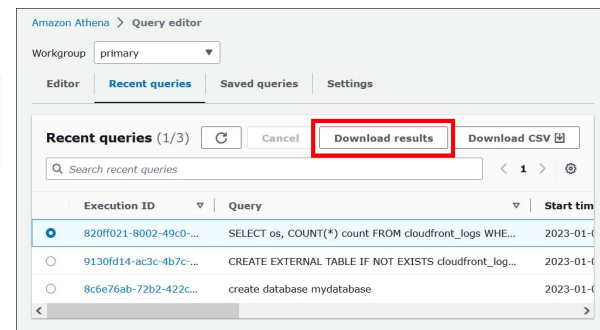

Amazon Athena

William Hay
November 25th, 2025

Amazon Athena

```
SELECT os, COUNT(*) count
FROM cloudfront_logs
WHERE date BETWEEN date '2014-07-05' AND date '2014-08-05'
GROUP BY os
```



What is it?

- Serverless - no infrastructure to manage
- SQL-based interactive query service for Amazon S3
- Uses Facebook Presto for the SQL query engine

What can you do?

- Using S3 and standard SQL, Athena helps analyze unstructured, semi-structured, and structured data that is stored in an S3 bucket
 - Unstructured data still needs to be defined through a schema
- Integration with Amazon QuickSight, Glue Data Catalog, Redshift, EMR
- Download recent queries for further analysis
 - Recent queries are retained for 45 days

History - Who

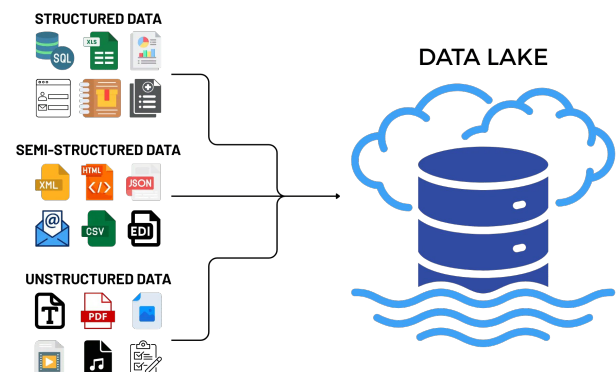
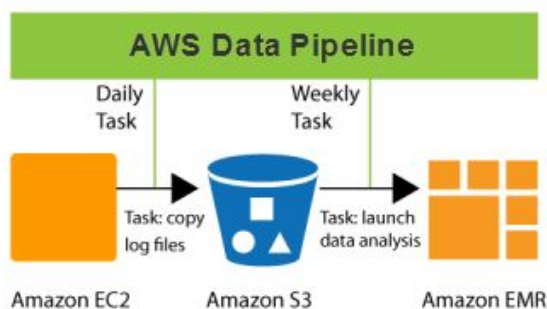
➤ General availability announced in 2016 at the AWS re:Invent event

Technology	Type	Storage	Pricing	Performance	Best For
Amazon Athena	Serverless SQL engine	S3	Per TB scanned	Varies depending on file format	Ad-hoc queries over S3 data lakes
Amazon Redshift	Warehouse	Internal + External	On-demand cluster	High, steady	High-volume analytics
Google BigQuery	Serverless warehouse	Internal + External	Per TB scanned	Very high, consistent	Large-scale data analytics
Snowflake	Warehouse	Internal + External	Compute + storage	Very high	Recurring analytics
Azure Synapse Analytics	Serverless "SQL pool"	Azure Data Lake Storage	Per TB scanned	High	Azure-native workloads

3

History - Why

- Massive growth of data stored in S3
- Need to analyze data without ETL or loading into databases
 - Before Athena, organizations analyzed large datasets by loading data into a database or build and maintain Hadoop or Spark clusters, requiring large amounts of maintenance
- Pay-as-you-go pricing, no idle computing costs
- Makes querying data lakes very easy

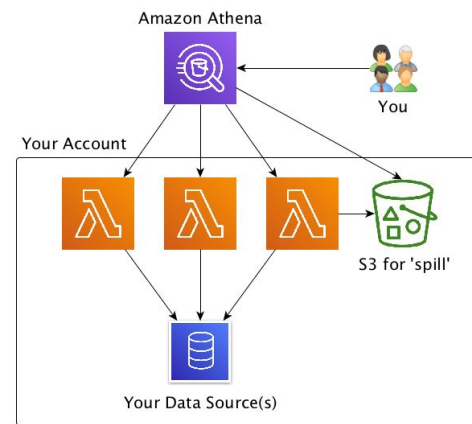


4

History - How

Since its release in 2016...

- Added support for columnar formats (e.g. Parquet)
- Integrated with AWS Glue Data Catalog
- Partition projection
 - Athena will understand S3 partitions logically
- Current engine is **Athena engine version 3**
- Support added for federated queries across multiple AWS data sources



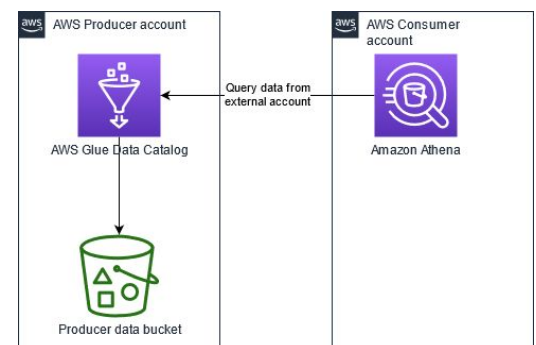
These changes were driven by a shift towards serverless analytics and increasing size of data lakes in Amazon S3.

5

Features - 1



- Serverless SQL query engine
- Queries data directly stored in Amazon S3
- Based on Facebook Presto distributed compute engine
- Supports structured and semi-structured formats
 - CSV, JSON, Parquet, and more...
- Integrates with multiple Amazon services (AWS Glue Data Catalog)
- Can query petabyte-scale datasets without provisioning any clusters
- Supports ANSI SQL
 - Window functions, complex joins, database and table creation, etc.
- Results can be exported to either S3 or into an ETL pipeline



6

Features - 2

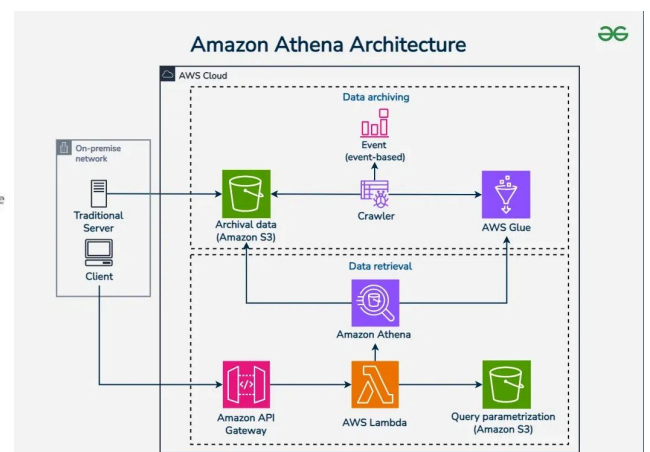
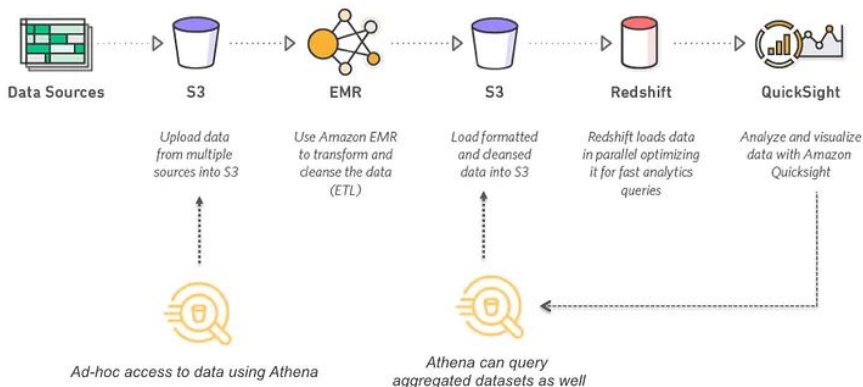
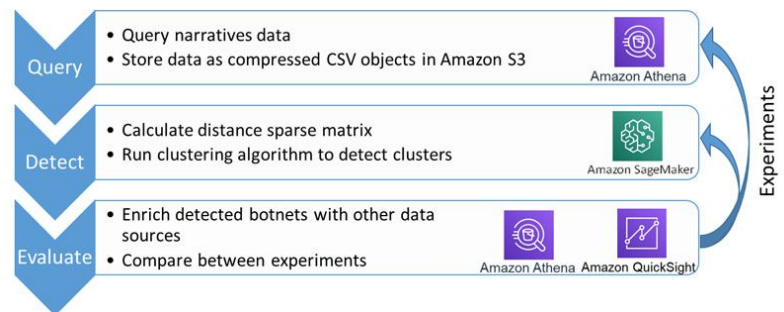
- Integrates with S3 lifecycle tiers for cost-optimized storage
- Ability to save queries
- Automatically executes queries in parallel
- Supports federated queries
- Invoke SageMaker machine learning models to use an Athena SQL query to run inference
 - Can use SQL experience to train ML models
- Ability to query from Azure Synapse Analytics data and visualize with Amazon QuickSight



7

Use Cases

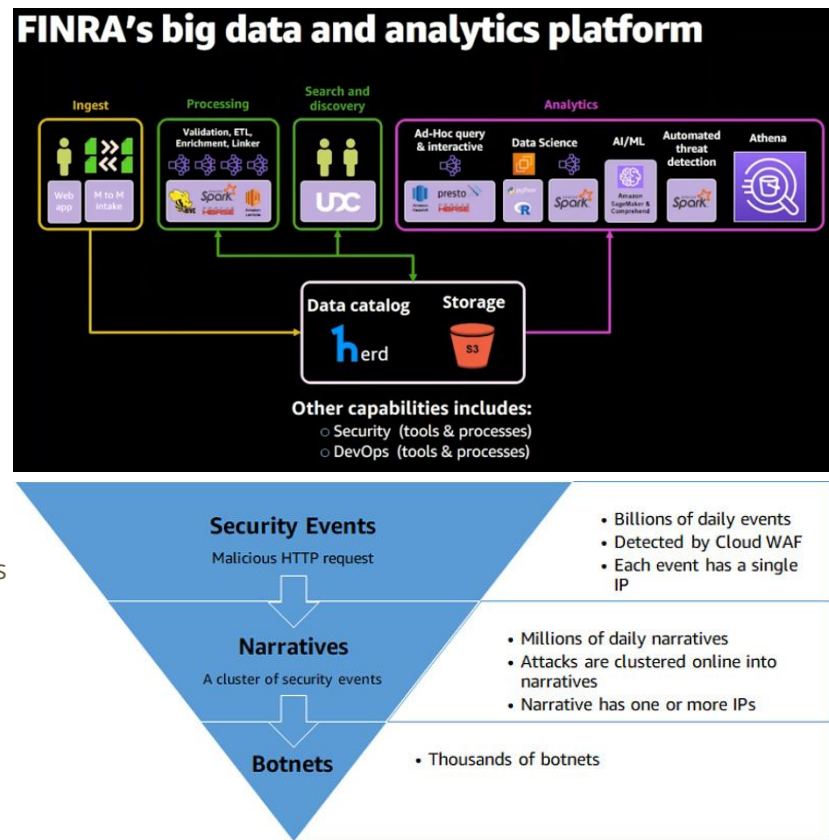
- Ad-hoc querying for analysis
- Build serverless data lake analysis
- Query information to display visually (e.g. using Amazon QuickSight)
- Machine learning botnet detection



8

Industry Deployments

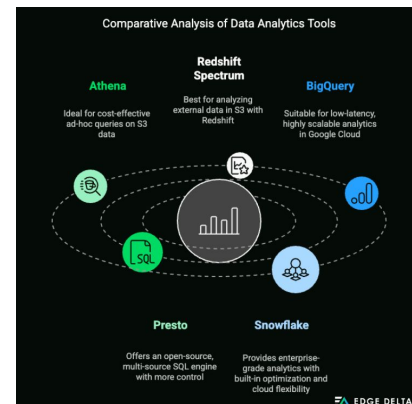
- **FINRA**
 - Processes 425+ billion market events each day (~10-20 TB of data)
 - Regulates 3,700 security firms and 630k+ brokers
 - Uses Athena for real time analysis
 - Athena eliminated problems with configuring and maintaining physical servers
- **Imperva**
 - Protects hundreds of thousands of websites
 - Blocks billions of security events each day
 - Uses Athena, SageMaker, and QuickSight to develop a ML clustering algorithm
 - Detects botnets, which contribute to attacks such as DDoS



9

Advantages

- Serverless
- Build interactive and advanced analytics applications
- Build with preferred tools and frameworks
- Simple and predictive pricing
- Pay only for data scanned
- Highly scalable
- Ideal for intermittent workloads
- Reduces operational costs and ETL costs



10

Disadvantages

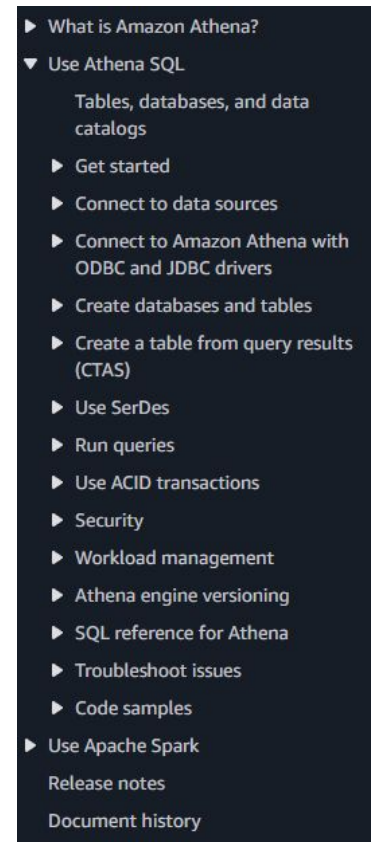
- Performance varies based on data layout within S3
- Can be costly if file compression is not used
- Not ideal for transactional workloads
- Concurrency limits
- Latency has been observed to be unpredictable
- Limited in terms of tuning compared to other options



11

Usability

- If you have SQL knowledge, the hardest part is the setup
 - Always double check to make sure Athena settings are configured correctly for whatever data formats you will be using
- To get Athena working at a minimum, an individual can set it up in about 5 minutes without much knowledge about Athena
- Lots of documentation, tutorials, and settings to mess around with



12

Cost



- Athena charges \$5 per TB of data scanned
- Save up to 90% per query and get better performance by compressing, partitioning, and converting data into columnar formats
- When using SQL queries on data sources, you are charged for the number of bytes per scanned query, rounded to the nearest megabyte
 - Minimum 10 MB per query unless Provisioned Capacity is used
 - Data definition language (DDL) statements have no charges
 - CREATE, ALTER, or DROP TABLE
- With 100k queries a day, with a minimum of 10MB: \$145.04 monthly
- With 1m queries a day, with a minimum of 100MB: \$14,503.80 monthly

13

Cost Example

FINRA:

Athena pricing

- Pay only for the queries you run vs. cost of maintaining long running cluster

Athena charges \$5 per TB of data scanned. For \$5 we can execute:

- 100,000 queries with an average of 10MB data scanning
- 10,000 queries with an average of 100MB data scanning
- 1,000 queries with an average of 1,000MB (1GB) data scanning

```
select rec_value
  from CATFOLA
 where trade_dt= date '2020-11-11'
    and cat_lifecycle_id = 1234567890
```

3.43 seconds
10.89 MB Datascan

Athena Benchmarks from Upsolver:

Athena vs BigQuery Benchmarks

	 Google BigQuery	 Amazon Athena Upsolver	 Amazon Athena 5-minute Parquet	 Amazon Athena 1-minute Parquet
Query Time (Seconds)	57.72	60.98 (+5.6%)	113.57 (+96.7%)	292.58 (+506%)
Total Cost (USD)	1.97	0.42 (-78.7%)	0.47 (-76.2%)	0.49 (-75.13%)

* Aggregated results for 9 SQL queries against NY Taxi Rides dataset

Sources used:

https://www.youtube.com/watch?v=72c_klMZdZ8

<https://www.upsolver.com/blog/benchmarking-aws-athena-bigquery-performance-price>

14

Amazon Athena Conclusion



Amazon Athena

- Serverless SQL query service for S3 analytics or ML inference
- Integrates with multiple AWS services
 - AWS Glue, CloudFormation, CloudTrail, EMR, S3, IAM
- Ideal for ad-hoc queries and scales to analyze petabyte-scale data
- Best cost-effective method is to optimize and compress data
- Automatically scales and completes queries in parallel
- Built to support standard SQL and federated queries
- Save queries and download results

15

Demo

To start, we create an S3 bucket to point Amazon Athena to our input/output source

Make sure that the user you are executing queries with has proper permissions (s3:GetObject & s3:ListBucket)

The screenshot shows the Amazon S3 console interface for a bucket named 'athena-demo-whay'. The 'Objects' tab is active, showing a list of objects. The table has columns for Name, Type, Last modified, Size, and Storage class. The objects listed are:

Name	Type	Last modified	Size	Storage class
20251124_174909_00160_wjgul_356b3a20-f13e-4e2c-a6c7-bc7d01f0ede2	-	November 24, 2025, 09:49:10 (UTC-08:00)	1000.0 B	Standard
athena-results/	Folder	-	-	-
web_events/	Folder	-	-	-

16

Editor Recent queries Saved queries **Query settings** Workgroup primary

Query settings

Query settings temporarily override your workgroup configurations. Clear all the query settings to enforce your workgroup configurations.

Query result encryption Manage

Query result location
s3://athena-demo-whay/athena-results/ [View](#)

Encrypt query results
-

Expected bucket owner
-

Assign bucket owner full control over query results
Turned off

Execution controls Info Edit controls

Set limits and control query execution behavior for the workgroup. Controls apply to all queries in the workgroup unless overridden.

Search execution controls

Control Name	Value	Unit
Data scanned limit	-	-
Min DPU per query	-	-
Max DPU per query	-	-

Manage query result location and encryption

Location of query result - optional
Enter an S3 prefix in the current region where the query results will be saved as an object.

Q s3://athena-demo-whay/athe [View](#) [Browse S3](#)

You can create and manage lifecycle rules for this bucket
Use Amazon S3 lifecycle rules to store your query results and metadata cost effectively or to delete them after a period of time. [Learn more](#)

[Lifecycle configuration](#)

Expected bucket owner - optional
Specify the AWS account ID that you expect to be the owner of your query results output location bucket.

Enter AWS account ID

☐ Assign bucket owner full control over query results
Enabling this option grants the owner of the S3 query results bucket full control over the query results. This means that if your query result location is owned by another account, you grant full control over your query results to the other account.

☐ Encrypt query results

[Cancel](#) [Save](#)

Within Amazon Athena, go to “Query settings” and select “Manage.”

This will take you to a screen where you can select the **output** of your Athena queries.

Other than selecting the S3 bucket, no other options are used.

17

Amazon Athena > Query editor

Query in Amazon SageMaker Unified Studio
Use your IAM role to analyze and build with your existing AWS resources

Editor Recent queries Saved queries Query settings

Data

Data source
AwsDataCatalog

Catalog
None

Database
demo_db

Tables and views [Create](#)

Filter tables and views

Tables (2)

Views (0)

Create a table from data source

- S3 bucket data
- AWS Glue Crawler

Create with SQL

- CREATE TABLE
- CREATE TABLE AS SELECT
- CREATE TABLE AS SELECT(ICEBERG)
- CREATE VIEW

Query 1

```

1 CREATE TABLE demo_db.web_events
2 WITH (
3   format = 'PARQUET',
4   external_location = 's3://athena-demo-whay/web_events'
5 ) AS
6 SELECT *
7 FROM web_events
8 WHERE session_duration < 100

```

Create table from S3 bucket data

Table details

Table name
web_events

Description - optional
Type something

Database configuration

Choose an existing database or create a new database
Choose to access an existing database or to create a new table. Athena stores the table schema in the AWS Glue Data Catalog.

☒ Create a database
☐ Choose an existing database

Database name
demo_db

Dataset

Location of input data set
Q s3://athena-demo-whay [View](#) [Browse S3](#)

Data format

Table type
Apache Hive

File format
CSV

Serde library
org.apache.hadoop.hive.serde2.LazySimpleSerDe

Serde properties - optional

Name
Field.delim

Value
,

[Add SerDe property](#)

Select an S3 bucket as the **input** for your Athena queries.

18

user_id	event_time	page	device	country	session_duration
1	2025-11-20T10:01:00Z	/home	desktop	US	120
2	2025-11-20T10:02:15Z	/products	mobile	CA	45
3	2025-11-20T10:03:10Z	/cart	mobile	US	300
4	2025-11-20T10:05:27Z	/home	desktop	GB	60
5	2025-11-20T10:06:55Z	/checkout	tablet	AU	210
6	2025-11-20T10:08:33Z	/products	mobile	US	95
7	2025-11-20T10:10:42Z	/home	desktop	US	180
8	2025-11-20T10:12:01Z	/checkout	mobile	IN	340
9	2025-11-20T10:14:29Z	/home	tablet	CA	75
10	2025-11-20T10:17:58Z	/cart	mobile	US	420

Make sure to insert your test file into your input S3 bucket!

Athena does **not** automatically infer a schema unless you use something like AWS Glue.

Under “Column details” after selecting our input, we can specify a schema that we want Athena to use.

AI was used to generate the test file.

19

Column details

Column name must be from 1-128 characters. Valid characters are a-z, A-Z, 0-9, _ (underscore). Certain advanced column types (namely, structs) are not exposed in this interface.

Column name	Column type	Description - optional	
user_id	int	Enter description	<button>Remove</button>
event_time	timestamp	Enter description	<button>Remove</button>
page	string	Enter description	<button>Remove</button>
device	string	Enter description	<button>Remove</button>
country	string	Enter description	<button>Remove</button>
session_duration	int	Enter description	<button>Remove</button>

Add a column
Bulk add columns

Amazon Athena

Query editor

Query in Amazon SageMaker Unified Studio

Use your IAM role to analyze and build with your existing AWS resources in a single data and AI development environment.

Editor

Recent queries

Saved queries

Query settings

Workgroup

primary

Data

Data source

AwsDataCatalog

Catalog

None

Database

default

Tables and views

Create

Filter tables and views

Tables (0)

Views (0)

Query 1

Query 2

```

1 CREATE EXTERNAL TABLE IF NOT EXISTS 'demo_db`.`web_events` (
2   'user_id' int,
3   'event_time' timestamp,
4   'page' string,
5   'device' string,
6   'country' string,
7   'session_duration' int
8 )
9 ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
10 WITH SERDEPROPERTIES ('field.delim' = ';')
11 STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.TextInputFormat' OUTPUTFORMAT 'org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat'
12 LOCATION 's3://athena-demo-uhay/'
13 TBLPROPERTIES ('classification' = 'csv');

```

SQL

Ln 1, Col 1

Run again

Explain (P)

Cancel

Clear

Create

Query results

Query stats

Completed

Time in queue: 88 ms

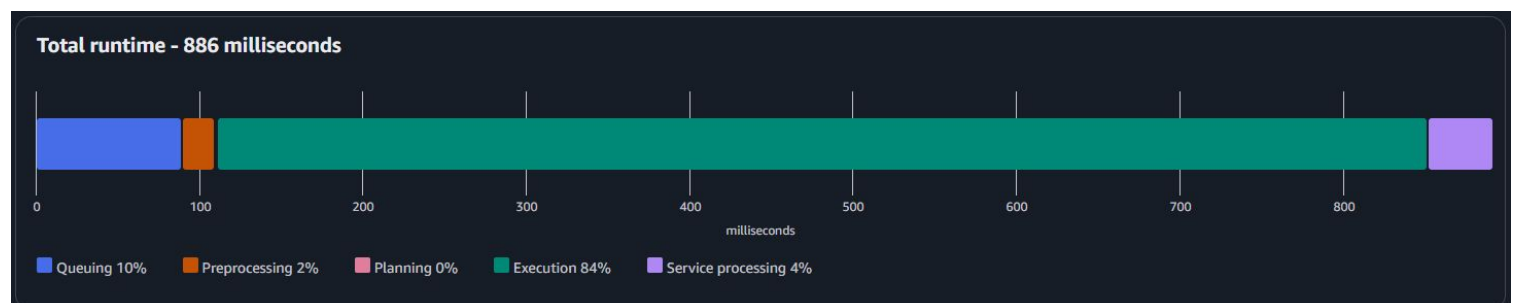
Run time: 740 ms

Data scanned: -

Query successful.

Once you’ve selected “Create table,” you should see an automatically generated query to create the table you specified.

You can also see query stats!



20

```

1 SELECT page, AVG(session_duration) AS avg_session_seconds
2 FROM web_events
3 GROUP BY page
4 ORDER BY avg_session_seconds DESC;

```

Query results Query stats

Completed Time in queue: 109 ms Run time: 471 ms Data scanned: 0.50 KB

Results (4)

Search rows

#	page	avg_session_seconds
1	/cart	360.0
2	/checkout	275.0
3	/home	108.75
4	/products	70.0

```

1 CREATE TABLE demo_db.long_sessions
2 WITH (
3   format = 'PARQUET',
4   external_location = 's3://athena-demo-whay/web_events_parquet'
5 ) AS
6 SELECT *
7 FROM web_events
8 WHERE session_duration > 300;

```

Objects (4)

Objects are the fundamental entities stored in Amazon S3. You can

Find objects by prefix

☐	Name	Type
☐	20251124_174909_00160_wjgqj_356b3a2-0-f13e-4e2c-a6c7-bc7d01f0ede2	-
☐	athena-results/	Folder
☐	web_events_parquet/	Folder
☐	web_events/	Folder

Query results Query stats

Completed Time in queue: 106 ms Run time: 502 ms Data scanned: 0.33 KB

Results (2)

Search rows

#	user_id	event_time	page	device	country	session_duration
1	8		/checkout	mobile	IN	340
2	10		/cart	mobile	US	420

Lets try some queries!

You can create multiple query tabs to save and rerun any queries

Athena can also write files with a specific format, capturing any query and storing it in S3 using that file format

As Athena also supports other formats than CSV, you can also run queries on Parquet files

Sources used: <https://docs.aws.amazon.com/athena/latest/ug/parquet-serde.html>

21

Questions?

22