AWS Glue and Athena

AWS Glue: AWS Glue is a serverless data integration service that makes it easy for analytics users to discover, prepare, move, and integrate data from multiple sources. You can use it for analytics, machine learning, and application development. It also includes additional productivity and data ops tooling for authoring, running jobs, and implementing business workflows.

With AWS Glue, you can discover and connect to more than 70 diverse data sources and manage your data in a centralized data catalog. You can visually create, run, and monitor extract, transform, and load (ETL) pipelines to load data into your data lakes. Also, you can immediately search and query cataloged data using Amazon Athena, Amazon EMR, and Amazon Redshift Spectrum.

AWS Glue consolidates major data integration capabilities into a single service. These include data discovery, modern ETL, cleansing, transforming, and centralized cataloging. It's also serverless, which means there's no infrastructure to manage. With flexible support for all workloads like ETL, ELT, and streaming in one service, AWS Glue supports users across various workloads and types of users.

Also, AWS Glue makes it easy to integrate data across your architecture. It integrates with AWS analytics services and Amazon S3 data lakes. AWS Glue has integration interfaces and job-authoring tools that are easy to use for all users, from developers to business users, with tailored solutions for varied technical skill sets. Learn More

Athena: Amazon Athena is an interactive query service that makes it easy to analyze data directly in Amazon Simple Storage Service (Amazon S3) using standard SQL. With a few actions in the AWS Management Console, you can point Athena at your data stored in Amazon S3 and begin using standard SQL to run ad-hoc queries and get results in seconds.

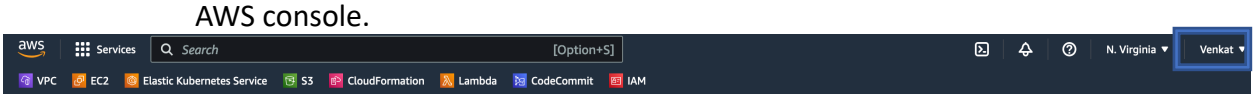
Amazon Athena also makes it easy to interactively run data analytics using Apache Spark without having to plan for, configure, or manage resources. When you run Apache Spark applications on Athena, you submit Spark code for processing and receive the results directly. Use the simplified notebook experience in Amazon Athena console to develop Apache Spark applications using Python or Athena notebook APIs.

Athena SQL and Apache Spark on Amazon Athena are serverless, so there is no infrastructure to set up or manage, and you pay only for the queries you run. Athena scales automatically—running queries in parallel—so results are fast, even with large datasets and complex queries. Learn more.

AWS Glue and Athena

Points to Remember before doing this Lab:

1. Please ensure that you attach all the screenshots labeled with "Note: This is a Deliverable" under each of them.
2. When capturing each screenshot, be certain that your AWS account name (It should be your NETId) is visible. Locate the name at the top-right corner of your AWS console.



Pre-requisite for this lab:

a.) Your file and folder names should start with <name>_<resource_name>.
b.) All labs must be performed in US East (N.Virginia) us-east-1 region.

Learning Outcome:

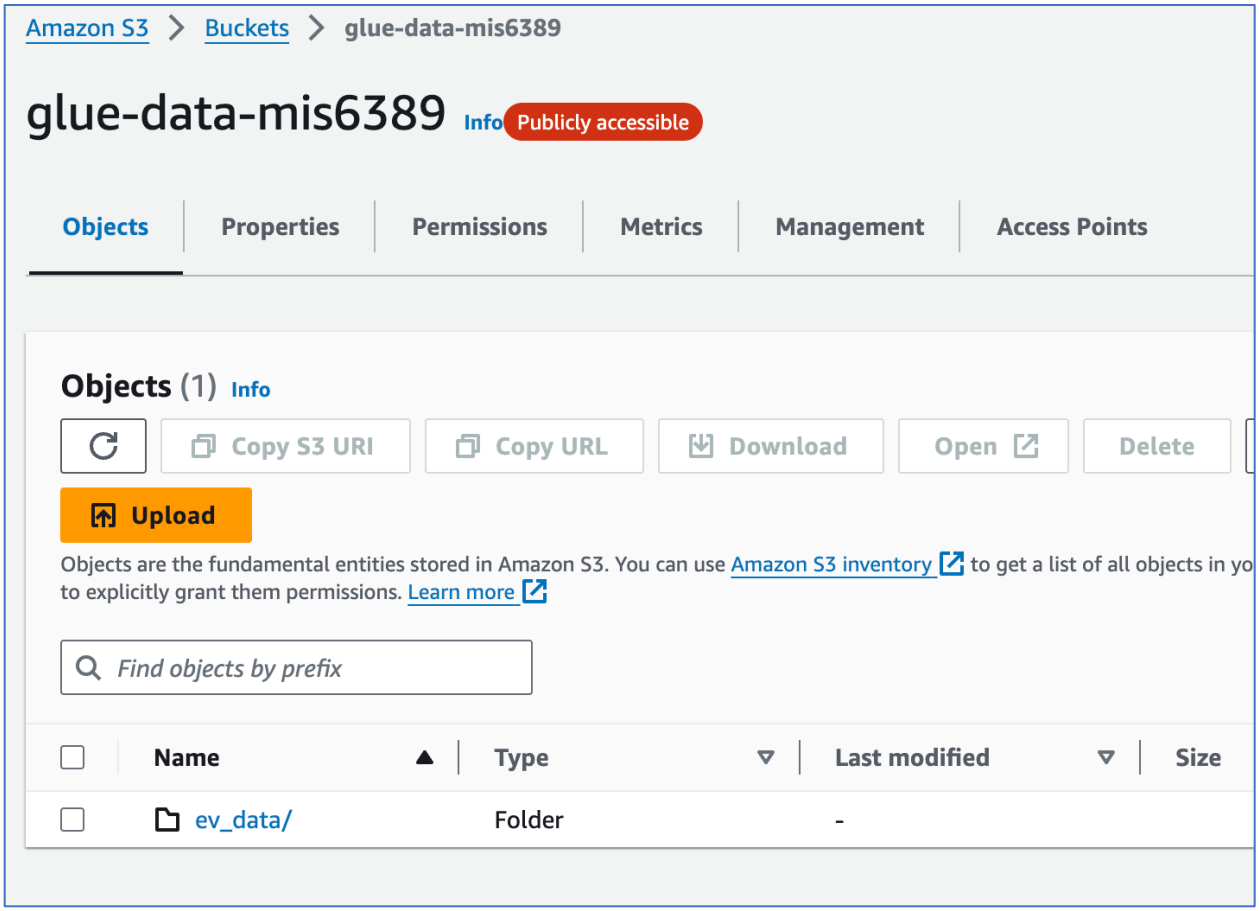
- How to define a database
- How to configure a crawler to explore data in Amazon S3 Bucket
- How to create tables
- How to query data with Amazon Athena

Let's get started…

Steps:

We are creating two buckets.

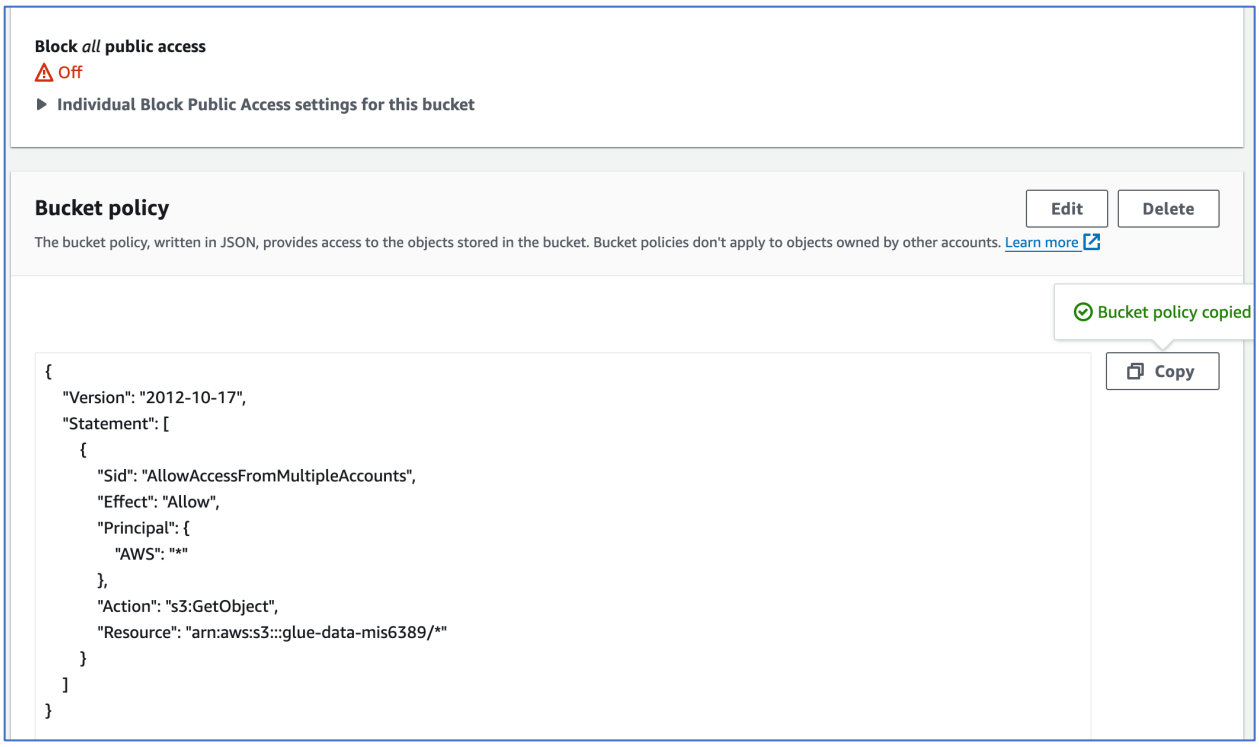
First one: Create the bucket name as given and then a folder inside it.



Upload the file in the ev_data folder: https://glue-data-mis6389.s3.amazonaws.com/ev_data/Electric_Vehicle_Population_Data.csv

Make sure the permissions tab looks as shown below. Unblock the public access and paste the policy:

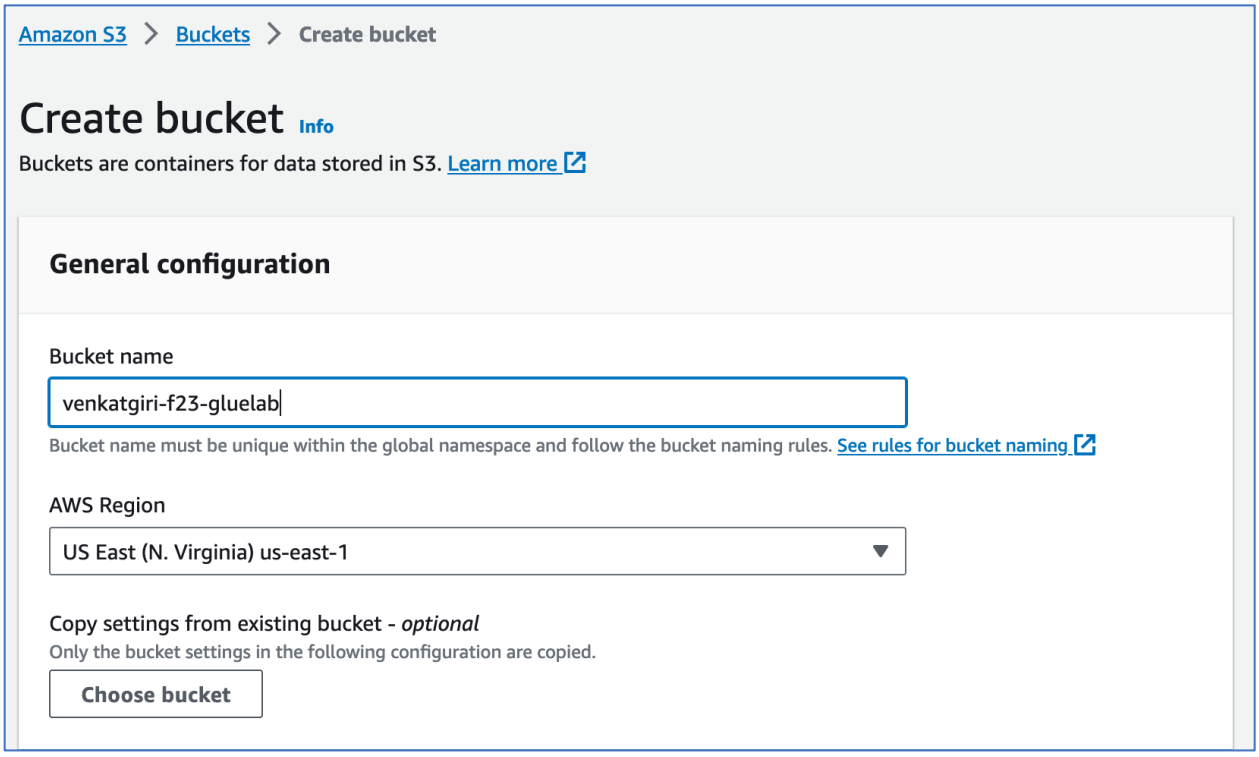
AWS Glue and Athena



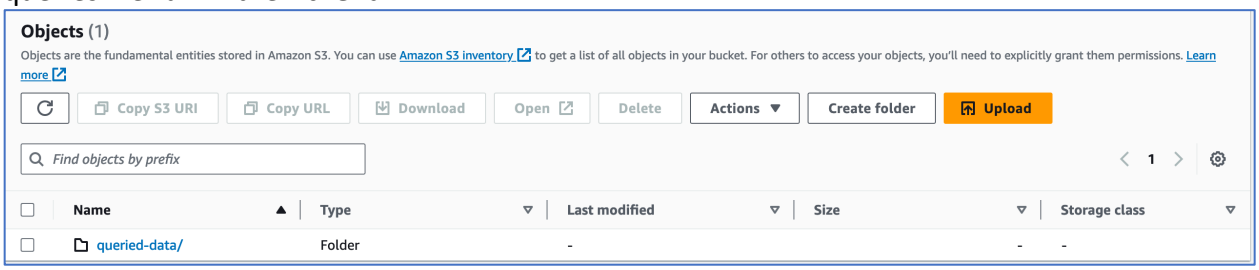

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccessFromMultipleAccounts",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::glue-data-mis6389/*"
        }
    ]
}
```

Second one:
Create an S3 bucket called **<yourname>-f23-gluelab**.

1. Give the **Bucket name** as suggested. Leave the rest of the options to default and scroll down to choose **Create bucket**.



Once the bucket is created, create a folder named **queried_data**. We store the results of all the queries we run in the **Athena.**



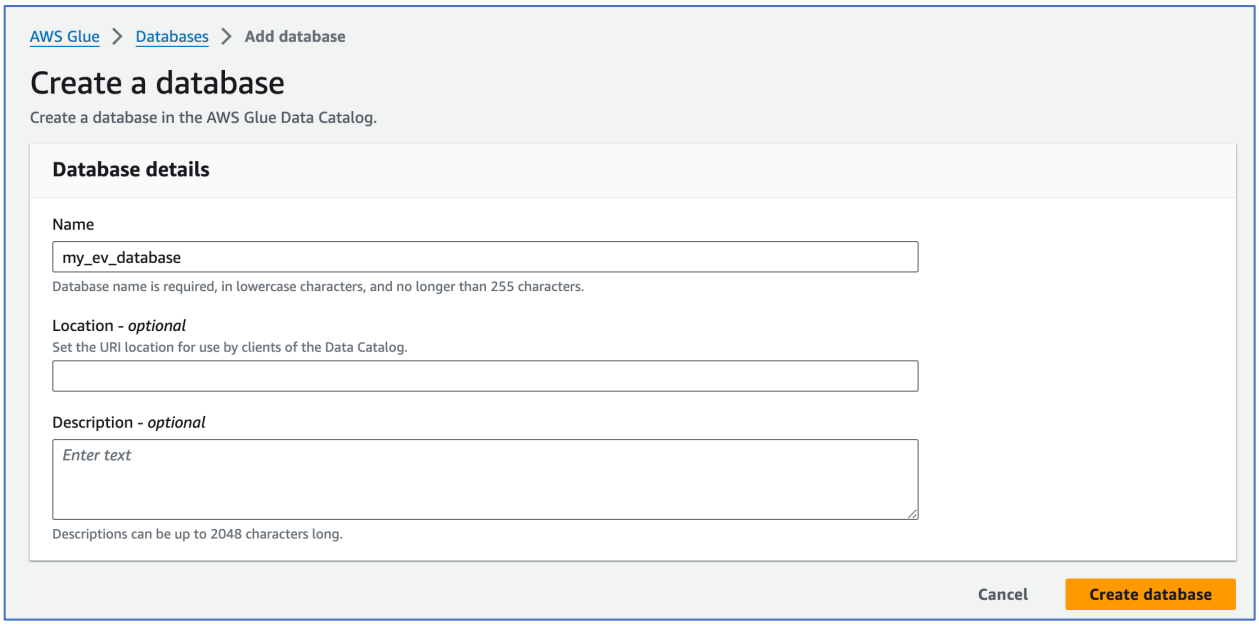
Note: This is a Deliverable

Discovering the Data

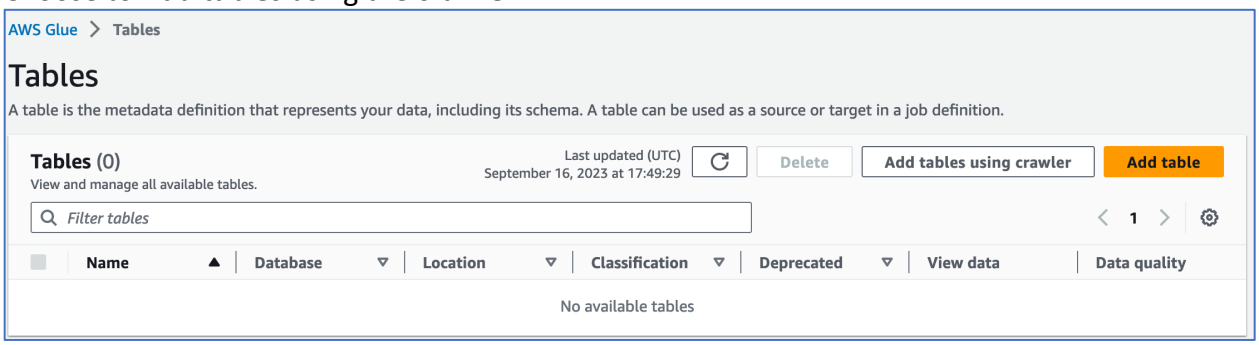
AWS Glue and Athena

In this section, we will create the Glue database, add a crawler, and populate the database table using a source CSV file.

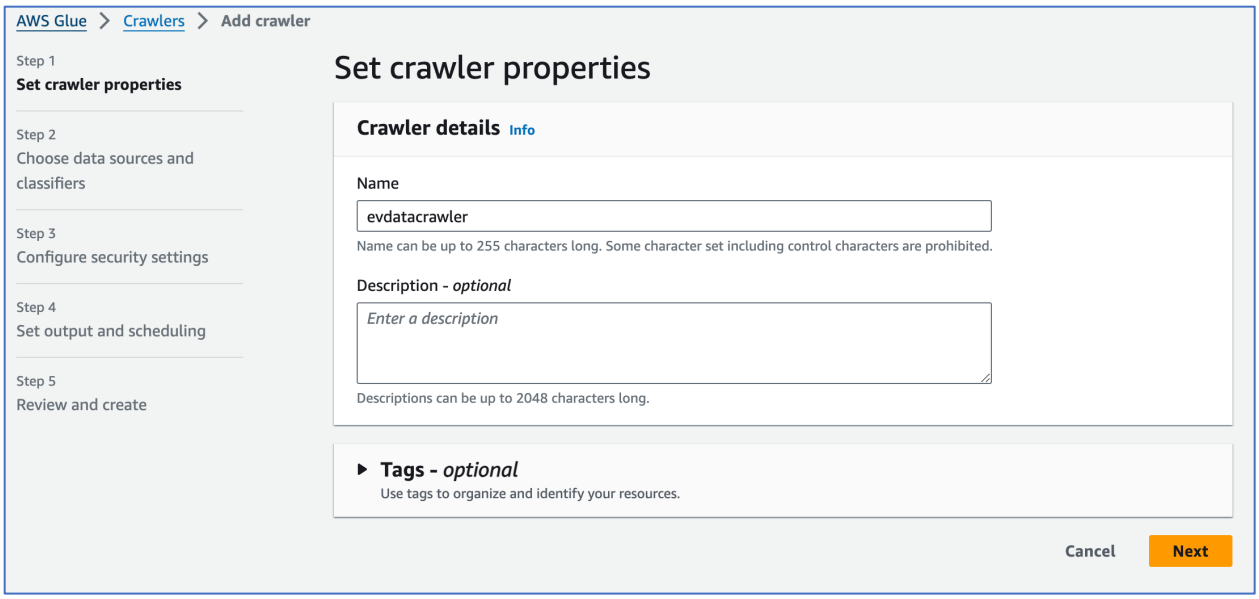
2. Choose Services and search for **AWS Glue**. Choose **Databases.** Choose **Add Database**. Paste/type in the following for the Database name: **my_ev_database**



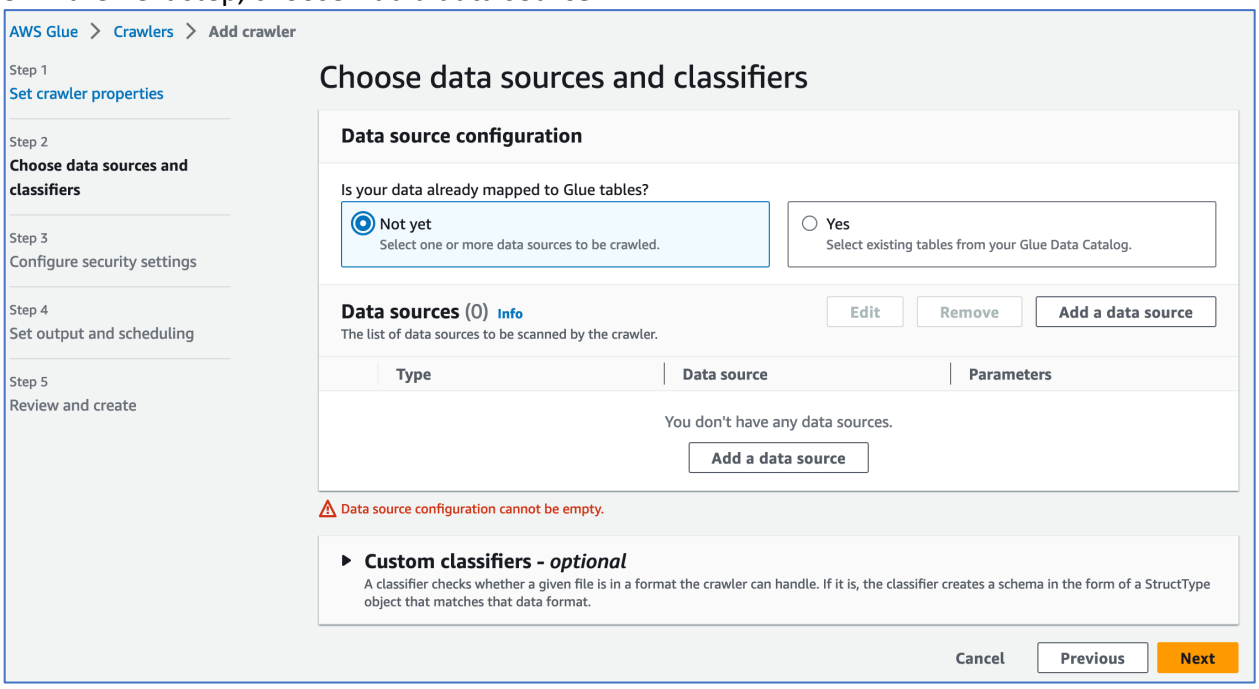
3. Choose **Tables**. You can add a table manually or by using a crawler. A crawler is a program that connects to a data store and progresses through a prioritized list of classifiers to determine the schema for your data. AWS Glue provides classifiers for common file types like CSV, JSON, Avro, and others. You can also write your classifier using a grok pattern.
Choose to **Add tables** using the **crawler**.



4. Paste in **evdatacrawler** for the Crawler name. Choose **Next**.



5. In the next step, choose **Add a data source**



6. Choose the Location of s3 data and select the options as shown below**.**
Path: s3://glue-data-mis6389/ev_data/. This S3 bucket contains the data file ev_data. Then, Choose to **Add an s3 data source.**

Add data source

Data source

Choose the source of data to be crawled.

S3

Network connection - *optional*

Optionally include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any other S3 targets will also use the same connection (or none, if left blank).

Clear selection | Add new connection

Location of S3 data

- (●) In this account
- ( ) In a different account

S3 path

Browse for or enter an existing S3 path.

s3://glue-data-mis6389/ev_data/ | View | Browse S3

All folders and files contained in the S3 path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

Subsequent crawler runs

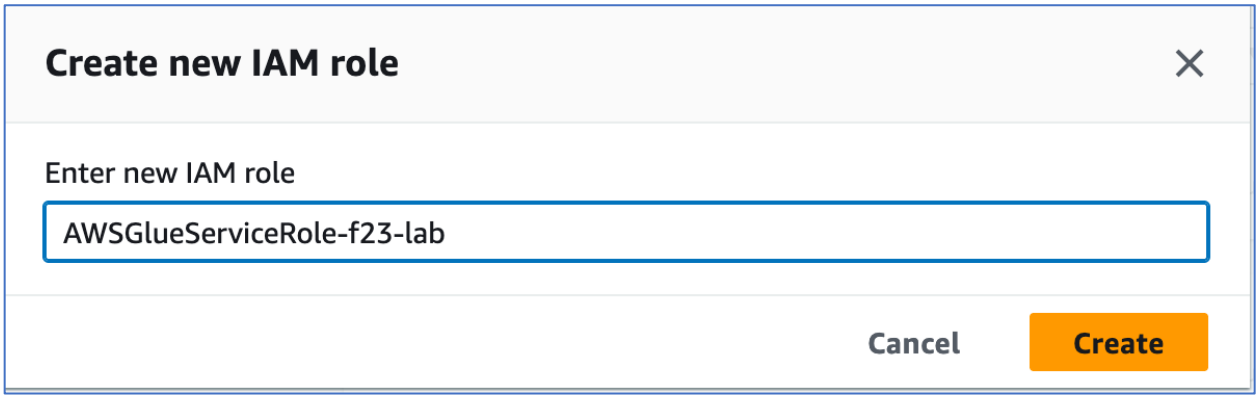
This field is a global field that affects all S3 data sources.

- (●) Crawl all sub-folders
  Crawl all folders again with every subsequent crawl.
- ( ) Crawl new sub-folders only
  Only Amazon S3 folders that were added since the last crawl will be crawled. If the schemas are compatible, new partitions will be added to existing tables.
- ( ) Crawl based on events
  Rely on Amazon S3 events to control what folders to crawl.

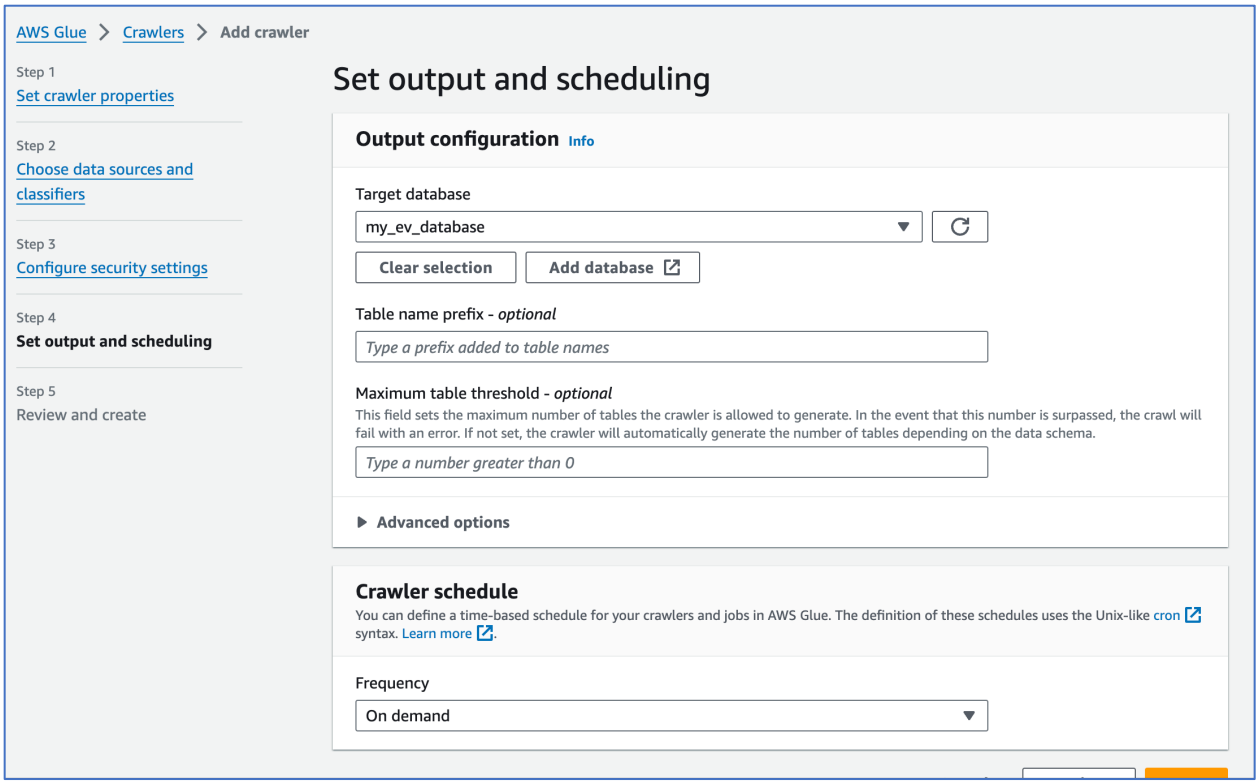
Cancel | Add an S3 data source

Once the Data sources are added, choose **Next.**

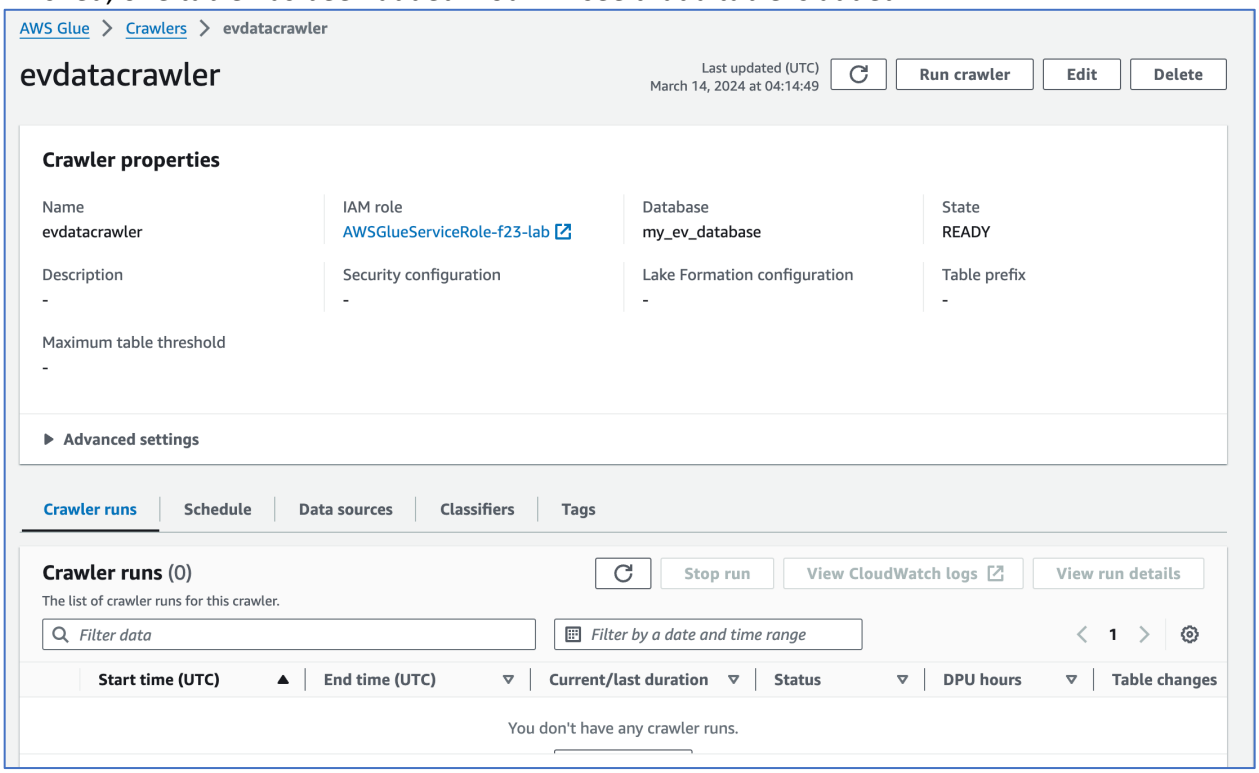
7. In the Configure security settings, choose **Create new IAM role,** and add **-f23-lab** to customize the default role name. Choose **Next.**

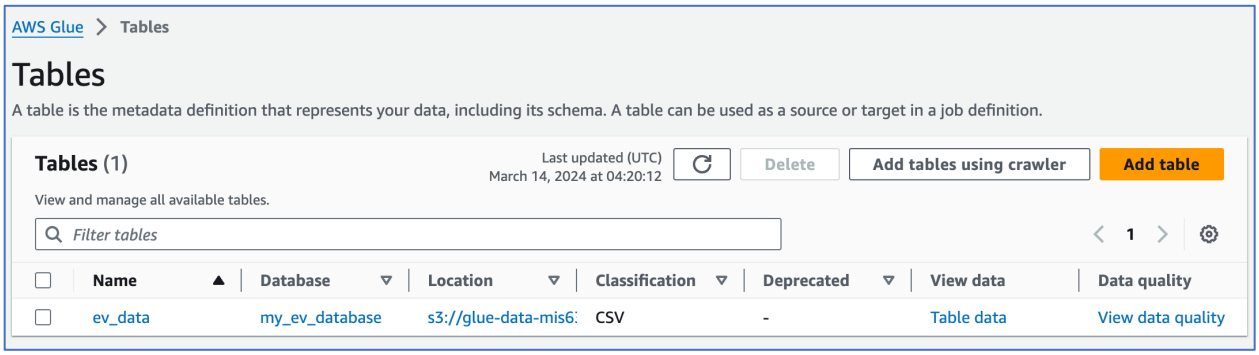


8. For the Target database, choose the **my_ev_database** database. Choose Next. Review and Choose **Create crawler**.



9. Select Crawlers and Select the **evdatacrawler** and choose **Run crawler**. When the crawler has finished, one table has been added. You will see that a table is added.

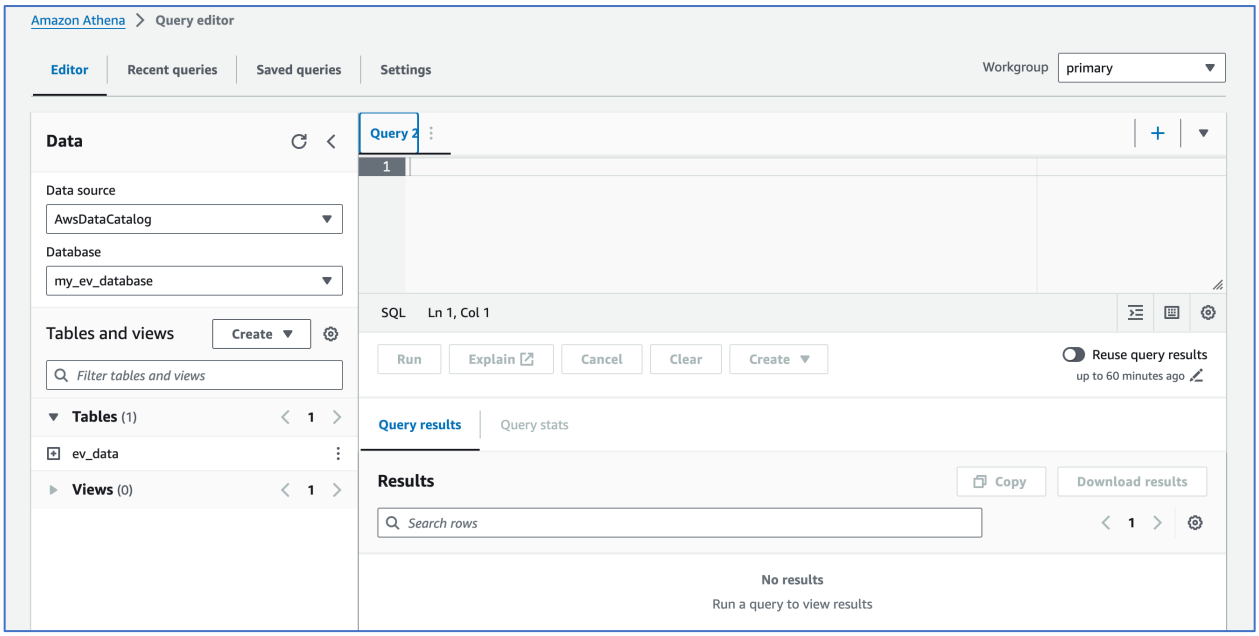




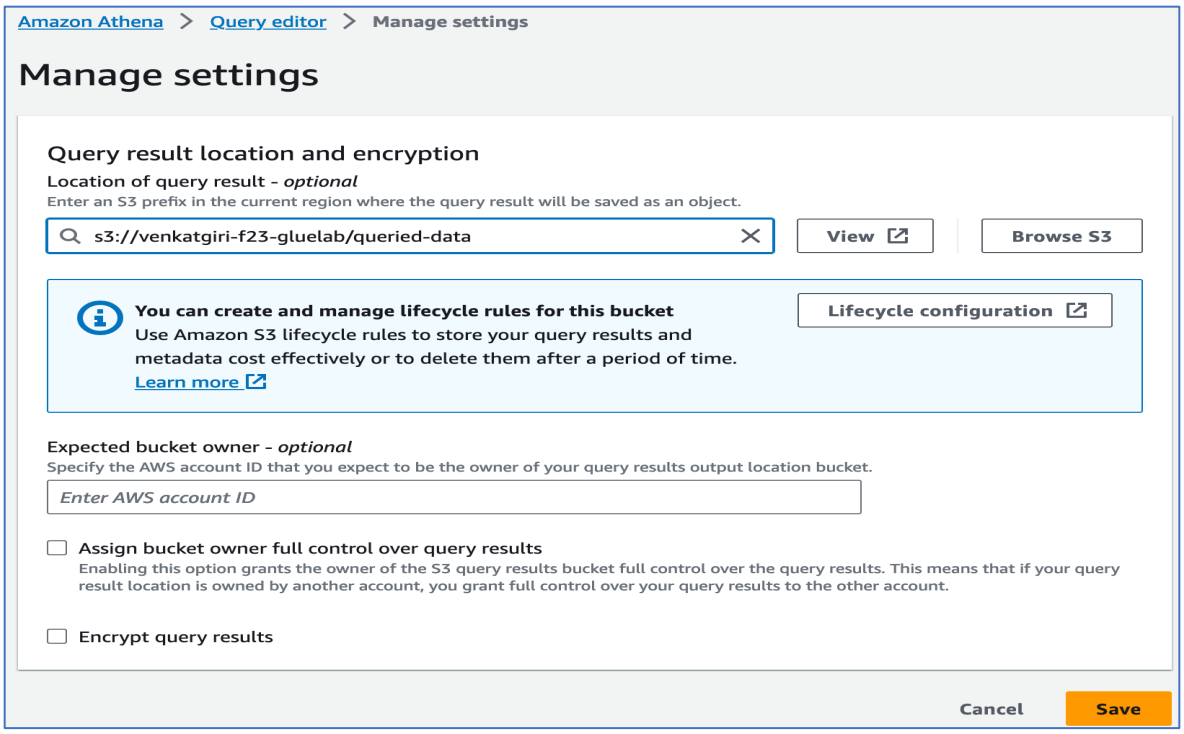
Note: This is a deliverable (Select Tables from the left menu)

Now that we have the Data catalog and table created, we can start querying using Athena

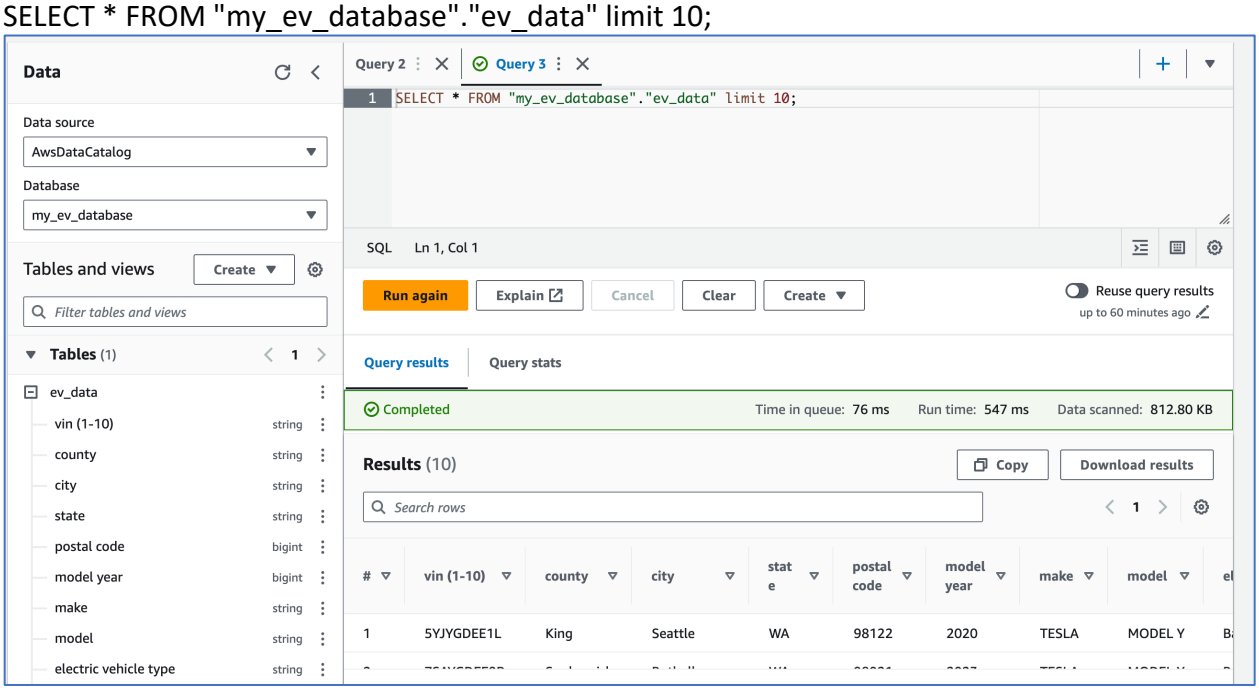
10. Choose Services and search for Athena. You may need to choose **Launch query editor**. The database will show that **my_ev_database** has been selected. Make sure your screen looks as shown below.



11. Choose the **Settings** tab and then choose **Manage.** Here, you will add the target location. Select **Browse S3** and click on the bucket you created earlier in this lab. After that you will see the **queried-data** folder. Select the folder and then click **Choose.** Click on **the Save** option**.**



12. Come back to the **Editor** tab. Now, you can start querying. Paste the below command.
Command1:
SELECT * FROM "my_ev_database"."ev_data" limit 10;



Note: This is a deliverable

Here are a few more commands for you:

- SELECT "vin (1-10)",make,"electric range" FROM "my_ev_database"."ev_data" where "electric range" > 200;
- select make from my_ev_database.ev_data where make != 'TESLA';;
- select make from my_ev_database.ev_data where "electric vehicle type" != 'Battery Electric Vehicle (BEV)';

Run the 2 SQL commands of your choice, or choose from the above commands.
(Note: This is a deliverable (screenshot of each command and respective output))

Deliverables: A total of 5 screenshots are expected from this lab.

Lab Cleanup:
Make sure to delete the crawler, table, database, and s3 bucket.