AWS re:Invent

DECEMBER 2 – 6, 2024 | LAS VEGAS, NV

STG370

Simplify data management with Amazon S3 Tables

David Lee

Principal Product Manager,
Amazon S3
AWS

Prathiban Mohanasundaram

Senior Software Development Manager,
Amazon S3
AWS

Agenda

01 Introduction to Parquet/Iceberg

02 Introducing Amazon S3 Tables

03 Data management with S3 Tables

04 S3 Tables demo

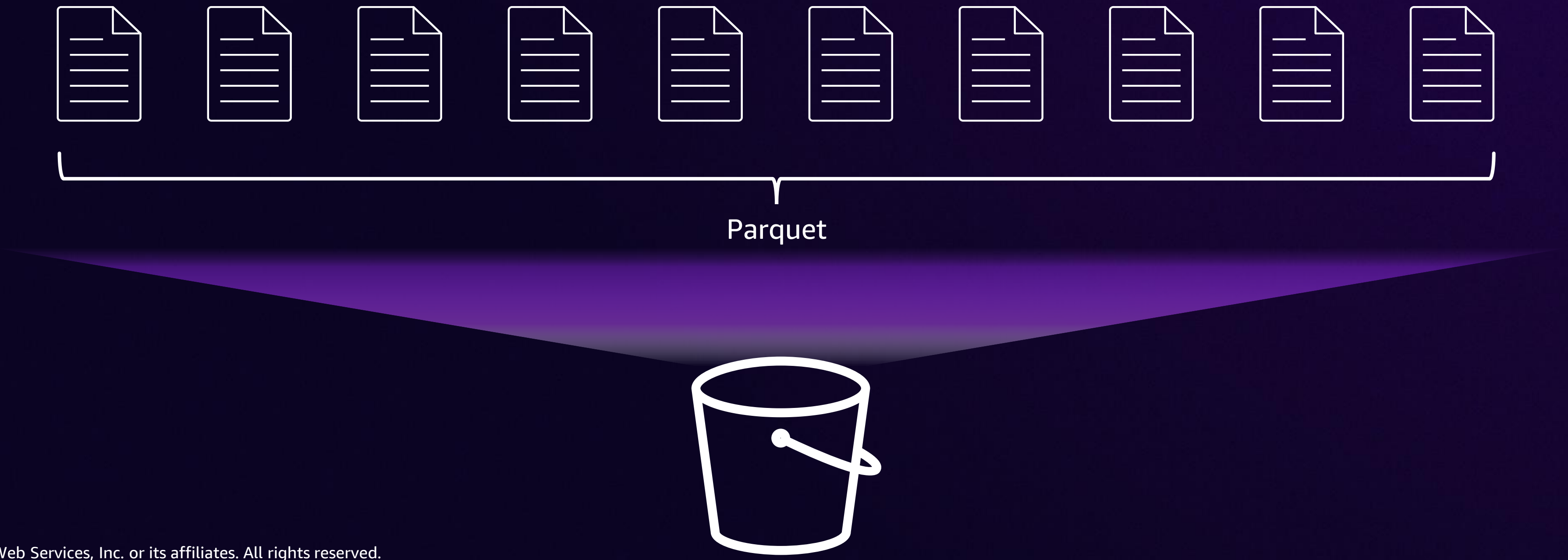
05 Q&A

Today, Amazon S3 is also a *tabular* data store

10+ exabytes of Parquet data stored

Servicing 15 million requests per second

Transmitting hundreds of petabytes every day

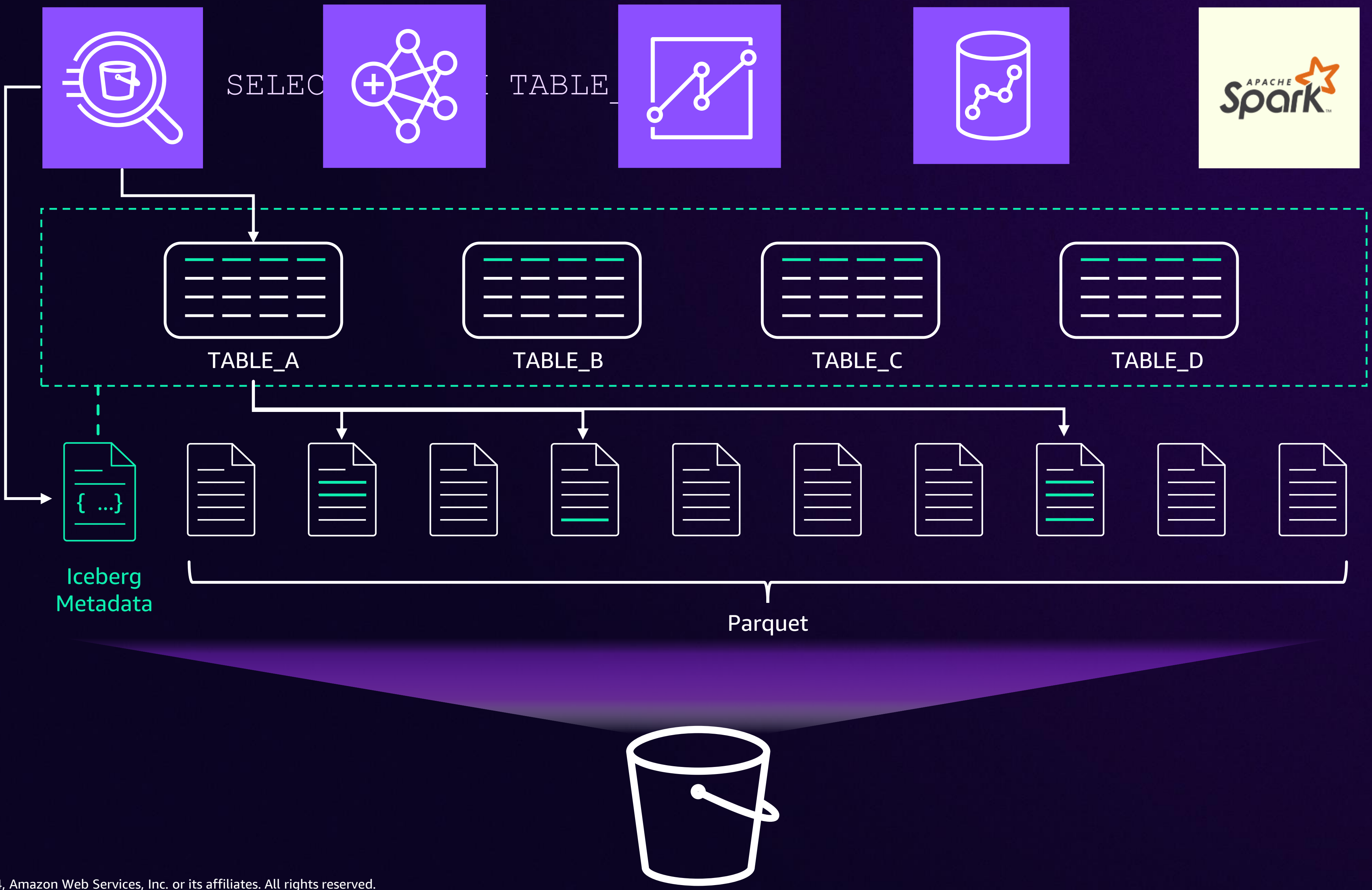


What is Apache Parquet?

Open source

Columnar data file format

Optimized for performance


SELEC
TABLE_
TABLE_A
TABLE_B
TABLE_C
TABLE_D
{ ...}
Iceberg Metadata
Parquet
APACHE Spark

Customer problems to solve

Growing scale requires more and more performance

Enforcing table-level security and integrity is complex

Optimizing storage cost drives unexpected operational burden

GA

Dec. 3, 2024

S3 Tables

Fully Managed Apache Iceberg Tables in S3

Improved query performance based on storage tuning and optimized data layout

Simplified table security and integrity controls

Automated storage cost optimization based on snapshot management and garbage collection

Enhanced scalability and performance

10x

Transactions per second (TPS)

3x

Improvement to query performance

Enhanced scalability and performance

10x

Transactions per second (TPS)

Optimized key naming and layout

Amazon S3 tuned specifically for Iceberg workloads

Enables a higher starting point for S3 TPS scaling

Learn more about Iceberg TPS scaling with Amazon S3 here:

Enhanced scalability and performance

Enhanced scalability and performance

Automatic compaction of underlying Parquet files

Fewer requests to S3

Better throughput

Better tail latencies

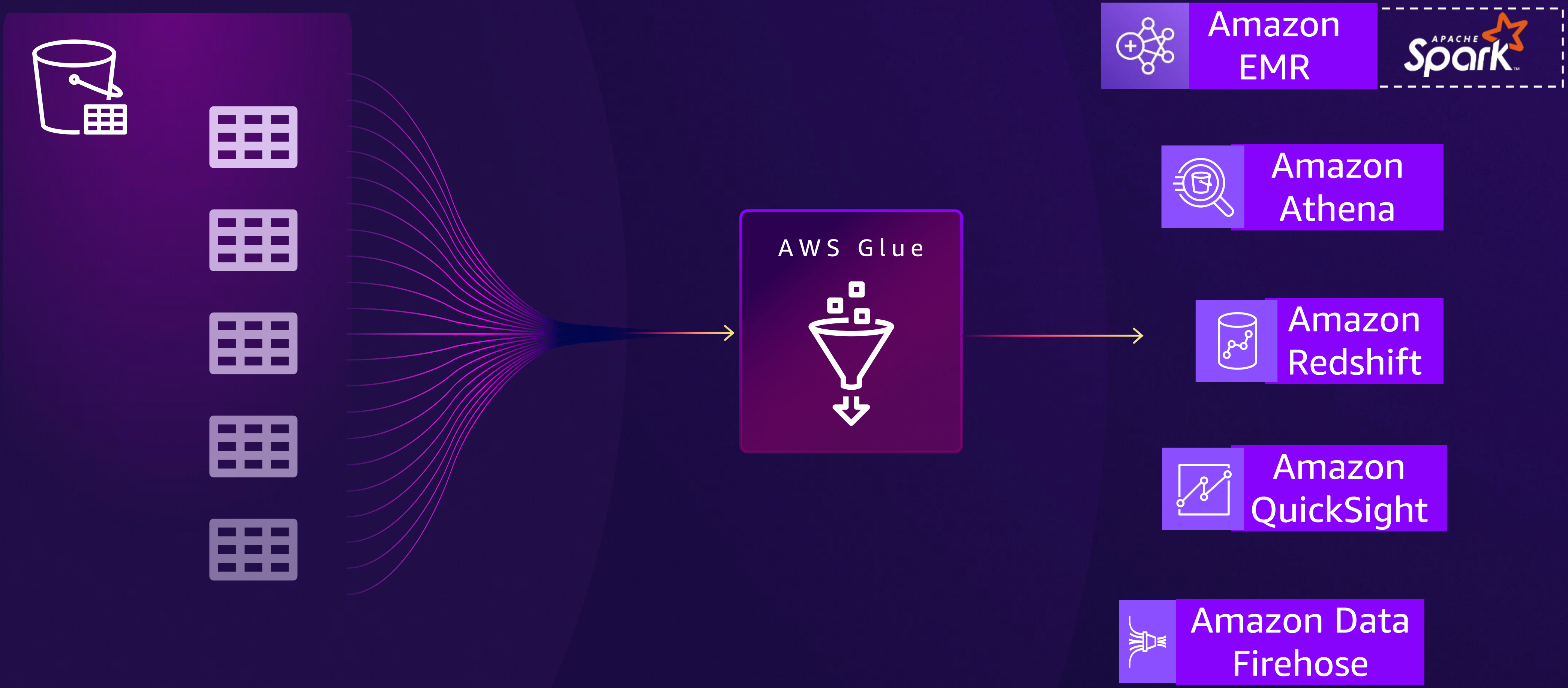
Enhanced scalability and performance

3x

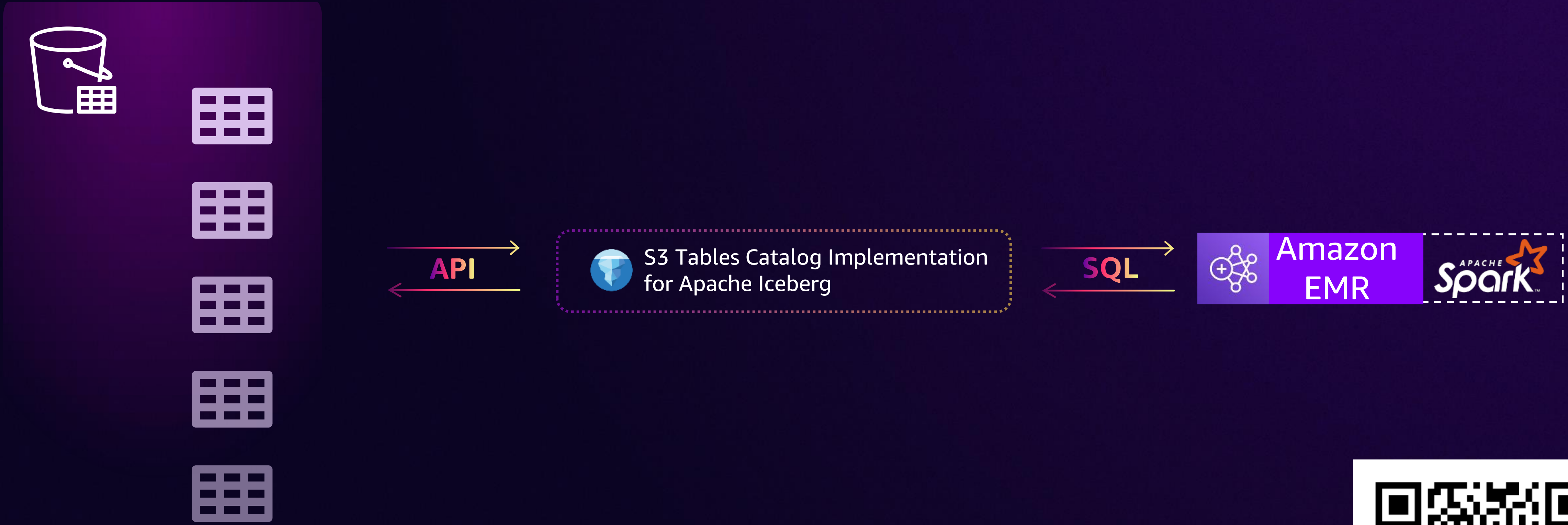
Improvement to query performance

Seamless integration

IN PREVIEW TODAY



Seamless Integration



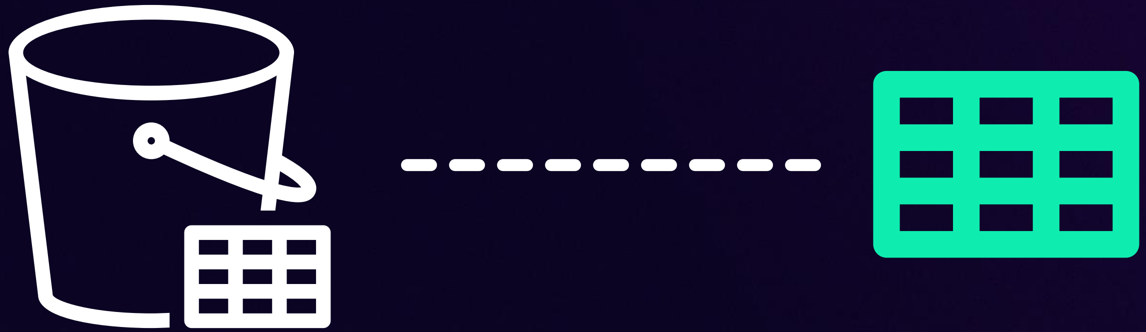
API

S3 Tables Catalog Implementation for Apache Iceberg

SQL

Amazon EMR

Simplified security



The table is a first class AWS resource!

- Has an ARN
- Can take an Amazon S3 resource policy
- Has a dedicated endpoint:

  `s3tables.region.amazonaws.com`

Fully managed

STORAGE COST OPTIMIZATION

Nightly maintenance runs

- Snapshot expiration
- Garbage collection

Fully managed – 100% policy-driven table maintenance

STORAGE COST OPTIMIZATION

Table management policy configuration examples

Garbage collection

```
aws s3tables put-table-bucket-maintenance-configuration \
--table-bucket-arn "arn:aws:s3tables:us-east-2:4236238:bucket/customer-sales-prod \
--type icebergUnreferencedFileRemoval \
--value '{"status":"enabled",\
"settings":{"icebergUnreferencedFileRemoval":{"unreferencedDays":1, "nonCurrentDays":1}}}'
```

Fully managed – 100% policy-driven table maintenance

STORAGE COST OPTIMIZATION

Table management policy configuration examples

Snapshot management

```
aws s3tables put-table-maintenance-configuration \
--table-bucket-arn "arn:aws:s3tables:us-east-2:423623854866:bucket/customer-sales-prod" \
--namespace customer-retail-sales \
--name customer-media-sales-table \
--type icebergSnapshotManagement \
--value '{"status":"enabled","settings": \
{"icebergSnapshotManagement":{"minSnapshotsToKeep":1, "maxSnapshotAgeHours":1}}}'
```

Fully managed – 100% policy-driven table maintenance

STORAGE COST OPTIMIZATION

Table management policy configuration examples

Compaction management

```
aws s3tables put-table-maintenance-configuration \
--table-bucket-arn "arn:aws:s3tables:us-east-2:423623854866:bucket/customer-sales-prod" \
--namespace customer-retail-sales \
--name customer-media-sales-table \
--type icebergCompaction \
--value '{"status":"enabled","settings":{"icebergCompaction":{"targetFileSizeMB":128}}}'
```

S3 Table bucket APIs

NEW APIS

Table operations

```
S3tables:ListTable
S3tables:CreateTable
S3tables:GetTableMetadataLocation
S3tables:UpdateTableMetadataLocation
S3tables:DeleteTable
```

Table management

```
S3tables:PutTablePolicy
S3tables:PutTableBucketPolicy
S3tables:PutTableMaintenanceConfig
S3tables:PutTableBucketMaintenanceConfig
```

Simplify data management with Amazon S3 Tables

Improved performance

Simplified security

ICEBERG

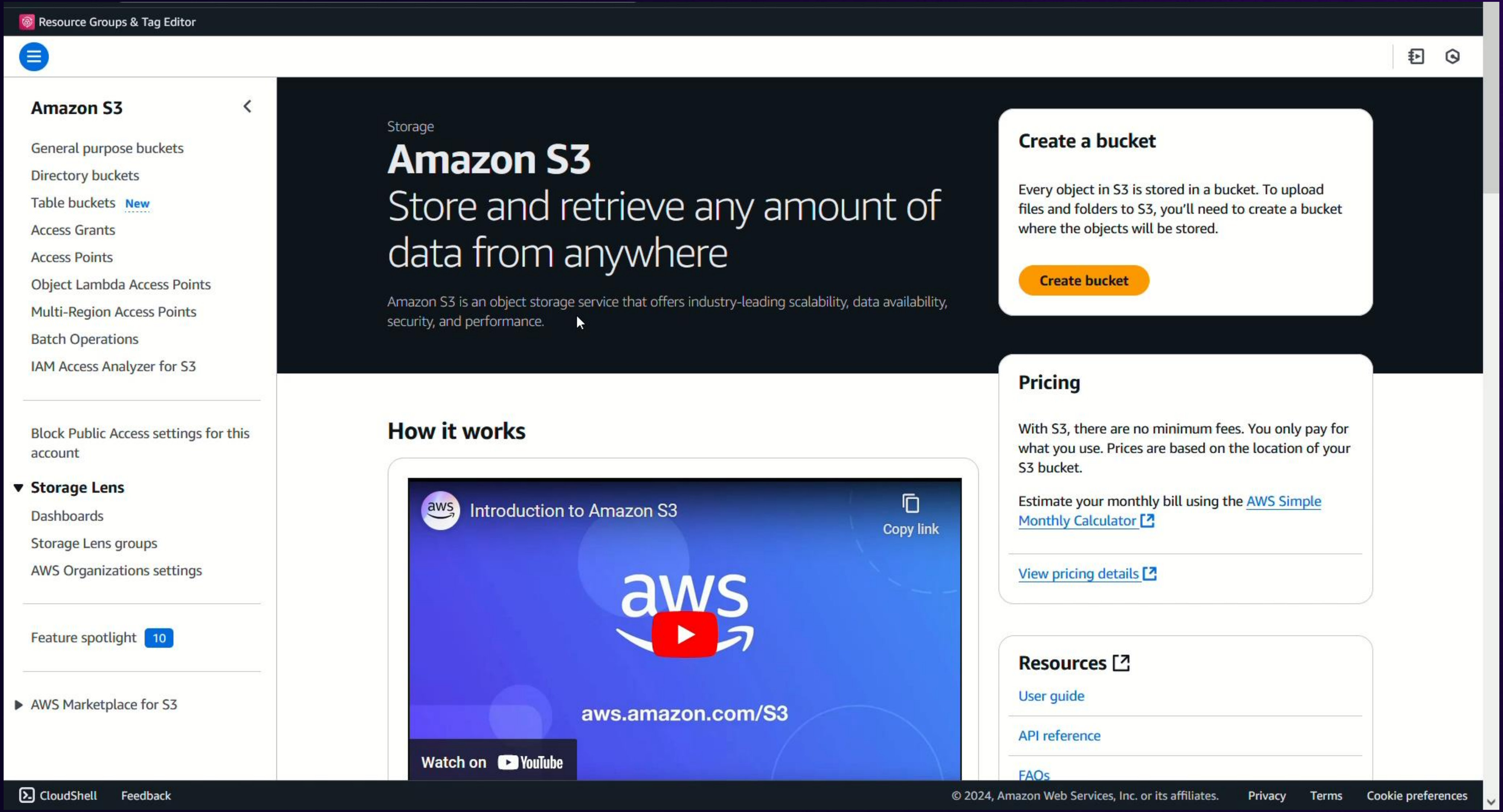
Spark

Seamless integration

Fully managed

S3 Tables demo

Create a table bucket



Resource Groups & Tag Editor

Amazon S3

- General purpose buckets
- Directory buckets
- Table buckets New
- Access Grants
- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- IAM Access Analyzer for S3

Block Public Access settings for this account

▼ Storage Lens

- Dashboards
- Storage Lens groups
- AWS Organizations settings

Feature spotlight 10

▶ AWS Marketplace for S3

Storage

Amazon S3

Store and retrieve any amount of data from anywhere

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

Create bucket

How it works

Pricing

With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket.

Estimate your monthly bill using the AWS Simple Monthly Calculator

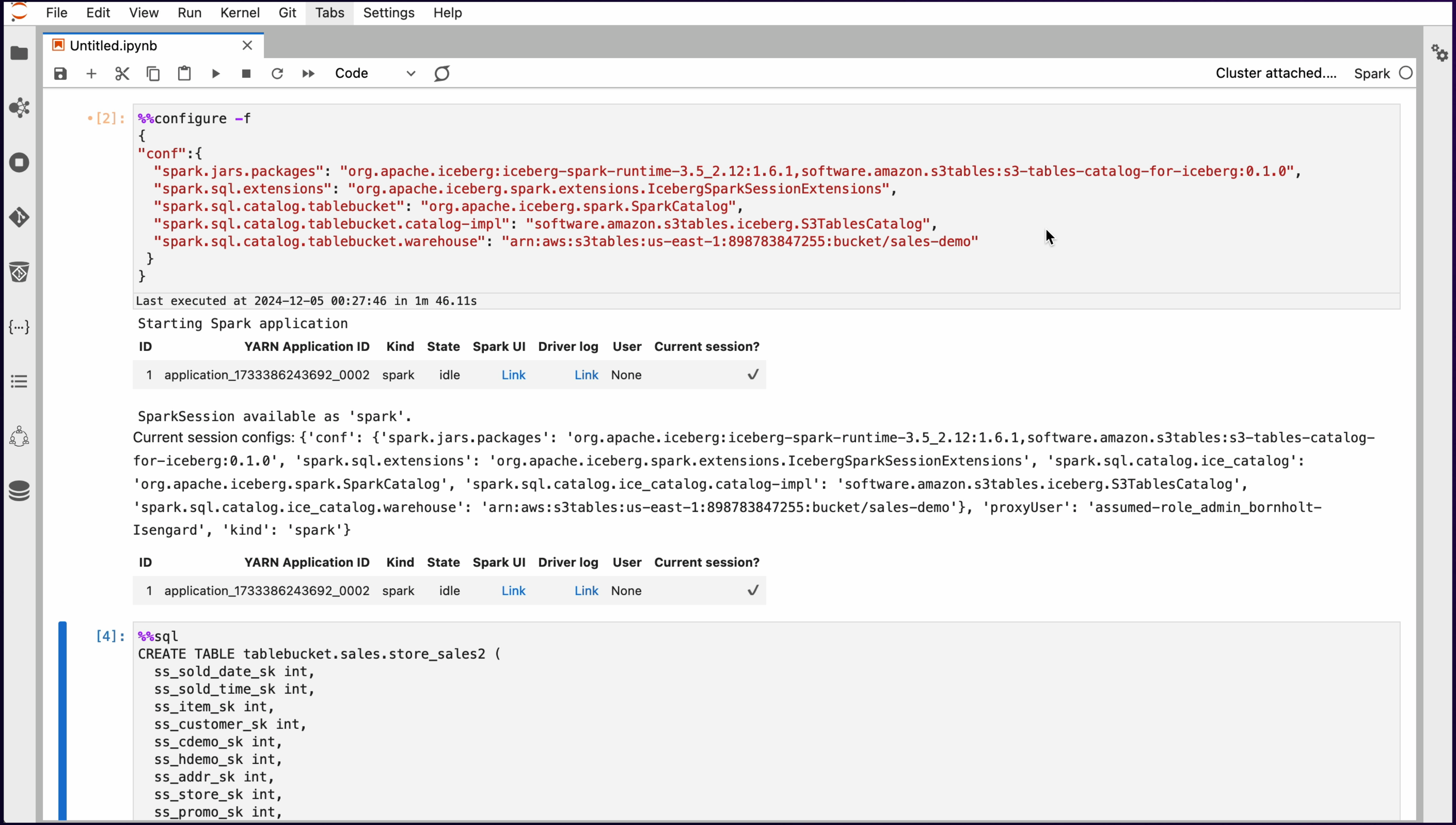
View pricing details

Resources

- User guide
- API reference
- FAQs

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Create a table and managing it using Amazon EMR



[2]:

```
%%configure -f
{
"conf":{
  "spark.jars.packages": "org.apache.iceberg:iceberg-spark-runtime-3.5_2.12:1.6.1,software.amazon.s3tables:s3-tables-catalog-for-iceberg:0.1.0",
  "spark.sql.extensions": "org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions",
  "spark.sql.catalog.tablebucket": "org.apache.iceberg.spark.SparkCatalog",
  "spark.sql.catalog.tablebucket.catalog-impl": "software.amazon.s3tables.iceberg.S3TablesCatalog",
  "spark.sql.catalog.tablebucket.warehouse": "arn:aws:s3tables:us-east-1:898783847255:bucket/sales-demo"
 }
}
```

```
Last executed at 2024-12-05 00:27:46 in 1m 46.11s
```

```
Starting Spark application
```

ID	YARN Application ID	Kind	State	Spark UI	Driver log	User	Current session?
1	application_1733386243692_0002	spark	idle	Link	Link	None	✓

```
SparkSession available as 'spark'.
```

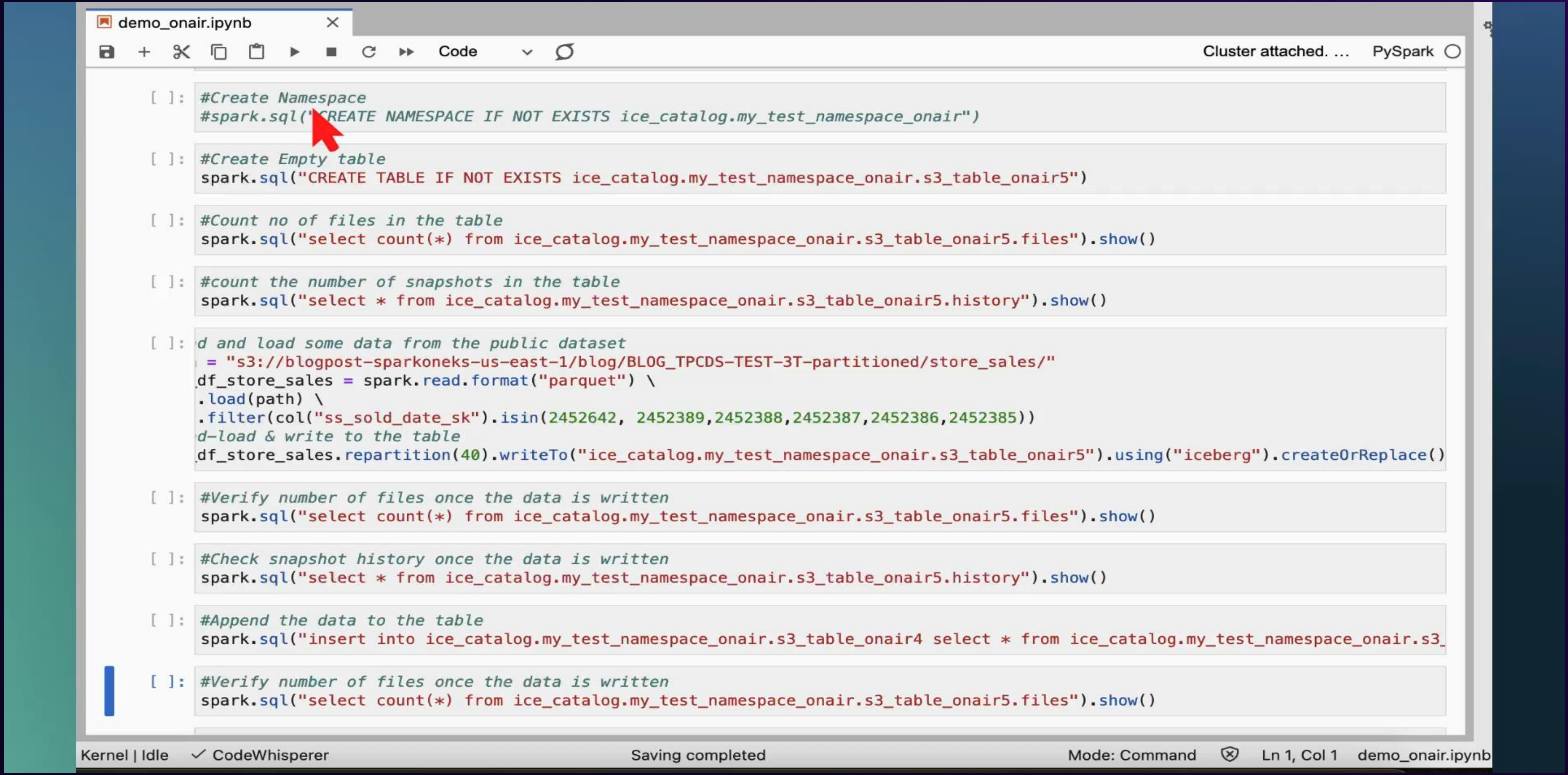
Current session configs: {'conf': {'spark.jars.packages': 'org.apache.iceberg:iceberg-spark-runtime-3.5_2.12:1.6.1,software.amazon.s3tables:s3-tables-catalog-for-iceberg:0.1.0', 'spark.sql.extensions': 'org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions', 'spark.sql.catalog.ice_catalog': 'org.apache.iceberg.spark.SparkCatalog', 'spark.sql.catalog.ice_catalog.catalog-impl': 'software.amazon.s3tables.iceberg.S3TablesCatalog', 'spark.sql.catalog.ice_catalog.warehouse': 'arn:aws:s3tables:us-east-1:898783847255:bucket/sales-demo'}, 'proxyUser': 'assumed-role_admin_bornholt-Isengard', 'kind': 'spark'}

ID	YARN Application ID	Kind	State	Spark UI	Driver log	User	Current session?
1	application_1733386243692_0002	spark	idle	Link	Link	None	✓

[4]:

```
%%sql
CREATE TABLE tablebucket.sales.store_sales2 (
  ss_sold_date_sk int,
  ss_sold_time_sk int,
  ss_item_sk int,
  ss_customer_sk int,
  ss_cdemo_sk int,
  ss_hdemo_sk int,
  ss_addr_sk int,
  ss_store_sk int,
  ss_promo_sk int,
```

Create a table and managing it using Amazon EMR



demo_onair.ipynb

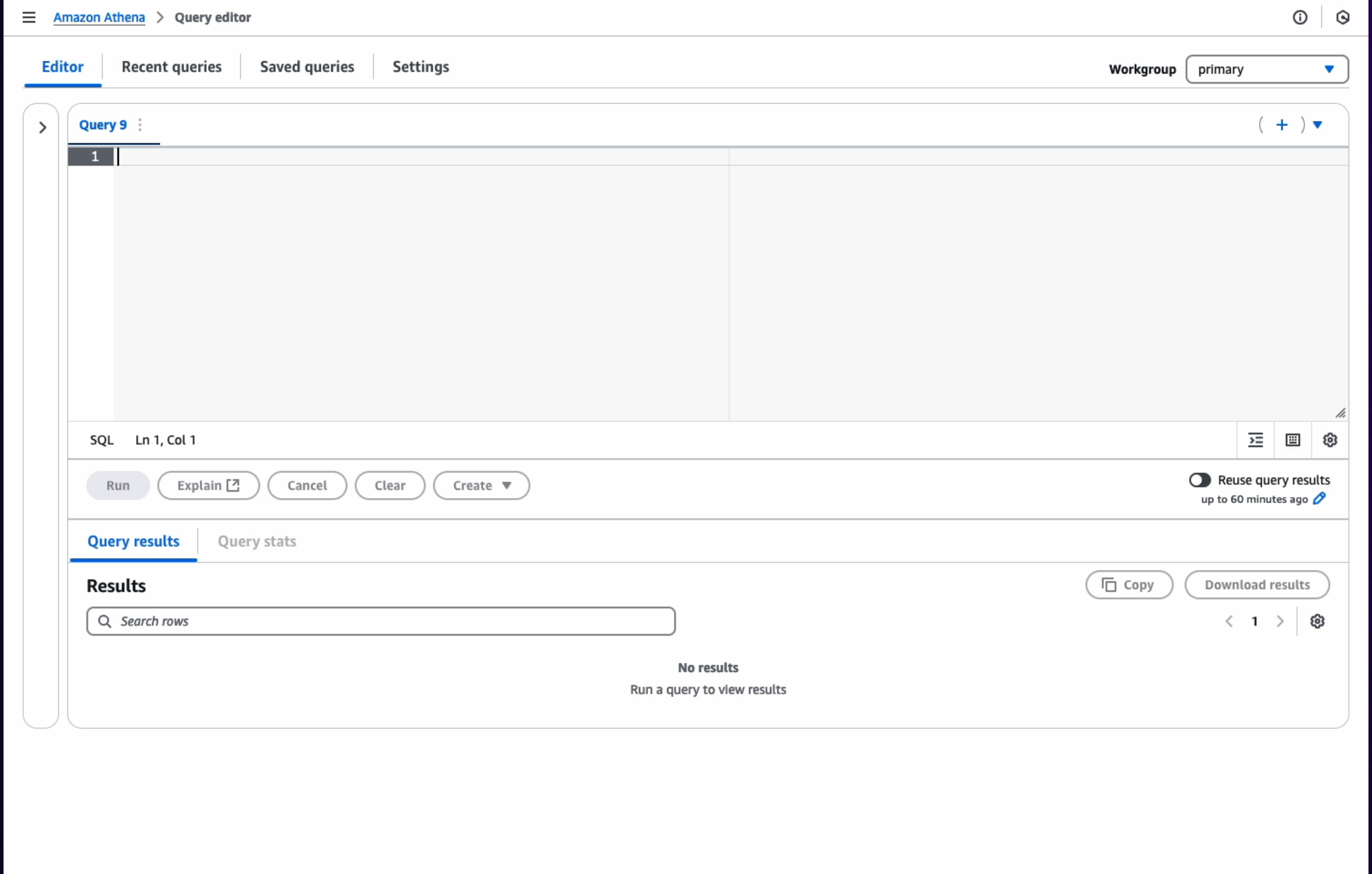
Code

Cluster attached. ... PySpark ○

```
[ ]: #Create Namespace
     #spark.sql("CREATE NAMESPACE IF NOT EXISTS ice_catalog.my_test_namespace_onair")

[ ]: #Create Empty table
     spark.sql("CREATE TABLE IF NOT EXISTS ice_catalog.my_test_namespace_onair.s3_table_onair5")

[ ]: #Count no of files in the table
     spark.sql("select count(*) from ice_catalog.my_test_namespace_onair.s3_table_onair5.files").show()

[ ]: #count the number of snapshots in the table
     spark.sql("select * from ice_catalog.my_test_namespace_onair.s3_table_onair5.history").show()

[ ]: d and load some data from the public dataset
      = "s3://blogpost-sparkoneks-us-east-1/blog/BLOG_TPCDS-TEST-3T-partitioned/store_sales/"
     df_store_sales = spark.read.format("parquet") \
     .load(path) \
     .filter(col("ss_sold_date_sk").isin(2452642, 2452389,2452388,2452387,2452386,2452385))
     d-load & write to the table
     df_store_sales.repartition(40).writeTo("ice_catalog.my_test_namespace_onair.s3_table_onair5").using("iceberg").createOrReplace()

[ ]: #Verify number of files once the data is written
     spark.sql("select count(*) from ice_catalog.my_test_namespace_onair.s3_table_onair5.files").show()

[ ]: #Check snapshot history once the data is written
     spark.sql("select * from ice_catalog.my_test_namespace_onair.s3_table_onair5.history").show()

[ ]: #Append the data to the table
     spark.sql("insert into ice_catalog.my_test_namespace_onair.s3_table_onair4 select * from ice_catalog.my_test_namespace_onair.s3_

[ ]: #Verify number of files once the data is written
     spark.sql("select count(*) from ice_catalog.my_test_namespace_onair.s3_table_onair5.files").show()
```

Kernel | Idle ✓ CodeWhisperer Saving completed Mode: Command Ln 1, Col 1 demo_onair.ipynb

Querying tables using Athena



Q&A

Thank you!

Please complete the session survey in the mobile app

David Lee

Principal Product Manager
Amazon S3
AWS

Prathiban Mohanasundaram

Senior Software Development Manager
Amazon S3
AWS