

AWS Athena querying S3 stored data

- Sri Lakshmi Prasanna Dwarampudi
(U81031815)

Introduction:

Amazon Athena is an interactive query service that facilitates data analysis using standard SQL that is situated in Amazon S3 directly. When working with huge datasets and intricate queries, Athena automatically scales by running queries in parallel, resulting in quick responses. Numerous formats are supported by it, such as Parquet, Avro, JSON, CSV, and ORC. It is the best option for quick, affordable data exploration since users can quickly construct tables, design schemas, and transform data into meaningful insights without having to change or load it into a specialized data store.

Learning Objectives:

In this project, we will learn the following:

1. Familiarizing with AWS S3 service
2. Uploading and downloading files in S3 buckets
3. Using AWS Athena for querying the data for analysis
4. Creation of separate workgroups for each data analysis in AWS Athena
5. Creating new table for S3 storage data
6. Storing queried data results back in S3 bucket for further analysis.

Pre-requisites:

1. AWS account
2. worldcities.csv (input file)
3. Basic knowledge of AWS console
4. Basic knowledge of SQL language

Requirements:

1. Create S3 bucket with default options.
2. Upload the input file 'worldcities.csv' into the created s3 bucket and copy the destination path of the uploaded file.
3. Create a S3 bucket to store queries results and store its location path.
4. Go to AWS Athena service editor, then go to settings, and change the 'Query result location' in "Query result and encryption settings" to above copied S3 bucket location path. So, all query results will be stored there.

5. Create a separate workgroup and table for the uploaded file in S3 bucket using copied S3 bucket paths.
6. Now, start querying the data for highest populated city in each country and total population of each city. Then submit both output files downloaded from S3 bucket as “Highest population cities.csv” and “Country total population and city count.csv” and

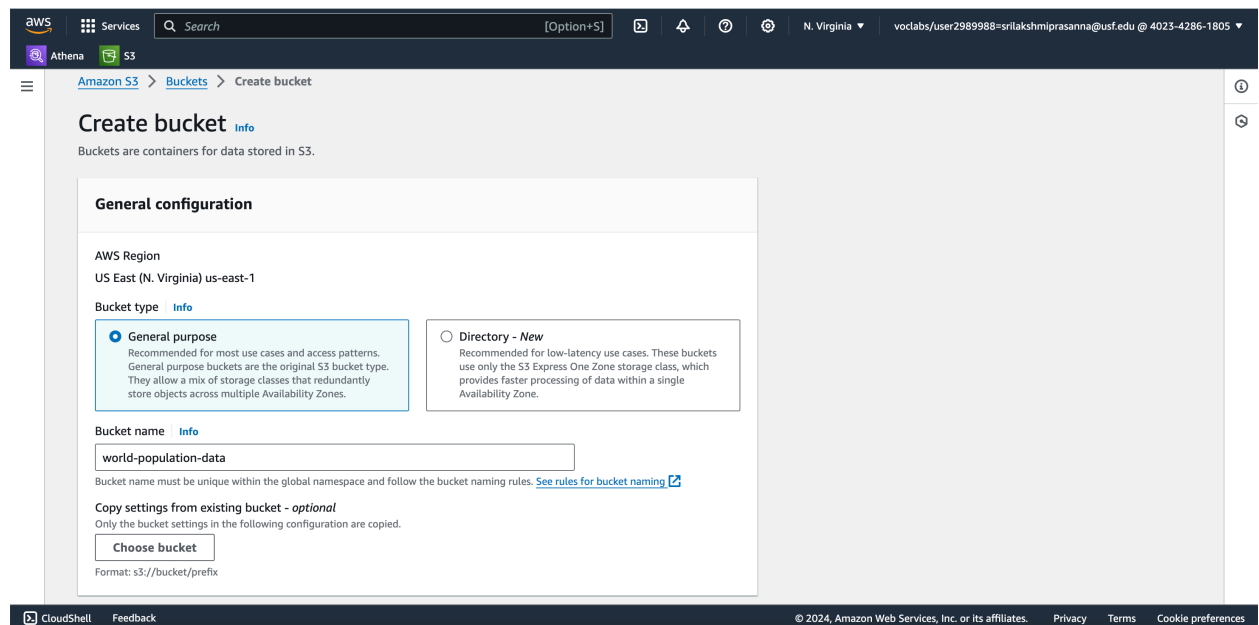
What to submit:

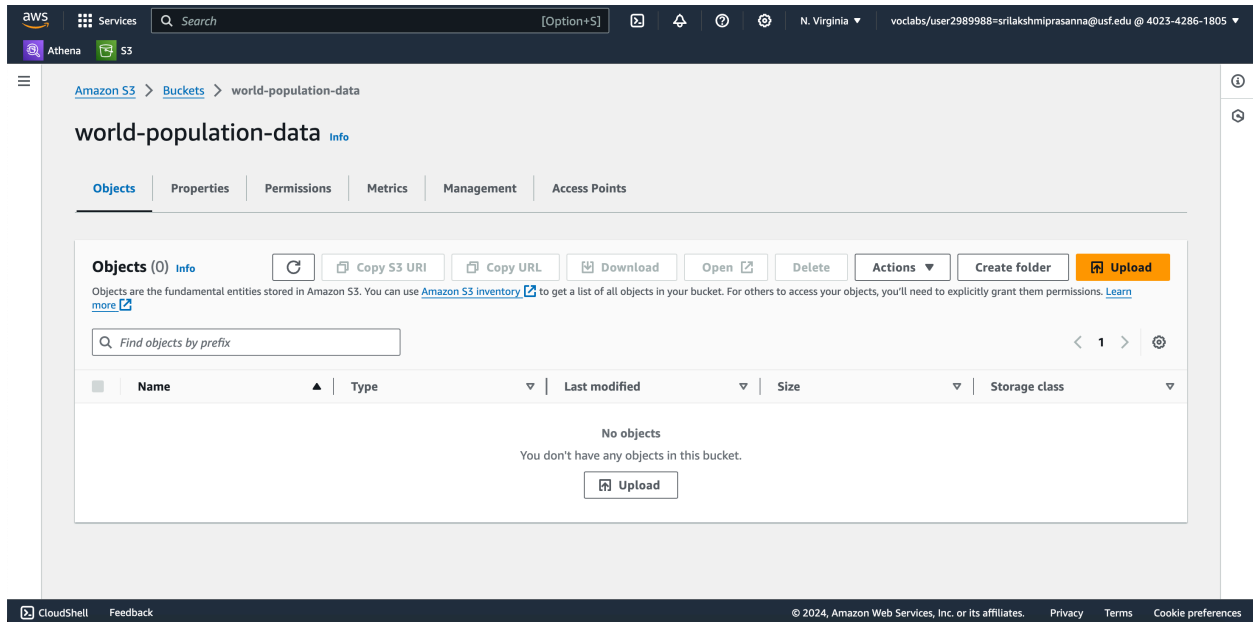
Detailed report, Zip file of all screenshots of the procedure, Output files of queries: “Country total population and city count.csv”, “Highest population cities.csv”

Implementation:

1. Creating S3 bucket for data storage:

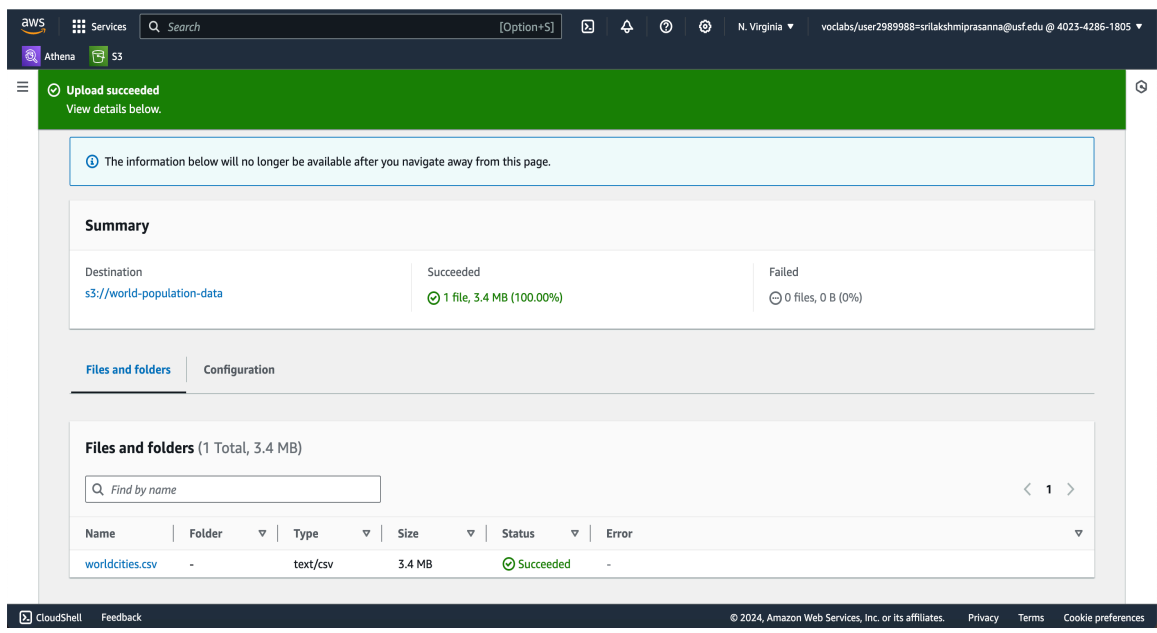
Set ‘Region’ to ‘US East’ and let the default options not changed. Then press on create bucket. S3 bucket created successfully as shown in the below screenshot.





2. Uploading the files in S3 bucket:

As shown in the below screenshot, ‘worldcities.csv’ file is uploaded in the ‘world-population-data’ s3 bucket as shown in the below screenshot. We can also see the destination path (s3://world-population-data/) of the uploaded file in the below screenshot.



3. Creating S3 bucket for query results storage:

Create another S3 bucket just for the queries results to be stored in. Then also copy the location path (s3://worldpopulation-queries/) of the ‘worldpopulation-queries’ S3 bucket.

4. Assigning the query results location:

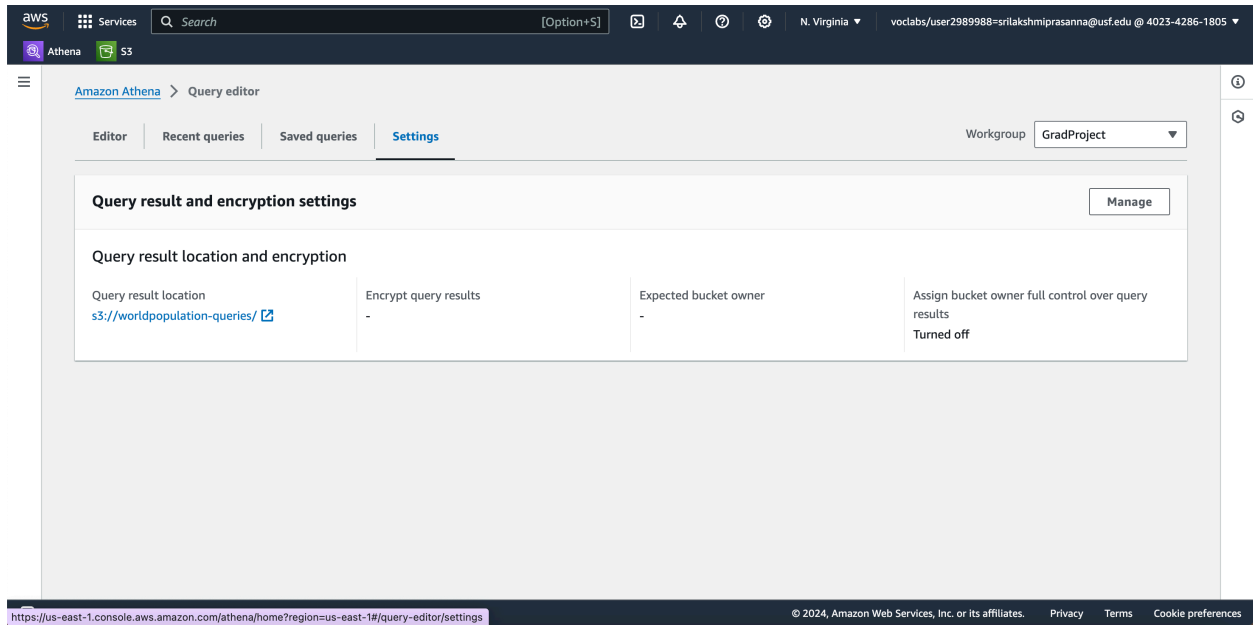
i. Creating workgroup for world-cities population data analysis:

Go to workgroups in the Amazon Athena and create the a new workgroup where we can use SQL queries to ana;yse the data without disturbing any other previous analysis as shown in the below screenshot.

The screenshot displays the 'Create workgroup' interface in the AWS Athena console. The breadcrumb navigation shows 'Amazon Athena > Workgroups > Create workgroup'. The main heading is 'Create workgroup'. Under 'Workgroup details', there is a note: 'Enter a unique name for your workgroup. To change the workgroup name, delete the workgroup and recreate it with a new name.' The 'Workgroup name' input field contains 'GradProject'. A validation message states: 'Workgroup name must be from 1-128 characters and must be unique per region of your account. Valid characters are a-z, A-Z, 0-9, _(underscore), .(period) and -(hyphen). This value cannot be changed after creation.' Below this is an optional 'Description' text area with a note: 'Workgroup description must be from 1-1024 characters. 1024 characters remaining.' The 'Analytics engine' section, marked as 'new' and 'info', asks to 'Choose the type of engine'. Two options are shown: 'Athena SQL' (selected with a radio button) and 'Apache Spark' (unselected). The footer of the console shows '© 2024 Amazon Web Services, Inc. or its affiliates.' and links for 'Privacy', 'Terms', and 'Cookie preferences'.

ii. Query Result Location:

Go to the AWS Athena query editor, then to the settings and click on the manage button and change the location of query result to our S3 bucket (s3://worldpopulation-queries/) as shown in the below screenshot. So, all our data query results done in ‘primary’ workgroup will be stored in ‘worldpopulation-queries’ S3 bucket.



5. Creating table of the 'worldcities.csv' file in AWS Athena using the following SQL query:

Used the below query and successfully created the table as shown in the below screenshot.

```
CREATE EXTERNAL TABLE IF NOT EXISTS `default`.`world-population` (
  `city` string, `city_ascii` string, `lat` double,
    `lng` double, `country` string, `iso2` string, `iso3` string, `admin_name`
string, `capital` string, `population` int, `id` int
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = ',',
  'field.delim' = ','
)
LOCATION 's3://world-population-data/'
TBLPROPERTIES ('skip.header.line.count'='1')
;
```

Typeahead suggestions are turned on by default. You can change this setting in query editor preferences.

Data

Data source: AwsDataCatalog

Database: default

Tables and views: [Create](#)

Filter tables and views

Tables (1): world-population

Views (0)

Query 3

```

1 CREATE EXTERNAL TABLE IF NOT EXISTS `default`.`world-population` (
2   `city` string,
3   `city_ascii` string,
4   `lat` double,
5   `lng` double,
6   `country` string,
7   `iso2` string,
8   `iso3` string,
9   `admin_name` string,
10  `capital` string,
11  `population` int,
12  `id` int
13 )
14 ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.LazySimpleSerDe'
15 WITH SERDEPROPERTIES (

```

SQL Ln 8, Col 17

[Run again](#) [Explain](#) [Cancel](#) [Clear](#) [Create](#)

[Query results](#) [Query stats](#)

Completed Time in queue: 76 ms Run time: 376 ms Data scanned: -

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

6. Querying data:

i. Query to get total population and cities count of the every country:

SELECT iso3, SUM(population) AS total_population, COUNT(city) AS city_count
FROM "default"."world-population" WHERE population IS NOT NULL GROUP BY
iso3 ORDER BY iso3;

Output file: “Country total population and city count.csv”

Results (223)

[Copy](#) [Download results](#)

Search rows

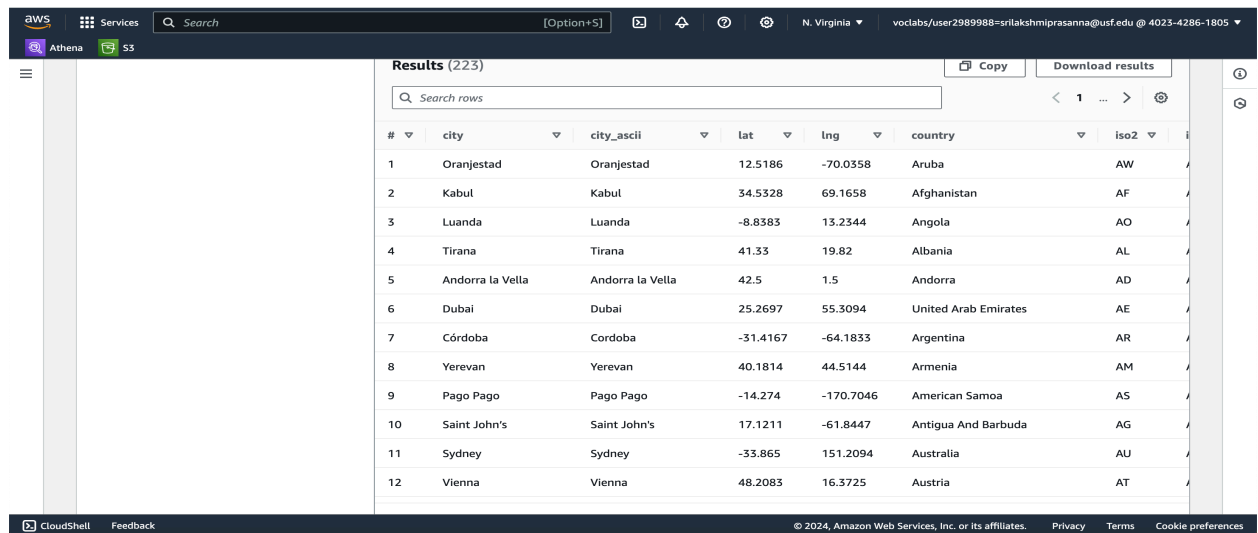
#	iso3	total_population	city_count
1	ABW	56475	2
2	AFG	8595633	40
3	AGO	22902843	93
4	ALB	1671434	51
5	AND	77354	7
6	ARE	6773563	9
7	ARG	22423029	313
8	ARM	2074010	55
9	ASM	12576	1
10	ATG	21926	1
11	AUS	21931147	359
12	AUT	5199681	237

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

ii. **Query to get the highest city data of all countries:**

```
SELECT a.*
FROM "default"."world-population" a
INNER JOIN (
    SELECT iso3, MAX(population) AS max_population
    FROM "default"."world-population"
    GROUP BY iso3
) b ON a.iso3 = b.iso3 AND a.population = b.max_population
ORDER BY a.iso3;
```

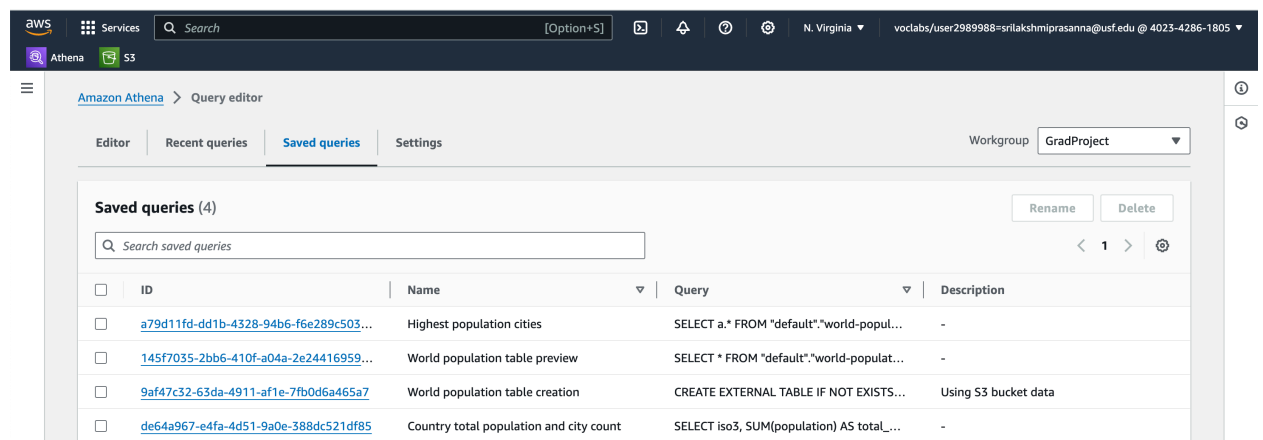
Output: “Highest population cities.csv”



The screenshot shows the AWS Athena console interface. The top navigation bar includes the AWS logo, 'Services', a search bar, and user information. The left sidebar shows 'Athena' and 'S3' services. The main content area displays 'Results (223)' with a search bar and pagination controls. Below this is a table with 12 rows and 7 columns: #, city, city_ascii, lat, lng, country, and iso2. The table lists cities like Oranjestad, Kabul, Luanda, Tirana, Andorra la Vella, Dubai, Córdoba, Yerevan, Pago Pago, Saint John's, Sydney, and Vienna.

#	city	city_ascii	lat	lng	country	iso2
1	Oranjestad	Oranjestad	12.5186	-70.0358	Aruba	AW
2	Kabul	Kabul	34.5328	69.1658	Afghanistan	AF
3	Luanda	Luanda	-8.8383	13.2344	Angola	AO
4	Tirana	Tirana	41.33	19.82	Albania	AL
5	Andorra la Vella	Andorra la Vella	42.5	1.5	Andorra	AD
6	Dubai	Dubai	25.2697	55.3094	United Arab Emirates	AE
7	Córdoba	Cordoba	-31.4167	-64.1833	Argentina	AR
8	Yerevan	Yerevan	40.1814	44.5144	Armenia	AM
9	Pago Pago	Pago Pago	-14.274	-170.7046	American Samoa	AS
10	Saint John's	Saint John's	17.1211	-61.8447	Antigua And Barbuda	AG
11	Sydney	Sydney	-33.865	151.2094	Australia	AU
12	Vienna	Vienna	48.2083	16.3725	Austria	AT

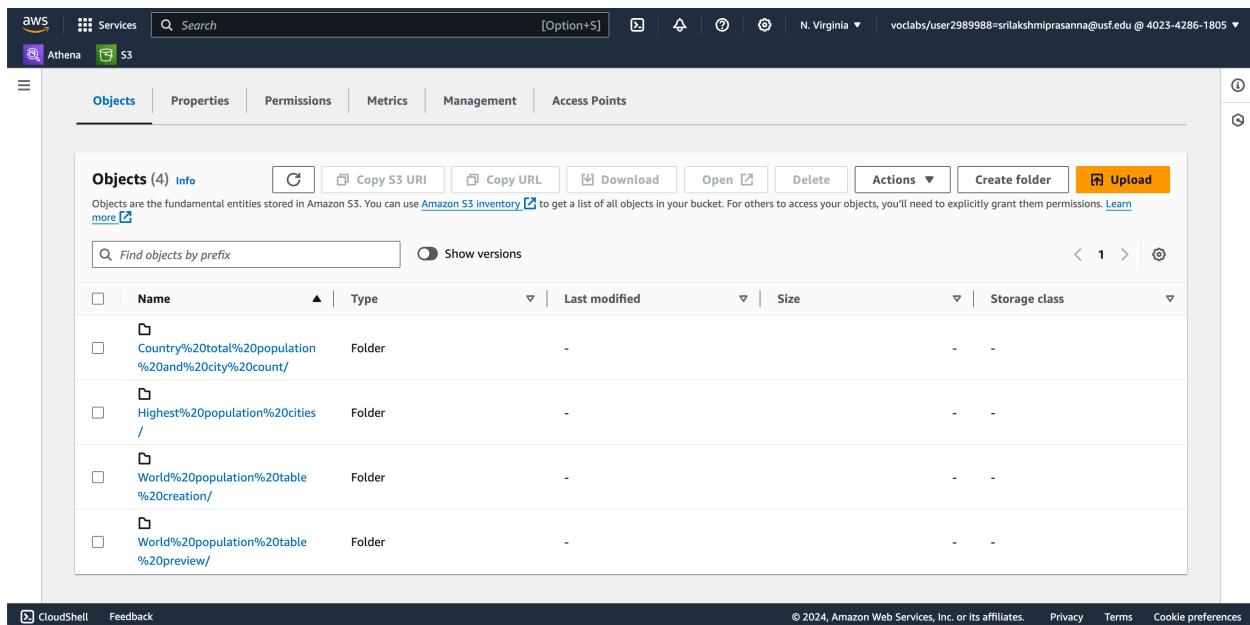
All queries can be saved by names to them as shown in the below screenshot.



The screenshot shows the 'Saved queries' tab in the AWS Athena console. It displays a list of 4 saved queries with columns for ID, Name, Query, and Description. The queries include 'Highest population cities', 'World population table preview', 'World population table creation', and 'Country total population and city count'.

ID	Name	Query	Description
a79d11fd-dd1b-4328-94b6-f6e289c503...	Highest population cities	SELECT a.* FROM "default"."world-popul...	-
145f7035-2bb6-410f-a04a-2e24416959...	World population table preview	SELECT * FROM "default"."world-populat...	-
9af47c32-63da-4911-af1e-7fb0d6a465a7	World population table creation	CREATE EXTERNAL TABLE IF NOT EXISTS...	Using S3 bucket data
de64a967-e4fa-4d51-9a0e-388dc521df85	Country total population and city count	SELECT iso3, SUM(population) AS total...	-

These results are saved in the ‘worldpopulation-queries’ S3 bucket successfully which can be downloaded or can be used for further data analysis or reporting as required as shown in the below screenshot.



Conclusion:

Through this project, the robust querying and analysis capabilities of AWS Athena for data stored in Amazon S3 buckets have been showcased. We successfully established an S3 bucket, uploaded CSV data, and used AWS Athena to run intricate SQL queries by following a step-by-step procedure. From data storage to analysis, the integration of several AWS services enabled a smooth workflow.

By using Athena, valuable insights might be extracted without the burden of maintaining conventional data warehouses. Athena's efficiency at handling massive datasets was demonstrated with queries such as finding the population total by country and the cities with the largest population. Additionally, keeping the query results in an S3 bucket made sure that our data was organized and available for any additional analysis or reporting requirements.

Overall, the integration of AWS Athena with S3 illustrates a robust, scalable, and cost-effective solution for data analytics. Organizations looking to

harness the power of big data can rely on these tools to drive decision-making and strategic planning. As cloud technologies evolve, the synergy between Athena and S3 will continue to be a cornerstone for data-driven enterprises seeking to capitalize on their data assets.