Building Data Lakes on AWS –

Lab 1: Build a Data Lake Using AWS Lake Formation

Lab overview

You are a data engineer at Any Company, a cloud marketing organization. You are asked to run a pilot project to build a data lake using AWS Lake Formation.

In this lab, you work through the basic steps to register storage with Lake Formation and define a crawler to crawl the initial dataset. You then create an AWS Glue job to generate a version of the dataset in a partitioned Apache Parquet format. You also use Amazon Athena to validate the tables, their schema, and the ability to access the underlying data source through the data catalog.

OBJECTIVES

By the end of this lab, you will be able to do the following:

- Create a data lake and a database.
- Crawl data with AWS Glue to create a table.
- Query data using Athena.
- Transform data from .csv to Parquet format.

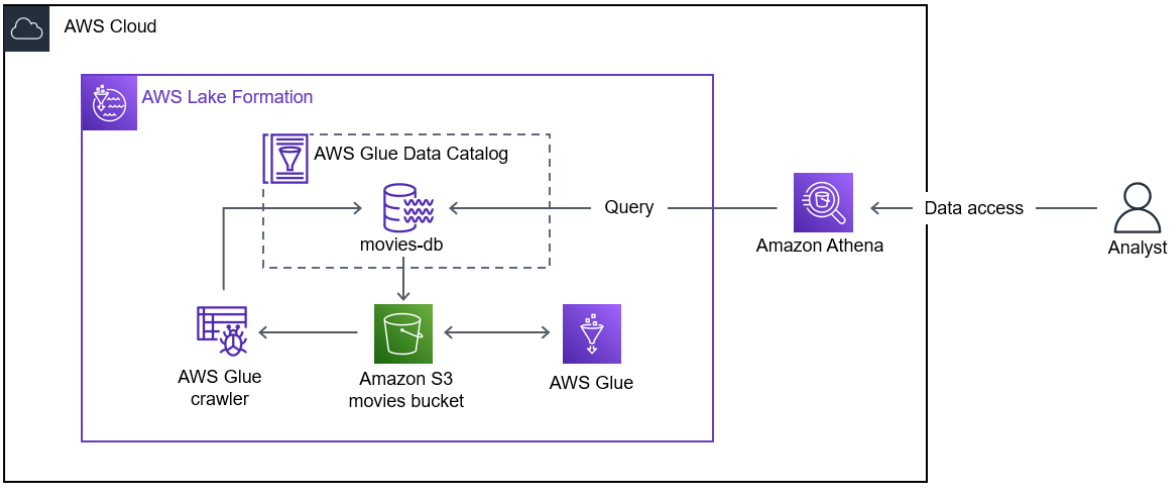
ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

- **Note:** A hint, tip, or important guidance.
- **Learn more:** Where to find more information.
- **Consider:** A moment to pause to consider how you might apply a concept in your own environment or to initiate a conversation about the topic at hand.
- **Hint:** A hint to a question or challenge.
- **Answer:** An answer to a question or challenge.

ENVIRONMENT OVERVIEW

The following diagram shows the basic architecture of the lab environment:



In the preceding diagram, an Amazon Simple Storage Service (Amazon S3) bucket containing movies data is registered to Lake Formation to create a data lake. The data is loaded in AWS Glue. A crawler transforms the data from comma-separated values (.csv) to Parquet format and adds it to the database. The database is connected to the Amazon S3 bucket. An analyst is shown querying the Lake Formation database with Athena. Athena uses the AWS Glue Data Catalog and reads data from the database.

Start lab

To launch the lab, at the top of the page, choose Start lab.

Caution: You must wait for the provisioned AWS services to be ready before you can continue.

To open the lab, choose Open Console.

You are automatically signed in to the AWS Management Console in a new web browser tab.

WARNING: Do not change the Region unless instructed.

COMMON SIGN-IN ERRORS

Error: You must first sign out

If you see the message, **You must first log out before logging into a different AWS account:**

Choose the **click here** link.

Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.

Choose Open Console again.

Error: Choosing Start Lab has no effect

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

Add the lab domain name to your pop-up or script blocker's allow list or turn it off.

Refresh the page and try again.

Task 1: Explore the lab environment

In this task, you review the account resources created before the lab started.

At the top of the AWS Management Console, in the search bar, search for and choose

S3.

Choose the link for the bucket name that starts with **databucket**.

Note: The **data/** folder contains your dataset. The **results/** folder stores the results of your Athena queries. You will specify the **results/** folder as the query results location in Athena later in this lab.

Choose the **data/** folder. You will specify the **data/** folder as your designated storage location for your data lake later in this lab.

Choose the **movies_csv/** folder.

Note: There is one .csv file in the folder named **movies.csv**. It contains movie data from 1920-2018.

The file contains data similar to the following:

year	title	directors_0	rating	genres_0	genres_1	rank	running_time_secs	actors_0	actors_1	actors_2	directors_1	directors_2
2013	Rush	Ron Howard	8.3	Action	Biography	2	7380	Daniel Bruhl	Chris Hemsworth	Olivia Wilde		
2013	Prisoners	Denis Villeneuve	8.2	Crime	Drama	3	9180	Hugh Jackman	Jake Gyllenhaal	Viola Davis		
2013	The Hunger Games: Catching Fire	Francis Lawrence		Action	Adventure	4	8760	Jennifer Lawrence	Josh Hutcherson	Liam Hemsworth		
2013	Thor: The Dark World	Alan Taylor		Action	Adventure	5		Chris Hemsworth	Natalie Portman	Tom Hiddleston		
2013	This Is the End	Evan Goldberg	7.2	Comedy	Fantasy	6	6420	James Franco	Jonah Hill	Seth Rogen	Seth Rogen	

Congratulations! You successfully explored the lab environment.

Task 2: Set up Lake Formation

In this task, you register your Amazon S3 data storage and create a database.

TASK 2.1: REGISTER YOUR AMAZON S3 STORAGE

Lake Formation manages access to designated storage locations in Amazon S3. Register the storage locations you want to be part of the data lake.

At the top of the AWS Management Console, in the search bar, search for and choose

AWS Lake Formation.

In the **Welcome to Lake Formation** popup window, make sure **Add myself** is selected and then choose Get started.

Note: If you see **Read Only Admin is not supported.**

Go to settings under **Data catalog** and choose Get started.

choose Save.

In the left navigation pane, in the **Administration** section, choose **Data Lake locations**.

Choose Register location.

On the **Register location** page, in the **Amazon S3 location** section:

For **Amazon S3 path**, copy and paste the **SourceDataLocation** value that is listed to the left of these instructions.

For **IAM role**, select **LakeFormationServiceRole**.

Note: The **LakeFormationService** role was created at the start of the lab and includes the **s3: PutObject**, **s3: GetObject**, **s3: DeleteObject**, and **s3: ListBucket** actions for your **DataBucket**.

Choose Register location.

In the left navigation pane, in the **Permissions** section, choose **Data locations**.

Choose Grant.

On the **Grant permissions** page, configure the following:

For **IAM users and roles**, select **AdminGlueServiceRole**.

For **Storage locations**, copy and paste the **SourceDataLocation** value that is listed to the left of these instructions.

Choose Grant.

Your Amazon S3 bucket **data/** folder is now registered as the storage location for your data lake.

TASK 2.2: CREATE A DATABASE

Lake Formation uses the AWS Glue Data Catalog to store metadata about data lakes, data sources, transforms, and targets. Metadata about data sources and targets is in the form of databases and tables. Tables store information about the underlying data, including schema information, partition information, and data location. Databases are collections of tables.

Create a database in the AWS Glue Data Catalog.

In the left navigation pane, in the **Data catalog** section, choose **Databases**.

Choose Create database.

On the **Create database** page, in the **Database details** section:

For **name**, enter

```
movies-db
```

.

For **Location**, copy and paste the **SourceDataLocation** value that is listed to the left of these instructions.

Choose Create database.

You created a database in the AWS Glue Data Catalog using Lake Formation.

Congratulations! You successfully registered your Amazon S3 data storage and created a database.

Task 3: Crawl data with AWS Glue

In this task, you use an AWS Glue crawler to create a table for your movie data in your Lake Formation database.

TASK 3.1: CREATE A CRAWLER

A crawler connects to a data store and progresses through a prioritized list of classifiers to determine the schema for your data. It then creates metadata tables in your data catalog.

The crawler reads data at the source location and creates tables in the AWS Glue Data Catalog. A table is the metadata definition that represents your data, including its schema. The tables in the AWS Glue Data Catalog do not contain data. Instead, you use these tables as a source or target in an AWS Glue job definition.

At the top of the AWS Management Console, in the search bar, search for and choose

AWS Glue.

In the left navigation pane, in the **Data catalog** section, choose **Tables**.

Choose Add tables using crawler.

For **Name**, enter

movies-table.

Choose Next.

In the **Data sources** section, choose Add a data source.

For **S3 path**, copy and paste the **SourceDataLocation** value that is listed to the left of these instructions.

Note: With the default settings, the crawler will crawl all subfolders. The crawler will find the **movies.csv** file in the **movies_csv/** subfolder when it runs.

Choose Add an S3 data source.

Choose Next.

For **Existing IAM role**, choose **AdminGlueServiceRole**.

Choose Next.

For **Target database**, select **movies-db**.

Choose Next.

Choose Create crawler.

You created an AWS Glue crawler you can use to add data to your table.

TASK 3.2: RUN THE CRAWLER

Run your crawler to add data to your table.

Choose Run crawler.

The crawling task can take a few minutes to complete.

In the left navigation pane, in the **Data catalog** section, choose **Crawlers**.

Choose the refresh icon to get the current crawler status.

The **movies-table** status changes from **RUNNING** to **STOPPING** to **READY**.

Wait until the **State** is **Completed**.

You have run the crawler and added data to your table.

TASK 3.3: VIEW THE TABLE SCHEMA

Review the AWS Glue logs and view the new table schema created by the crawler.

Choose **View Cloudwatch logs** to review the logs in Amazon CloudWatch, including logs related to data classification and table creation.

Consider: Take a moment to view the logs. Which event confirms the partitions for the table were created?

Return to the AWS Glue Console tab.

In the left navigation pane, in the **Data catalog** section, choose **Tables**.

Choose the refresh icon to get the current list of tables.

You see a **data** table.

Note: Since there is only one folder and file in the **data** folder, when the crawler runs it creates a table with the root folder name.

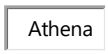
On the **Tables** page, choose the **data** table to see its schema.

Consider: Do the columns listed in the **Schema** section match the table sample shown in Task 1?

TASK 3.4: RUN A QUERY

Run a query in Athena to view a sample of the table.

At the top of the AWS Management Console, in the search bar, search for and choose



Note: If the Athena **Get Started** menu appears, choose **Query your data with Trino SQL** option and then choose Launch query editor.

In the **Workgroup primary settings** window, choose Acknowledge.

Note: The workgroup stores the Athena results in an Amazon S3 bucket. The primary workgroup saves the results to your data bucket in a *results/* folder.

In the **Data** section, you see the **AWSDataCatalog** data source and **movies-db** database automatically selected. In the **Tables** section, you see the **data** table listed.

In the query editor, enter the following:

```
SELECT * FROM "data" limit 10;
```

Choose Run.

In the **Results** section, you see 10 records from the **movies** table.

CHALLENGE A: CUSTOMIZE YOUR QUERY

AnyCompany wants to confirm the table contains the expected records. They know there are **1,002** movies with **Action** as the primary genre. Run a new query that counts the number of movies with a **genres_0** of **Action**.

Hint: Use the plus sign **+** to create a new query. Count all the records where **genres_0='Action'**.

Answer: Navigate **here** for a solution.

Congratulations! You successfully crawled data with AWS Glue, viewed logs in CloudWatch, and verified the results in Athena.

Task 4: Transform data using AWS Glue

AWS Glue provides a set of built-in transforms you can use to process your data. You can call these transforms from your extract, transform, and load (ETL) script. Your data passes from transform to transform in a data structure called a DynamicFrame, which is an extension to an Apache Spark SQL DataFrame. The DynamicFrame contains your data, and you reference its schema to process your data.

Learn more: Refer to *AWS Glue PySpark transforms reference* in the *Additional resources* section for more information.

In this task, you transform your data in .csv format to Parquet format. After completing the transformation, you run your crawlers to add the converted data to the new table. Finally, you compare the query performance using the csv file and Parquet data sources.

TASK 4.1: CREATE AN AWS GLUE JOB TO TRANSFORM DATA

Create an AWS Glue job to transform the data to the Parquet format.

At the top of the AWS Management Console, in the search bar, search for and choose

`AWS Glue`.

In the left navigation pane, in the **Data Integration and ETL** section, choose **ETL Jobs**.

In the **Create job** section, choose Visual ETL.

The AWS Glue Studio visual editor opens.

First, create the source using the AWS Glue Data Catalog.

In the **Add nodes** pane, under **Sources** tab, choose **AWS Glue Data Catalog**.

In the editor canvas, choose the **AWS Glue Data Catalog** node.

In the **Data source properties - Data Catalog** tab:

For **Database**, choose **movies-db**.

For **Table**, choose **data**.

Next, create the target bucket node.

At the upper-left side of the editor canvas, choose plus **+** button to open **Add nodes** pane.

In the **Add nodes** pane, under **Targets** tab, choose **Amazon S3**.

In the editor canvas, choose the **Data target - S3 bucket** node.

In the **Data target properties - S3** tab:

For **Format**, choose **Parquet**.

For **Compression Type**, choose **Uncompressed**.

For **S3 Target Location**, copy and paste the **ParquetDataLocation** value that is listed to the left of these instructions.

Lastly, update the AWS Glue job name, role, and script name.

In the node toolbar, choose **Job details**.

In the **Basic properties** section:

For **Name**, enter

`ParquetConversion`.

For **IAM Role**, choose **AdminGlueServiceRole**.

Expand **Advanced properties**, for **Script filename**, enter

ParquetConversion.py.

Choose Save.

Choose Run.

Choose the **Runs** tab to monitor the AWS Glue job status.

Choose the refresh icon to get the current AWS Glue job status.

Note: There are currently 10 workers using the worker type G.1X. For a simple conversion AWS Glue job, 10 workers are more than enough. When setting up an AWS Glue job, you can customize the number of workers and the worker type to fit your use case.

The AWS Glue job run status changes from **Running** to **Succeeded**.

Wait until the **Run status** is **Succeeded**.

You created an AWS Glue job to transform data.

TASK 4.2: RUN CRAWLERS TO ADD PARQUET DATA IN YOUR TABLE

In this task you run your existing crawler to add the table for the newly transformed data.

At the top of the AWS Management Console, in the search bar, search for and choose

AWS Glue.

In the left navigation pane, in the **Data Catalog** section, choose **Crawlers**.

Select the **movies-table** crawler link, and then choose Run crawler.

In the left navigation pane, in the **Data catalog** section, choose **Crawlers**.

Choose the refresh icon to get the current crawler status.

The **movies-table** status changes from **RUNNING** to **STOPPING** to **READY**.

Wait until the **State** is **Completed**.

At the top of the AWS Management Console, in the search bar, search for and choose

AWS Lake Formation.

In the left navigation pane, in the **Data catalog** section, choose **Tables**.

You see three tables:

movies_parquet

movies_csv

data

On the **Tables** page, choose the **movies_parquet** table to see its details.

Consider: There is a significant time and size complexity difference between querying the data using a .csv file as a datasource and using the columnar storage format of Parquet.

TASK 4.3: COMPARE ATHENA QUERY PERFORMANCE FOR EACH DATA SOURCE

In this task you run queries using the csv file and Parquet format to comare the query performance results.

At the top of the AWS Management Console, in the search bar, search for and choose

Athena.

In the **Data** section, you see the **AWSDataCatalog** data source and **movies-db** database automatically selected. In the **Tables** section, you see the **data** table listed.

Choose the plus sign + to create a new query using the csv file data source.

In the query editor, enter the following:

```
SELECT Count(*) AS action_movies FROM "movies_csv" where genres_0='Action';
```

Choose Run.

In the **Query Results** tab, note the **Time in queue**, **Run time** and **Data scanned** values. Also note there are 1002 results.

Choose the plus sign + to create a new query. This time, use the Parquet data source.

In the query editor, enter the following:

```
SELECT Count(*) AS action_movies FROM "movies_parquet" where genres_0='Action';
```

Choose Run.

In the **Query Results** tab, note the **Time in queue**, **Run time**, and **Data scanned** values and compare to the previous query values. While the **Time in queue** and **Run time** are comparable, the **Data scanned** size is significantly smaller for the Parquet data source due to the columnar format of the data structure, despite the same results count.

Consider: It is important to consider the impact data size has on query performance and time complexity. What would happen if you queried a million movies?

Congratulations! You have successfully transformed your data from .csv to Parquet format, run crawlers to add the converted Parquet data to a new table, and compared the Athena queries using csv file and Parquet data sources.

Conclusion

Congratulations! You now have successfully:

Created a data lake and a database.

Crawled data with AWS Glue to create a table.

Queried data using Athena.

Transformed data from .csv to Parquet format.

End lab

Follow these steps to close the console and end your lab.

Return to the **AWS Management Console**.

At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.

Choose End lab and then confirm that you want to end your lab.

Additional resources

AWS Glue PySpark transforms reference

Appendix

CHALLENGE A SOLUTION

Query your table for all the movies with a primary genre of **Action**.

Choose the plus sign + to create a new query.

In the query editor, enter the following:

```
SELECT COUNT(*) AS action_movies FROM "data" WHERE genres_0='Action';
```

Choose Run.

In the **Results** section, you see **1002** returned as the count for **action_movies**.

You successfully queried all the movies with a primary genre of **Action**.

To continue this lab, go to Task 4.