privat gehostete Zonen

Ein Container, der Informationen darüber enthält, wie Amazon Route 53 auf DNS-Abfragen für eine Domain und deren Subdomains innerhalb einer oder mehrerer VPCs Domains antworten soll. Weitere Informationen finden Sie unter Arbeiten mit privat gehosteten Zonen in der Route-53-Dokumentation.

proaktive Steuerung

Eine Sicherheitskontrolle, die den Einsatz nicht richtlinienkonformer Ressourcen verhindern soll. Diese Steuerelemente scannen Ressourcen, bevor sie bereitgestellt werden. Wenn die Ressource nicht der Kontrolle entspricht, wird sie nicht bereitgestellt. Weitere Informationen finden Sie im Referenzhandbuch zu Kontrollen in der AWS Control Tower Dokumentation und unter Proaktive Kontrollen unter Implementierung von Sicherheitskontrollen am AWS.

Produktlebenszyklusmanagement (PLM)

Das Management von Daten und Prozessen für ein Produkt während seines gesamten Lebenszyklus, vom Design, der Entwicklung und Markteinführung über Wachstum und Reife bis hin zur Markteinführung und Markteinführung.

Produktionsumgebung

Siehe Umgebung.

Speicherprogrammierbare Steuerung (SPS)

In der Fertigung ein äußerst zuverlässiger, anpassungsfähiger Computer, der Maschinen überwacht und Fertigungsprozesse automatisiert.

schnelle Verkettung

Verwendung der Ausgabe einer LLM-Eingabeaufforderung als Eingabe für die nächste Aufforderung, um bessere Antworten zu generieren. Diese Technik wird verwendet, um eine komplexe Aufgabe in Unteraufgaben zu unterteilen oder um eine vorläufige Antwort iterativ zu verfeinern oder zu erweitern. Sie trägt dazu bei, die Genauigkeit und Relevanz der Antworten eines Modells zu verbessern und ermöglicht detailliertere, personalisierte Ergebnisse.

Pseudonymisierung

Der Prozess, bei dem persönliche Identifikatoren in einem Datensatz durch Platzhalterwerte ersetzt werden. Pseudonymisierung kann zum Schutz der Privatsphäre beitragen. Pseudonymisierte Daten gelten weiterhin als personenbezogene Daten.

publish/subscribe (pub/sub)

Ein Muster, das asynchrone Kommunikation zwischen Microservices ermöglicht, um die Skalierbarkeit und Reaktionsfähigkeit zu verbessern. In einem auf Microservices basierenden MES kann ein Microservice beispielsweise Ereignismeldungen in einem Kanal veröffentlichen, den andere Microservices abonnieren können. Das System kann neue Microservices hinzufügen, ohne den Veröffentlichungsservice zu ändern.

Q

Abfrageplan

Eine Reihe von Schritten, wie Anweisungen, die für den Zugriff auf die Daten in einem relationalen SQL-Datenbanksystem verwendet werden.

Abfrageplanregression

Wenn ein Datenbankserviceoptimierer einen weniger optimalen Plan wählt als vor einer bestimmten Änderung der Datenbankumgebung. Dies kann durch Änderungen an Statistiken, Beschränkungen, Umgebungseinstellungen, Abfrageparameter-Bindungen und Aktualisierungen der Datenbank-Engine verursacht werden.

R

RACI-Matrix

Siehe verantwortlich, rechenschaftspflichtig, konsultiert, informiert (RACI).

RAG

Siehe Erweiterte Generierung beim Abrufen.

Ransomware

Eine bösartige Software, die entwickelt wurde, um den Zugriff auf ein Computersystem oder Daten zu blockieren, bis eine Zahlung erfolgt ist.

RASCI-Matrix

Siehe verantwortlich, rechenschaftspflichtig, konsultiert, informiert (RACI).

RCAC

Siehe Zugriffskontrolle für Zeilen und Spalten.

Read Replica

Eine Kopie einer Datenbank, die nur für Lesezwecke verwendet wird. Sie können Abfragen an das Lesereplikat weiterleiten, um die Belastung auf Ihrer Primärdatenbank zu reduzieren.

neu strukturieren

Siehe 7 Rs.

Recovery Point Objective (RPO)

Die maximal zulässige Zeitspanne seit dem letzten Datenwiederherstellungspunkt. Damit wird festgelegt, was als akzeptabler Datenverlust zwischen dem letzten Wiederherstellungspunkt und der Serviceunterbrechung gilt.

Wiederherstellungszeitziel (RTO)

Die maximal zulässige Verzögerung zwischen der Betriebsunterbrechung und der Wiederherstellung des Dienstes.

Refaktorierung

Siehe 7 Rs.

Region

Eine Sammlung von AWS Ressourcen in einem geografischen Gebiet. Jeder AWS-Region ist isoliert und unabhängig von den anderen, um Fehlertoleranz, Stabilität und Belastbarkeit zu gewährleisten. Weitere Informationen finden Sie unter Geben Sie an, was AWS-Regionen Ihr Konto verwenden kann.

Regression

Eine ML-Technik, die einen numerischen Wert vorhersagt. Zum Beispiel, um das Problem „Zu welchem Preis wird dieses Haus verkauft werden?“ zu lösen Ein ML-Modell könnte ein lineares Regressionsmodell verwenden, um den Verkaufspreis eines Hauses auf der Grundlage bekannter Fakten über das Haus (z. B. die Quadratmeterzahl) vorherzusagen.

rehosten

Siehe 7 Rs.

Veröffentlichung

In einem Bereitstellungsprozess der Akt der Förderung von Änderungen an einer Produktionsumgebung.

umziehen

Siehe 7 Rs.

neue Plattform

Siehe 7 Rs.

Rückkauf

Siehe 7 Rs.

Ausfallsicherheit

Die Fähigkeit einer Anwendung, Störungen zu widerstehen oder sich von ihnen zu erholen. Hochverfügbarkeit und Notfallwiederherstellung sind häufig Überlegungen bei der Planung der Ausfallsicherheit in der. AWS Cloud Weitere Informationen finden Sie unter AWS Cloud Resilienz.

Ressourcenbasierte Richtlinie

Eine mit einer Ressource verknüpfte Richtlinie, z. B. ein Amazon-S3-Bucket, ein Endpunkt oder ein Verschlüsselungsschlüssel. Diese Art von Richtlinie legt fest, welchen Prinzipalen der Zugriff gewährt wird, welche Aktionen unterstützt werden und welche anderen Bedingungen erfüllt sein müssen.

RACI-Matrix (verantwortlich, rechenschaftspflichtig, konsultiert, informiert)

Eine Matrix, die die Rollen und Verantwortlichkeiten aller an Migrationsaktivitäten und Cloud-Operationen beteiligten Parteien definiert. Der Matrixname leitet sich von den in der Matrix definierten Zuständigkeitstypen ab: verantwortlich (R), rechenschaftspflichtig (A), konsultiert (C) und informiert (I). Der Unterstützungstyp (S) ist optional. Wenn Sie Unterstützung einbeziehen, wird die Matrix als RASCI-Matrix bezeichnet, und wenn Sie sie ausschließen, wird sie als RACI-Matrix bezeichnet.

Reaktive Kontrolle

Eine Sicherheitskontrolle, die darauf ausgelegt ist, die Behebung unerwünschter Ereignisse oder Abweichungen von Ihren Sicherheitsstandards voranzutreiben. Weitere Informationen finden Sie unter Reaktive Kontrolle in Implementieren von Sicherheitskontrollen in AWS.

Beibehaltung

Siehe 7 Rs.

zurückziehen

Siehe 7 Rs.

Retrieval Augmented Generation (RAG)

Eine generative KI-Technologie, bei der ein LLM auf eine maßgebliche Datenquelle verweist, die sich außerhalb seiner Trainingsdatenquellen befindet, bevor eine Antwort generiert wird. Ein RAG-Modell könnte beispielsweise eine semantische Suche in der Wissensdatenbank oder in

benutzerdefinierten Daten einer Organisation durchführen. Weitere Informationen finden Sie unter Was ist RAG.

Drehung

Der Vorgang, bei dem ein Geheimnis regelmäßig aktualisiert wird, um es einem Angreifer zu erschweren, auf die Anmeldeinformationen zuzugreifen.

Zugriffskontrolle für Zeilen und Spalten (RCAC)

Die Verwendung einfacher, flexibler SQL-Ausdrücke mit definierten Zugriffsregeln. RCAC besteht aus Zeilenberechtigungen und Spaltenmasken.

RPO

Siehe Recovery Point Objective.

RTO

Siehe Ziel für die Erholungszeit.

Runbook

Eine Reihe manueller oder automatisierter Verfahren, die zur Ausführung einer bestimmten Aufgabe erforderlich sind. Diese sind in der Regel darauf ausgelegt, sich wiederholende Operationen oder Verfahren mit hohen Fehlerquoten zu rationalisieren.

S

SAML 2.0

Ein offener Standard, den viele Identitätsanbieter (IdPs) verwenden. Diese Funktion ermöglicht föderiertes Single Sign-On (SSO), sodass sich Benutzer bei den API-Vorgängen anmelden AWS-Managementkonsole oder die AWS API-Operationen aufrufen können, ohne dass Sie einen Benutzer in IAM für alle in Ihrer Organisation erstellen müssen. Weitere Informationen zum SAML-2.0.-basierten Verbund finden Sie unter Über den SAML-2.0-basierten Verbund in der IAM-Dokumentation.

SCADA

Siehe Aufsichtskontrolle und Datenerfassung.

SCP

Siehe Richtlinie zur Dienstkontrolle.

Secret

Interne AWS Secrets Manager, vertrauliche oder eingeschränkte Informationen, wie z. B. ein Passwort oder Benutzeranmeldeinformationen, die Sie in verschlüsselter Form speichern. Es besteht aus dem geheimen Wert und seinen Metadaten. Der geheime Wert kann binär, eine einzelne Zeichenfolge oder mehrere Zeichenketten sein. Weitere Informationen finden Sie unter Was ist in einem Secrets Manager Manager-Geheimnis? in der Secrets Manager Manager-Dokumentation.

Sicherheit durch Design

Ein systemtechnischer Ansatz, der die Sicherheit während des gesamten Entwicklungsprozesses berücksichtigt.

Sicherheitskontrolle

Ein technischer oder administrativer Integritätsschutz, der die Fähigkeit eines Bedrohungsakteurs, eine Schwachstelle auszunutzen, verhindert, erkennt oder einschränkt. Es gibt vier Haupttypen von Sicherheitskontrollen: präventiv, detektiv, reaktionsschnell und proaktiv.

Härtung der Sicherheit

Der Prozess, bei dem die Angriffsfläche reduziert wird, um sie widerstandsfähiger gegen Angriffe zu machen. Dies kann Aktionen wie das Entfernen von Ressourcen, die nicht mehr benötigt werden, die Implementierung der bewährten Sicherheitsmethode der Gewährung geringster Berechtigungen oder die Deaktivierung unnötiger Feature in Konfigurationsdateien umfassen.

System zur Verwaltung von Sicherheitsinformationen und Ereignissen (security information and event management – SIEM)

Tools und Services, die Systeme für das Sicherheitsinformationsmanagement (SIM) und das Management von Sicherheitsereignissen (SEM) kombinieren. Ein SIEM-System sammelt, überwacht und analysiert Daten von Servern, Netzwerken, Geräten und anderen Quellen, um Bedrohungen und Sicherheitsverletzungen zu erkennen und Warnmeldungen zu generieren.

Automatisierung von Sicherheitsreaktionen

Eine vordefinierte und programmierte Aktion, die darauf ausgelegt ist, automatisch auf ein Sicherheitsereignis zu reagieren oder es zu beheben. Diese Automatisierungen dienen als detektive oder reaktionsschnelle Sicherheitskontrollen, die Sie bei der Implementierung bewährter AWS Sicherheitsmethoden unterstützen. Beispiele für automatisierte Antwortaktionen sind das Ändern einer VPC-Sicherheitsgruppe, das Patchen einer Amazon EC2 EC2-Instance oder das Rotieren von Anmeldeinformationen.

Serverseitige Verschlüsselung

Verschlüsselung von Daten am Zielort durch denjenigen AWS-Service , der sie empfängt.

Service-Kontrollrichtlinie (SCP)

Eine Richtlinie, die eine zentrale Steuerung der Berechtigungen für alle Konten in einer Organisation in ermöglicht AWS Organizations. SCPs Definieren Sie Leitplanken oder legen Sie Grenzwerte für Aktionen fest, die ein Administrator an Benutzer oder Rollen delegieren kann. Sie können sie SCPs als Zulassungs- oder Ablehnungslisten verwenden, um festzulegen, welche Dienste oder Aktionen zulässig oder verboten sind. Weitere Informationen finden Sie in der AWS Organizations Dokumentation unter Richtlinien zur Dienststeuerung.

Service-Endpunkt

Die URL des Einstiegspunkts für einen AWS-Service. Sie können den Endpunkt verwenden, um programmgesteuert eine Verbindung zum Zielservice herzustellen. Weitere Informationen finden Sie unter AWS-Service -Endpunkte in der Allgemeine AWS-Referenz.

Service Level Agreement (SLA)

Eine Vereinbarung, in der klargestellt wird, was ein IT-Team seinen Kunden zu bieten verspricht, z. B. in Bezug auf Verfügbarkeit und Leistung der Services.

Service-Level-Indikator (SLI)

Eine Messung eines Leistungsaspekts eines Dienstes, z. B. seiner Fehlerrate, Verfügbarkeit oder Durchsatz.

Service-Level-Ziel (SLO)

Eine Zielkennzahl, die den Zustand eines Dienstes darstellt, gemessen anhand eines Service-Level-Indikators.

Modell der geteilten Verantwortung

Ein Modell, das die Verantwortung beschreibt, mit der Sie gemeinsam AWS für Cloud-Sicherheit und Compliance verantwortlich sind. AWS ist für die Sicherheit der Cloud verantwortlich, während Sie für die Sicherheit in der Cloud verantwortlich sind. Weitere Informationen finden Sie unter Modell der geteilten Verantwortung.

SIEM

Siehe Sicherheitsinformations- und Event-Management-System.

Single Point of Failure (SPOF)

Ein Fehler in einer einzelnen, kritischen Komponente einer Anwendung, der das System stören kann.

SLA

Siehe Service Level Agreement.

SLI

Siehe Service-Level-Indikator.

ALSO

Siehe Service-Level-Ziel.

split-and-seed Modell

Ein Muster für die Skalierung und Beschleunigung von Modernisierungsprojekten. Sobald neue Features und Produktversionen definiert werden, teilt sich das Kernteam auf, um neue Produktteams zu bilden. Dies trägt zur Skalierung der Fähigkeiten und Services Ihrer Organisation bei, verbessert die Produktivität der Entwickler und unterstützt schnelle Innovationen. Weitere Informationen finden Sie unter Schrittweiser Ansatz zur Modernisierung von Anwendungen in der AWS Cloud.

SPOTTEN

Siehe Single Point of Failure.

Sternschema

Eine Datenbank-Organisationsstruktur, die eine große Faktentabelle zum Speichern von Transaktions- oder Messdaten und eine oder mehrere kleinere dimensionale Tabellen zum Speichern von Datenattributen verwendet. Diese Struktur ist für die Verwendung in einem Data Warehouse oder für Business Intelligence-Zwecke konzipiert.

Strangler-Fig-Muster

Ein Ansatz zur Modernisierung monolithischer Systeme, bei dem die Systemfunktionen schrittweise umgeschrieben und ersetzt werden, bis das Legacy-System außer Betrieb genommen werden kann. Dieses Muster verwendet die Analogie einer Feigenrebe, die zu einem etablierten Baum heranwächst und schließlich ihren Wirt überwindet und ersetzt. Das Muster wurde eingeführt von Martin Fowler als Möglichkeit, Risiken beim Umschreiben monolithischer Systeme zu managen. Ein Beispiel für die Anwendung dieses Musters finden Sie

unter Schrittweises Modernisieren älterer Microsoft ASP.NET (ASMX)-Webservices mithilfe von Containern und Amazon API Gateway.

Subnetz

Ein Bereich von IP-Adressen in Ihrer VPC. Ein Subnetz muss sich in einer einzigen Availability Zone befinden.

Aufsichtskontrolle und Datenerfassung (SCADA)

In der Fertigung ein System, das Hardware und Software zur Überwachung von Sachanlagen und Produktionsabläufen verwendet.

Symmetrische Verschlüsselung

Ein Verschlüsselungsalgorithmus, der denselben Schlüssel zum Verschlüsseln und Entschlüsseln der Daten verwendet.

synthetisches Testen

Testen eines Systems auf eine Weise, die Benutzerinteraktionen simuliert, um potenzielle Probleme zu erkennen oder die Leistung zu überwachen. Sie können Amazon CloudWatch Synthetics verwenden, um diese Tests zu erstellen.

Systemaufforderung

Eine Technik, mit der einem LLM Kontext, Anweisungen oder Richtlinien zur Verfügung gestellt werden, um sein Verhalten zu steuern. Systemaufforderungen helfen dabei, den Kontext festzulegen und Regeln für Interaktionen mit Benutzern festzulegen.

T

tags

Schlüssel-Wert-Paare, die als Metadaten für die Organisation Ihrer Ressourcen dienen. AWS Mit Tags können Sie Ressourcen verwalten, identifizieren, organisieren, suchen und filtern. Weitere Informationen finden Sie unter Markieren Ihrer AWS -Ressourcen.

Zielvariable

Der Wert, den Sie in überwachtem ML vorhersagen möchten. Dies wird auch als Ergebnisvariable bezeichnet. In einer Fertigungsumgebung könnte die Zielvariable beispielsweise ein Produktfehler sein.

Aufgabenliste

Ein Tool, das verwendet wird, um den Fortschritt anhand eines Runbooks zu verfolgen. Eine Aufgabenliste enthält eine Übersicht über das Runbook und eine Liste mit allgemeinen Aufgaben, die erledigt werden müssen. Für jede allgemeine Aufgabe werden der geschätzte Zeitaufwand, der Eigentümer und der Fortschritt angegeben.

Testumgebungen

Siehe Umgebung.

Training

Daten für Ihr ML-Modell bereitstellen, aus denen es lernen kann. Die Trainingsdaten müssen die richtige Antwort enthalten. Der Lernalgorithmus findet Muster in den Trainingsdaten, die die Attribute der Input-Daten dem Ziel (die Antwort, die Sie voraussagen möchten) zuordnen. Es gibt ein ML-Modell aus, das diese Muster erfasst. Sie können dann das ML-Modell verwenden, um Voraussagen für neue Daten zu erhalten, bei denen Sie das Ziel nicht kennen.

Transit-Gateway

Ein Netzwerk-Transit-Hub, über den Sie Ihre Netzwerke VPCs und Ihre lokalen Netzwerke miteinander verbinden können. Weitere Informationen finden Sie in der Dokumentation unter Was ist ein Transit-Gateway. AWS Transit Gateway

Stammbasierter Workflow

Ein Ansatz, bei dem Entwickler Feature lokal in einem Feature-Zweig erstellen und testen und diese Änderungen dann im Hauptzweig zusammenführen. Der Hauptzweig wird dann sequentiell für die Entwicklungs-, Vorproduktions- und Produktionsumgebungen erstellt.

Vertrauenswürdiger Zugriff

Gewährung von Berechtigungen für einen Dienst, den Sie angeben, um Aufgaben in Ihrer Organisation AWS Organizations und in deren Konten in Ihrem Namen auszuführen. Der vertrauenswürdige Service erstellt in jedem Konto eine mit dem Service verknüpfte Rolle, wenn diese Rolle benötigt wird, um Verwaltungsaufgaben für Sie auszuführen. Weitere Informationen finden Sie in der AWS Organizations Dokumentation unter Verwendung AWS Organizations mit anderen AWS Diensten.

Optimieren

Aspekte Ihres Trainingsprozesses ändern, um die Genauigkeit des ML-Modells zu verbessern. Sie können das ML-Modell z. B. trainieren, indem Sie einen Beschriftungssatz generieren,

Beschriftungen hinzufügen und diese Schritte dann mehrmals unter verschiedenen Einstellungen wiederholen, um das Modell zu optimieren.

Zwei-Pizzen-Team

Ein kleines DevOps Team, das Sie mit zwei Pizzen ernähren können. Eine Teamgröße von zwei Pizzen gewährleistet die bestmögliche Gelegenheit zur Zusammenarbeit bei der Softwareentwicklung.

U

Unsicherheit

Ein Konzept, das sich auf ungenaue, unvollständige oder unbekannte Informationen bezieht, die die Zuverlässigkeit von prädiktiven ML-Modellen untergraben können. Es gibt zwei Arten von Unsicherheit: Epistemische Unsicherheit wird durch begrenzte, unvollständige Daten verursacht, wohingegen aleatorische Unsicherheit durch Rauschen und Randomisierung verursacht wird, die in den Daten liegt. Weitere Informationen finden Sie im Leitfaden Quantifizieren der Unsicherheit in Deep-Learning-Systemen.

undifferenzierte Aufgaben

Diese Arbeit wird auch als Schwerstarbeit bezeichnet. Dabei handelt es sich um Arbeiten, die zwar für die Erstellung und den Betrieb einer Anwendung erforderlich sind, aber dem Endbenutzer keinen direkten Mehrwert bieten oder keinen Wettbewerbsvorteil bieten. Beispiele für undifferenzierte Aufgaben sind Beschaffung, Wartung und Kapazitätsplanung.

höhere Umgebungen

Siehe Umgebung.

V

Vacuuming

Ein Vorgang zur Datenbankwartung, bei dem die Datenbank nach inkrementellen Aktualisierungen bereinigt wird, um Speicherplatz zurückzugewinnen und die Leistung zu verbessern.

Versionskontrolle

Prozesse und Tools zur Nachverfolgung von Änderungen, z. B. Änderungen am Quellcode in einem Repository.

VPC-Peering

Eine Verbindung zwischen zwei VPCs , die es Ihnen ermöglicht, den Verkehr mithilfe privater IP-Adressen weiterzuleiten. Weitere Informationen finden Sie unter Was ist VPC-Peering? in der Amazon-VPC-Dokumentation.

Schwachstelle

Ein Software- oder Hardwarefehler, der die Sicherheit des Systems beeinträchtigt.

W

Warmer Cache

Ein Puffer-Cache, der aktuelle, relevante Daten enthält, auf die häufig zugegriffen wird. Die Datenbank-Instance kann aus dem Puffer-Cache lesen, was schneller ist als das Lesen aus dem Hauptspeicher oder von der Festplatte.

warme Daten

Daten, auf die selten zugegriffen wird. Bei der Abfrage dieser Art von Daten sind mäßig langsame Abfragen in der Regel akzeptabel.

Fensterfunktion

Eine SQL-Funktion, die eine Berechnung für eine Gruppe von Zeilen durchführt, die sich in irgendeiner Weise auf den aktuellen Datensatz beziehen. Fensterfunktionen sind nützlich für die Verarbeitung von Aufgaben wie die Berechnung eines gleitenden Durchschnitts oder für den Zugriff auf den Wert von Zeilen auf der Grundlage der relativen Position der aktuellen Zeile.

Workload

Ein Workload ist eine Sammlung von Ressourcen und Code, die einen Unternehmenswert bietet, wie z. B. eine kundenorientierte Anwendung oder ein Backend-Prozess.

Workstream

Funktionsgruppen in einem Migrationsprojekt, die für eine bestimmte Reihe von Aufgaben verantwortlich sind. Jeder Workstream ist unabhängig, unterstützt aber die anderen Workstreams

im Projekt. Der Portfolio-Workstream ist beispielsweise für die Priorisierung von Anwendungen, die Wellenplanung und die Erfassung von Migrationsmetadaten verantwortlich. Der Portfolio-Workstream liefert diese Komponenten an den Migrations-Workstream, der dann die Server und Anwendungen migriert.

WURM

Sehen Sie einmal schreiben, viele lesen.

WQF

Siehe AWS Workload-Qualifizierungsrahmen.

einmal schreiben, viele lesen (WORM)

Ein Speichermodell, das Daten ein einziges Mal schreibt und verhindert, dass die Daten gelöscht oder geändert werden. Autorisierte Benutzer können die Daten so oft wie nötig lesen, aber sie können sie nicht ändern. Diese Datenspeicherinfrastruktur gilt als unveränderlich.

Z

Zero-Day-Exploit

Ein Angriff, in der Regel Malware, der eine Zero-Day-Sicherheitslücke ausnutzt.

Zero-Day-Sicherheitslücke

Ein unfehlbarer Fehler oder eine Sicherheitslücke in einem Produktionssystem. Bedrohungsakteure können diese Art von Sicherheitslücke nutzen, um das System anzugreifen. Entwickler werden aufgrund des Angriffs häufig auf die Sicherheitsanfälligkeit aufmerksam.

Eingabeaufforderung ohne Zwischenfälle

Bereitstellung von Anweisungen für die Ausführung einer Aufgabe an einen LLM, jedoch ohne Beispiele (Schnappschüsse), die ihm als Orientierungshilfe dienen könnten. Der LLM muss sein vortrainiertes Wissen einsetzen, um die Aufgabe zu bewältigen. Die Effektivität von Zero-Shot Prompting hängt von der Komplexität der Aufgabe und der Qualität der Aufforderung ab. Siehe auch Few-Shot-Prompting.

Zombie-Anwendung

Eine Anwendung, deren durchschnittliche CPU- und Arbeitsspeichernutzung unter 5 Prozent liegt. In einem Migrationsprojekt ist es üblich, diese Anwendungen außer Betrieb zu nehmen.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.