University of New Haven
TAGLIATELA COLLEGE OF ENGINEERING
Electrical & Computer Engineering and Computer Science

DATA SCIENCE

TECHNICAL REPORT

SEMESTER-2

CONTENTS

Project Name

Executive Summary

This project harnesses AWS's powerful cloud infrastructure and Python's analytical capabilities to mine, analyze, and derive insights from COVID-19 data sets. It aims to uncover patterns, trends, and risk factors associated with COVID-19 infection rates across the globe.

Team Members:
Santosh Kumar Saladi
Boddoju Venkata Naga Sai Priyanka
Ifra Naaz Mohammed
Sai Gowtham Reddy Gurrala

Questions?
Contact :

Technical Report

Project Title-

Streamlining COVID-19 Data Analysis with AWS

Highlights of Project

The project addresses the challenges posed by the COVID-19 pandemic in consolidating and analyzing data from diverse sources, which has impeded effective decision-making. To overcome these challenges, a comprehensive data pipeline leveraging cloud-based technologies is proposed to aggregate and transform COVID-19 data for efficient analysis and decision support.

Key objectives of the project include establishing a robust data pipeline, ensuring data integrity, developing a centralized data warehouse, deriving meaningful insights, and providing a user-friendly platform for data access. By harnessing cloud services such as Amazon Web Services (AWS), the solution streamlines data integration and analysis processes related to COVID-19.

The solution overview emphasizes the implementation of a scalable, cloud-native platform powered by AWS services including Amazon S3, AWS Glue, Amazon Redshift, and Amazon Athena. This platform facilitates efficient data processing and analysis, enabling stakeholders to derive actionable insights through interactive dashboards and reporting tools.
Challenges encountered include addressing data quality and reliability, ensuring scalability and performance, and managing complex data transformations and integrations. These challenges were mitigated through strategic planning, leveraging online resources, AWS documentation, and collaboration with domain experts.

In conclusion, the project offers an efficient solution for extracting, transforming, and analyzing COVID-19 datasets, leading to valuable insights crucial for informed decision-making during the pandemic. By automating data processes and providing visualization tools, the solution empowers stakeholders with timely and accurate information on COVID-19 trends. This supports proactive measures to protect public health, enabling risk assessment and adoption of preventive measures based on a deeper understanding of the pandemic's impact. Ultimately, the project contributes to the global efforts in combating and managing the COVID-19 crisis through effective data utilization and decision support.

Submitted on:

04-22-2024

Abstract

This project focuses on building a robust data pipeline using cloud-based technologies to consolidate, transform, and analyze COVID-19 data effectively. Leveraging Amazon Web Services (AWS) infrastructure, including Amazon S3, AWS Glue, Amazon Redshift, and Amazon Athena, the solution enables seamless integration of diverse data sources and scalable analytics. By automating data processing and leveraging cloud-native services, stakeholders can derive actionable insights crucial for informed decision-making in response to the pandemic. The data pipeline facilitates efficient extraction, transformation, and loading (ETL) processes, ensuring data integrity and reliability throughout the analysis. Through the utilization of AWS services, such as Amazon Redshift for large-scale analytics and Amazon Athena for ad-hoc querying, the project enables comprehensive data exploration and visualization. This cloud-based approach empowers stakeholders to monitor COVID-19 trends, identify patterns, and derive meaningful conclusions from the data. Ultimately, the project contributes to global efforts by providing a scalable, secure, and intuitive platform for managing and analyzing COVID-19 data, supporting proactive measures to mitigate the impact of the pandemic on public health and well-being.

Methodology

The CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology is a widely used framework for guiding data mining and analytics projects. It provides a structured approach to managing the lifecycle of data-driven projects, including those focused on data analysis and modeling. CRISP-DM consists of six phases, each representing a distinct stage in the project lifecycle:

Business Understanding:

In this initial phase, stakeholders and data scientists collaborate to understand the project objectives, requirements, and business goals. Key questions are addressed, such as: What are the business objectives? What problems are we trying to solve? What value will the project deliver?

Data Understanding:

This phase involves data collection, exploration, and initial assessment. Data sources are identified, and relevant datasets are gathered for analysis. Data quality is evaluated, and initial insights are derived to understand the nature and characteristics of the data.

Data Preparation:

In this phase, data preprocessing tasks are performed to clean, transform, and prepare the data for modeling. Steps may include handling missing values, feature engineering, scaling, and formatting the data in a suitable structure for analysis.

Modeling:

The modeling phase focuses on selecting and applying appropriate machine learning algorithms or analytical techniques to build predictive or descriptive models. Multiple models may be developed and evaluated to identify the most effective approach for addressing the project objectives.

Evaluation:

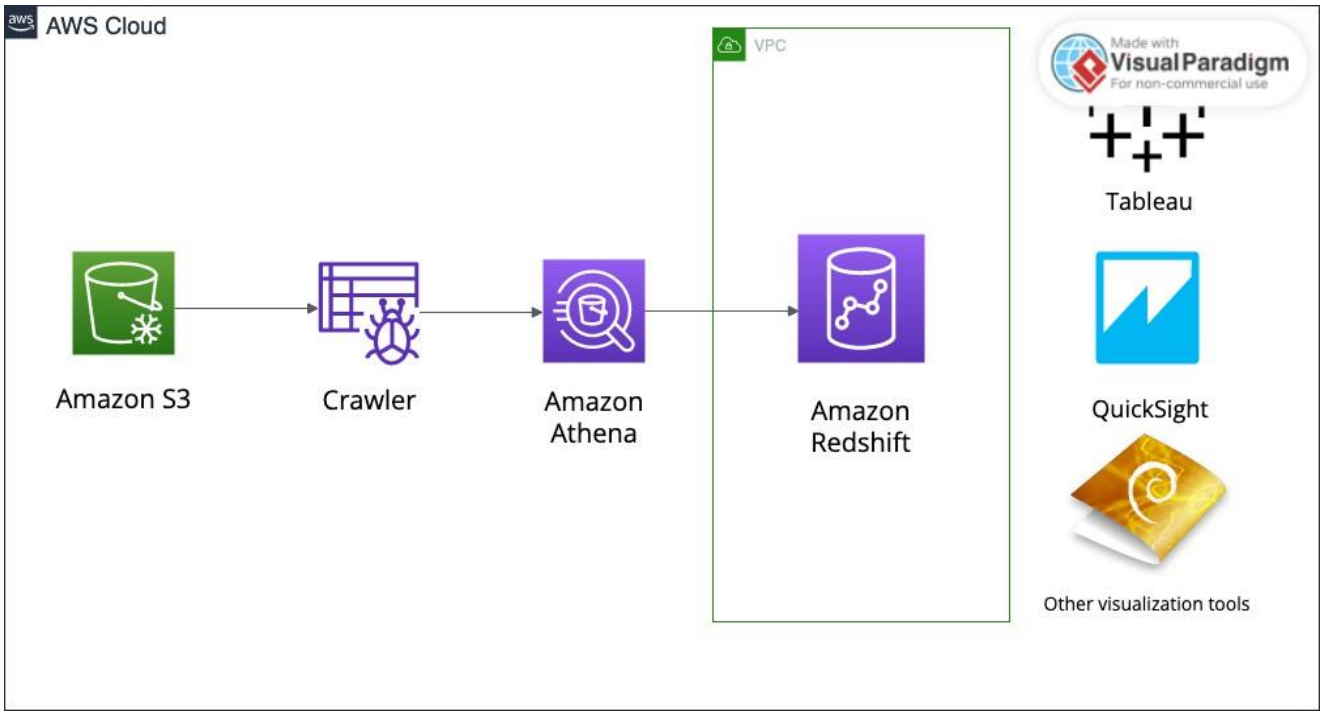
Models developed in the previous phase are evaluated against predefined performance metrics and criteria. This involves assessing model accuracy, reliability, and generalizability using validation techniques such as cross-validation or holdout sampling.

Deployment:

The final phase involves deploying the selected model(s) into operational use. This may involve integrating the model into business processes or applications to generate actionable insights and support decision-making. Monitoring and maintenance procedures are established to ensure the continued performance and relevance of deployed models.

Results Section

Data Engineering Pipeline:



Data Ingestion: Boto3

Data Storage: AWS S3 Bucket

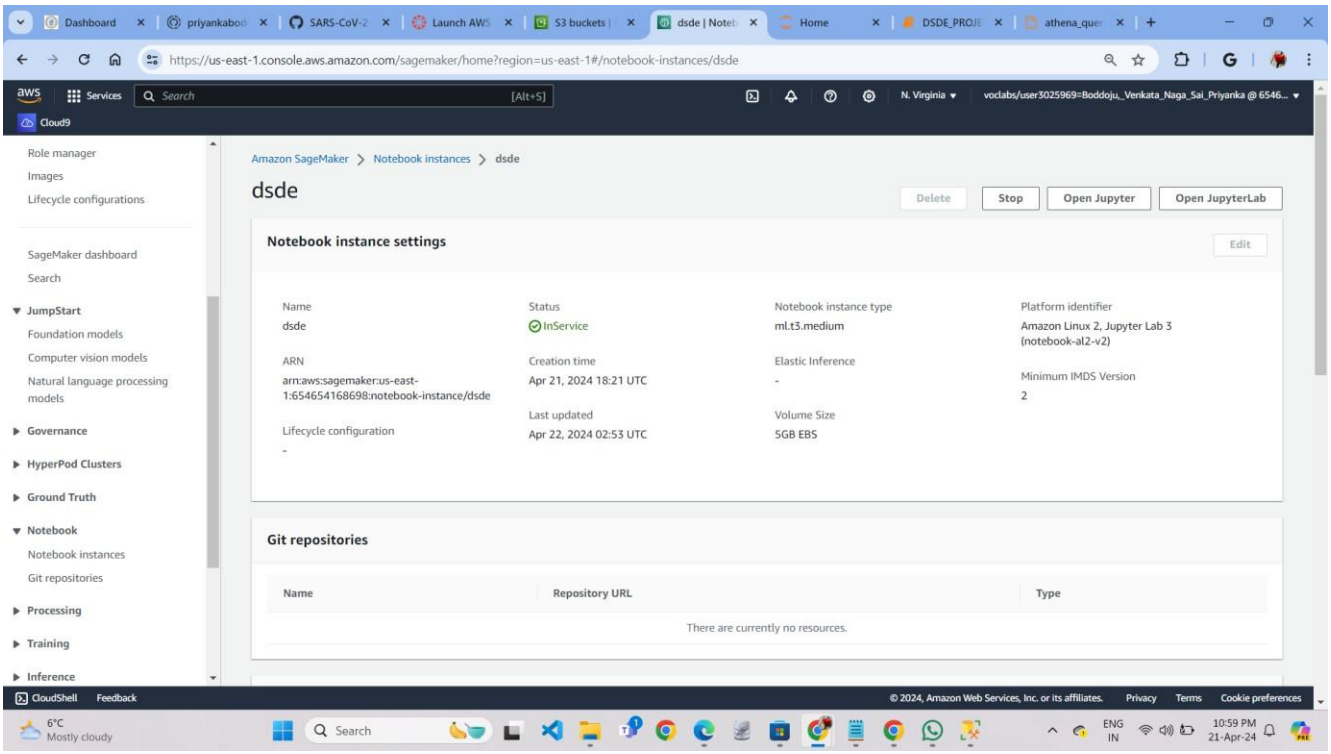
Data Processing: AWS Sagemaker

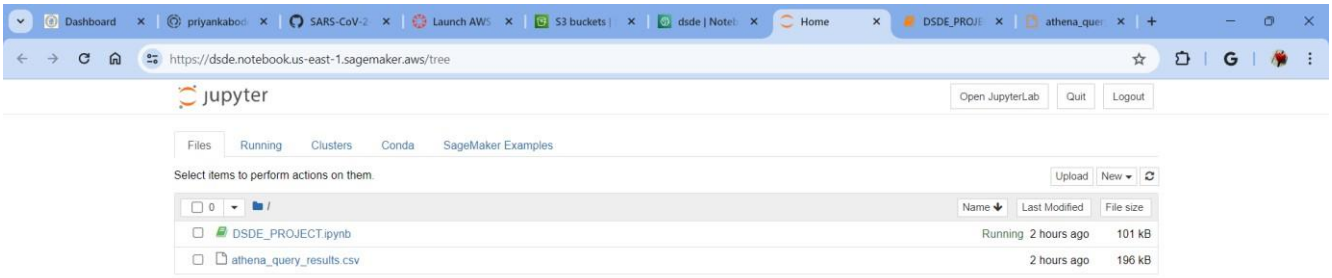
Data Consumption: AWS Athena

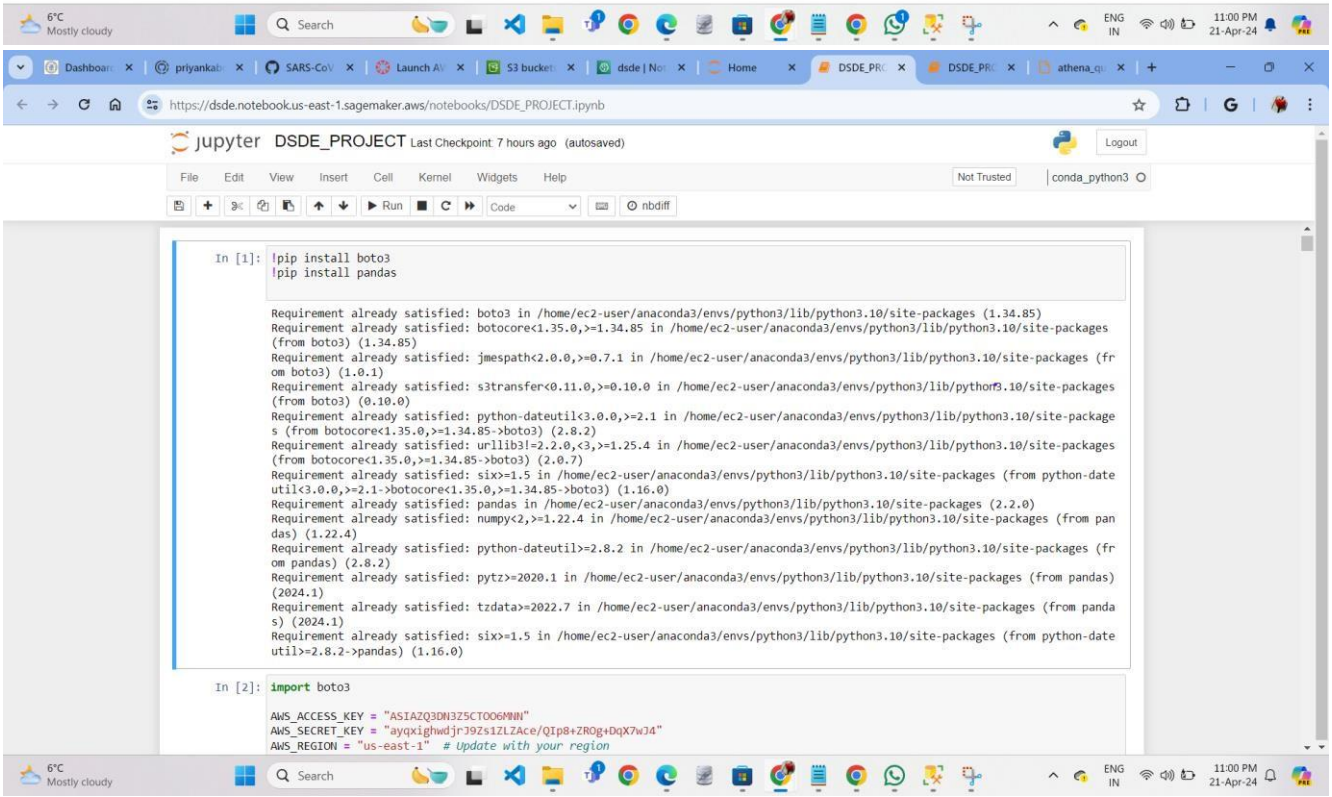
Discussion

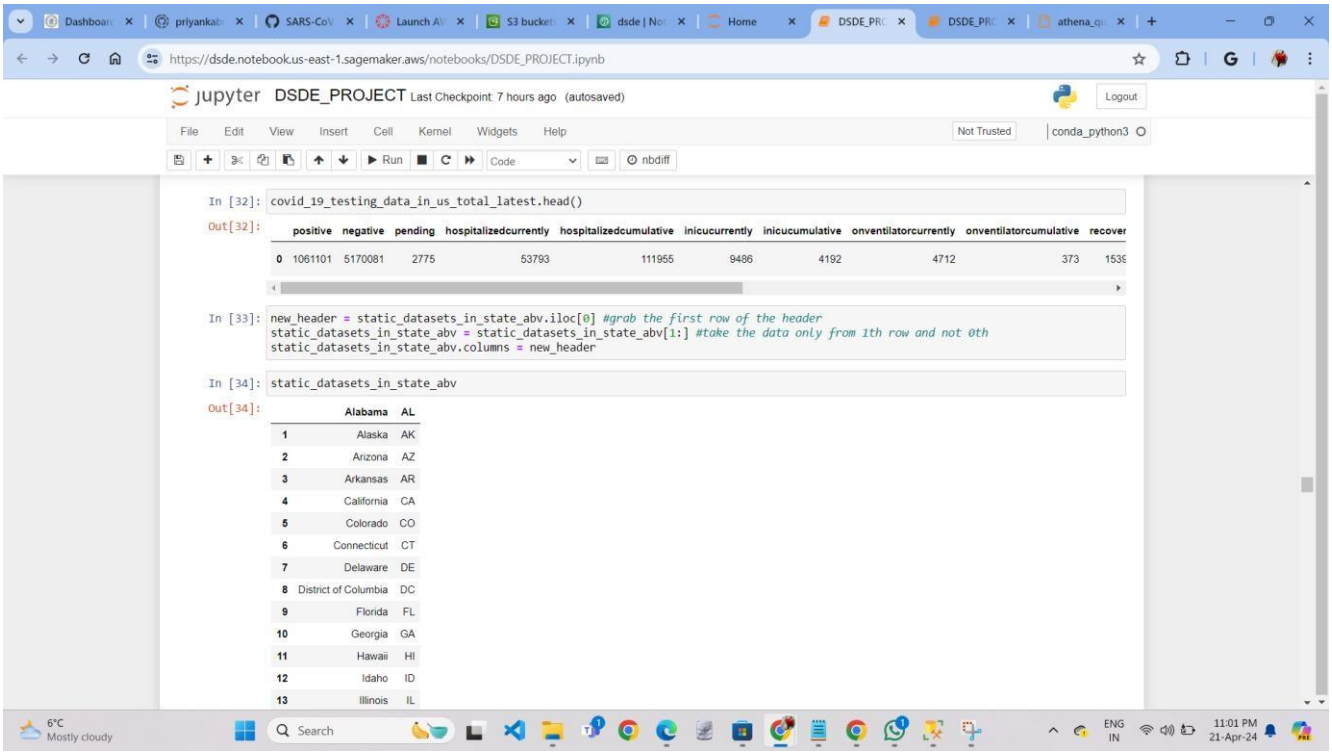
Using AWS Sagemaker

Amazon SageMaker provides a streamlined platform for end-to-end machine learning, covering data preparation, model development, deployment, and monitoring. It offers built-in algorithms, managed training, and hyperparameter tuning to optimize model performance. SageMaker simplifies deployment with real-time endpoints and batch processing, while its pay-as-you-go pricing and AWS integration ensure cost-effectiveness and scalability.




https://dsde.notebook.us-east-1.sagemaker.aws/tree
jupyter
Open JupyterLab
Quit
Logout
Files
Running
Clusters
Conda
SageMaker Examples
Select items to perform actions on them.
Upload
New
Name
Last Modified
File size
DSDE_PROJECT.ipynb
Running 2 hours ago
101 kB
athena_query_results.csv
2 hours ago
196 kB


https://dsde.notebook.us-east-1.sagemaker.aws/notebooks/DSDE_PROJECT.ipynb
jupyter DSDE_PROJECT Last Checkpoint: 7 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
conda_python3
Run
Code
nbdiff
In [1]:
!pip install boto3
!pip install pandas
Requirement already satisfied: boto3 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (1.34.85)
Requirement already satisfied: botocore<1.35.0,>=1.34.85 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from boto3) (1.34.85)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from boto3) (1.0.1)
Requirement already satisfied: s3transfer<0.11.0,>=0.10.0 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from boto3) (0.10.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from botocore<1.35.0,>=1.34.85->boto3) (2.8.2)
Requirement already satisfied: urllib3!=2.2.0,<3,>=1.25.4 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from botocore<1.35.0,>=1.34.85->boto3) (2.0.7)
Requirement already satisfied: six>=1.5 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.35.0,>=1.34.85->boto3) (1.16.0)
Requirement already satisfied: pandas in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (2.2.0)
Requirement already satisfied: numpy<2,>=1.22.4 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from pandas) (1.22.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from pandas) (2024.1)
Requirement already satisfied: six>=1.5 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
In [2]:
import boto3
AWS_ACCESS_KEY = "ASIAZQ3DN3Z5CTOO6MNN"
AWS_SECRET_KEY = "ayqxighwdjrJ9Zs1ZLZAce/QIp8+ZROg+DqX7wJ4"
AWS_REGION = "us-east-1" # Update with your region



jupyter DSDE_PROJECT Last Checkpoint: 7 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | conda_python3

Run Code nbdiff

In [32]:
```
covid_19_testing_data_in_us_total_latest.head()
```

Out[32]:

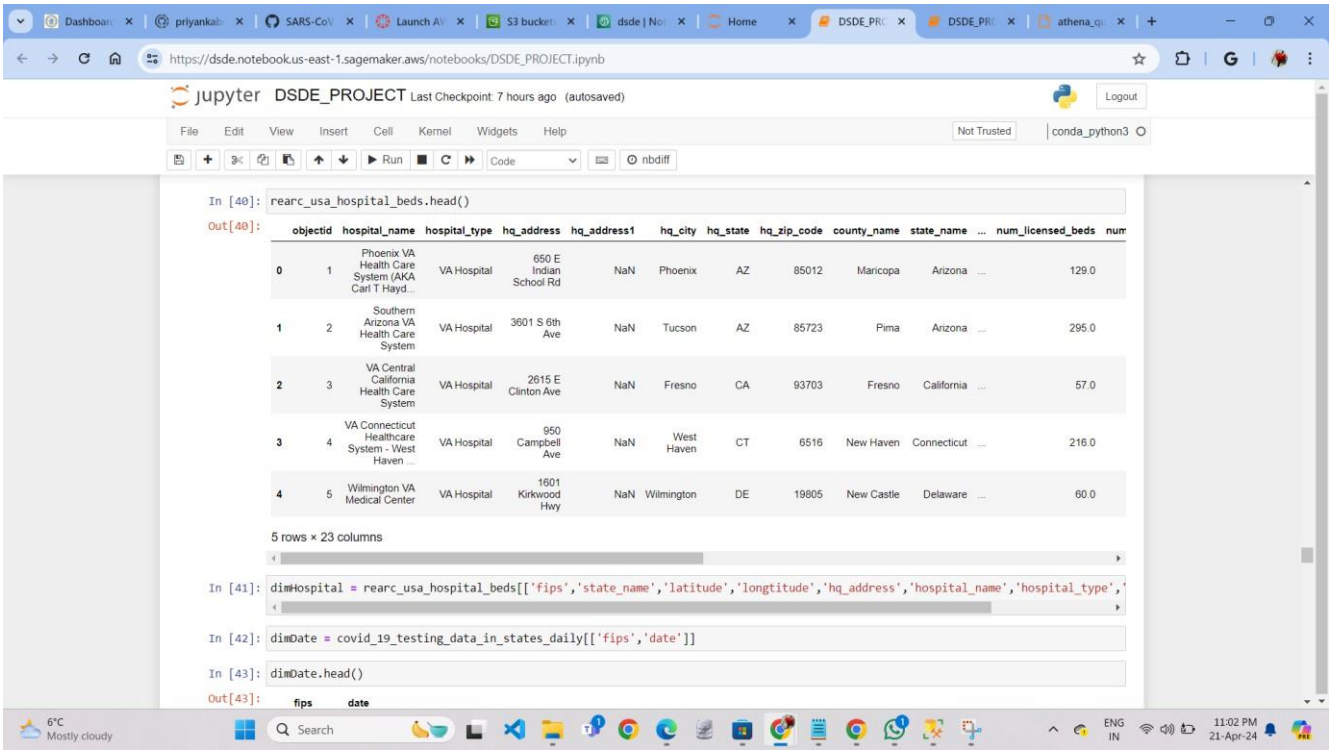
	positive	negative	pending	hospitalizedcurrently	hospitalizedcumulative	inicucurrently	inicucumulative	onventilatorcurrently	onventilatorcumulative	recover
0	1061101	5170081	2775	53793	111955	9486	4192	4712	373	153

In [33]:
```
new_header = static_datasets_in_state_abv.iloc[0] #grab the first row of the header
static_datasets_in_state_abv = static_datasets_in_state_abv[1:] #take the data only from 1th row and not 0th
static_datasets_in_state_abv.columns = new_header
```

In [34]:
```
static_datasets_in_state_abv
```

Out[34]:

	Alabama	AL
1	Alaska	AK
2	Arizona	AZ
3	Arkansas	AR
4	California	CA
5	Colorado	CO
6	Connecticut	CT
7	Delaware	DE
8	District of Columbia	DC
9	Florida	FL
10	Georgia	GA
11	Hawaii	HI
12	Idaho	ID
13	Illinois	IL



jupyter DSDE_PROJECT Last Checkpoint: 7 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | conda_python3

Run Code nbdiff

In [40]:
```
rearc_usa_hospital_beds.head()
```

Out[40]:

	objectid	hospital_name	hospital_type	hq_address	hq_address1	hq_city	hq_state	hq_zip_code	county_name	state_name	...	num_licensed_beds	num
0	1	Phoenix VA Health Care System (AKA Carl T Hayd...	VA Hospital	650 E Indian School Rd	NaN	Phoenix	AZ	85012	Maricopa	Arizona	...	129.0	
1	2	Southern Arizona VA Health Care System	VA Hospital	3601 S 6th Ave	NaN	Tucson	AZ	85723	Pima	Arizona	...	295.0	
2	3	VA Central California Health Care System	VA Hospital	2615 E Clinton Ave	NaN	Fresno	CA	93703	Fresno	California	...	57.0	
3	4	VA Connecticut Healthcare System - West Haven ...	VA Hospital	950 Campbell Ave	NaN	West Haven	CT	6516	New Haven	Connecticut	...	216.0	
4	5	Wilmington VA Medical Center	VA Hospital	1601 Kirkwood Hwy	NaN	Wilmington	DE	19805	New Castle	Delaware	...	60.0	

5 rows × 23 columns

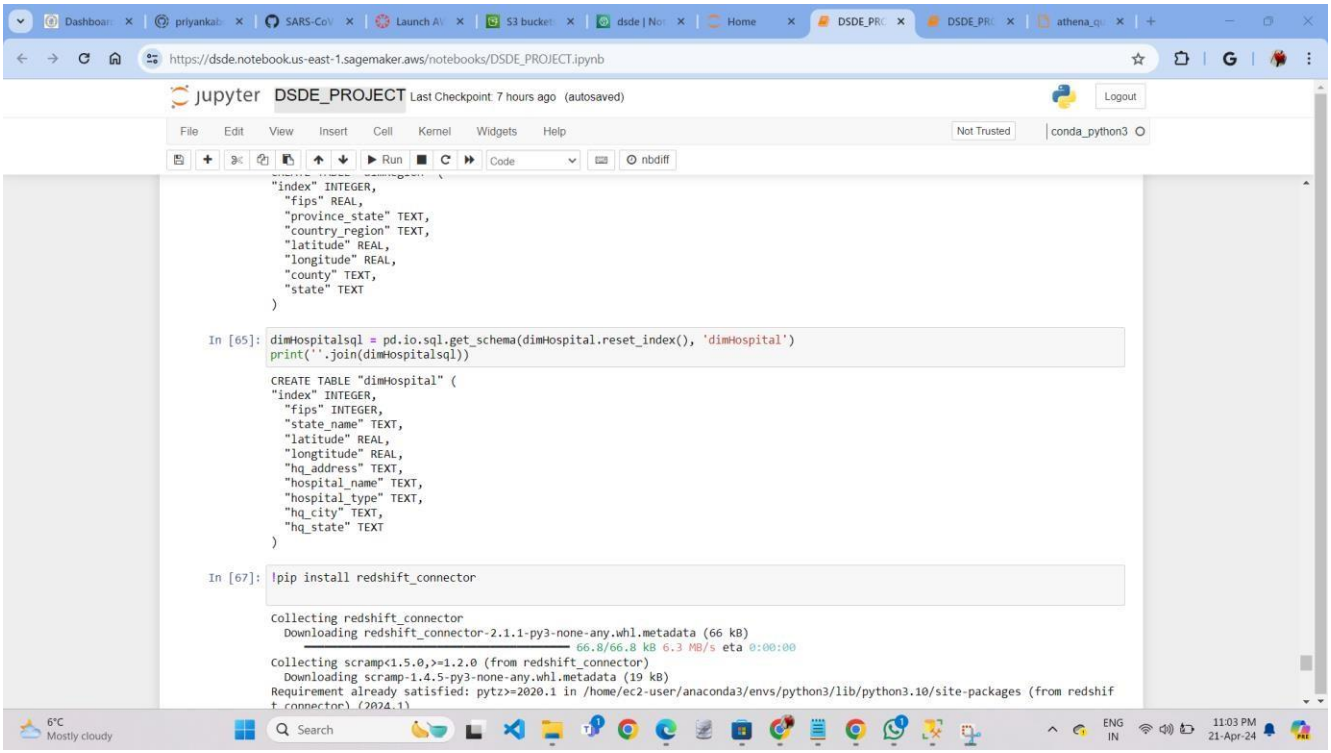
In [41]:
```
dimHospital = rearc_usa_hospital_beds[['fips','state_name','latitude','longtitude','hq_address','hospital_name','hospital_type',
```

In [42]:
```
dimDate = covid_19_testing_data_in_states_daily[['fips','date']]
```

In [43]:
```
dimDate.head()
```

Out[43]:

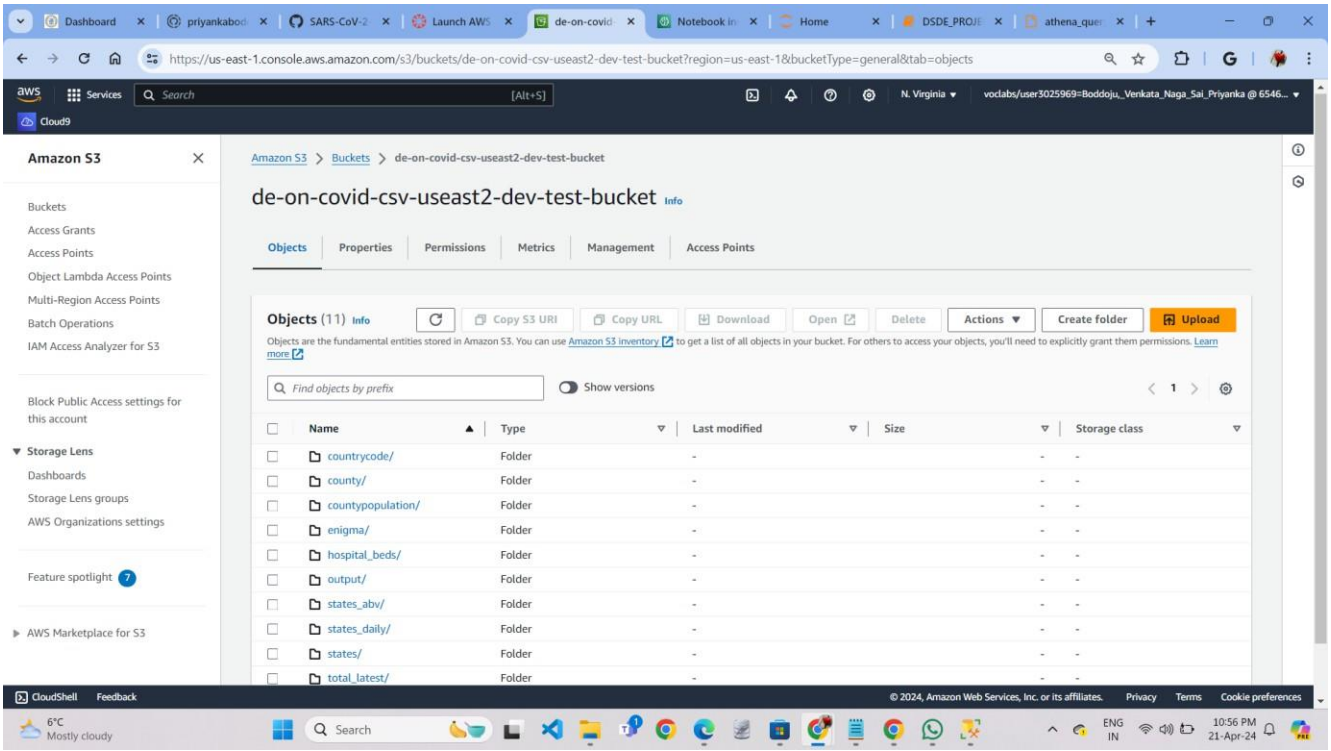
	fips	date

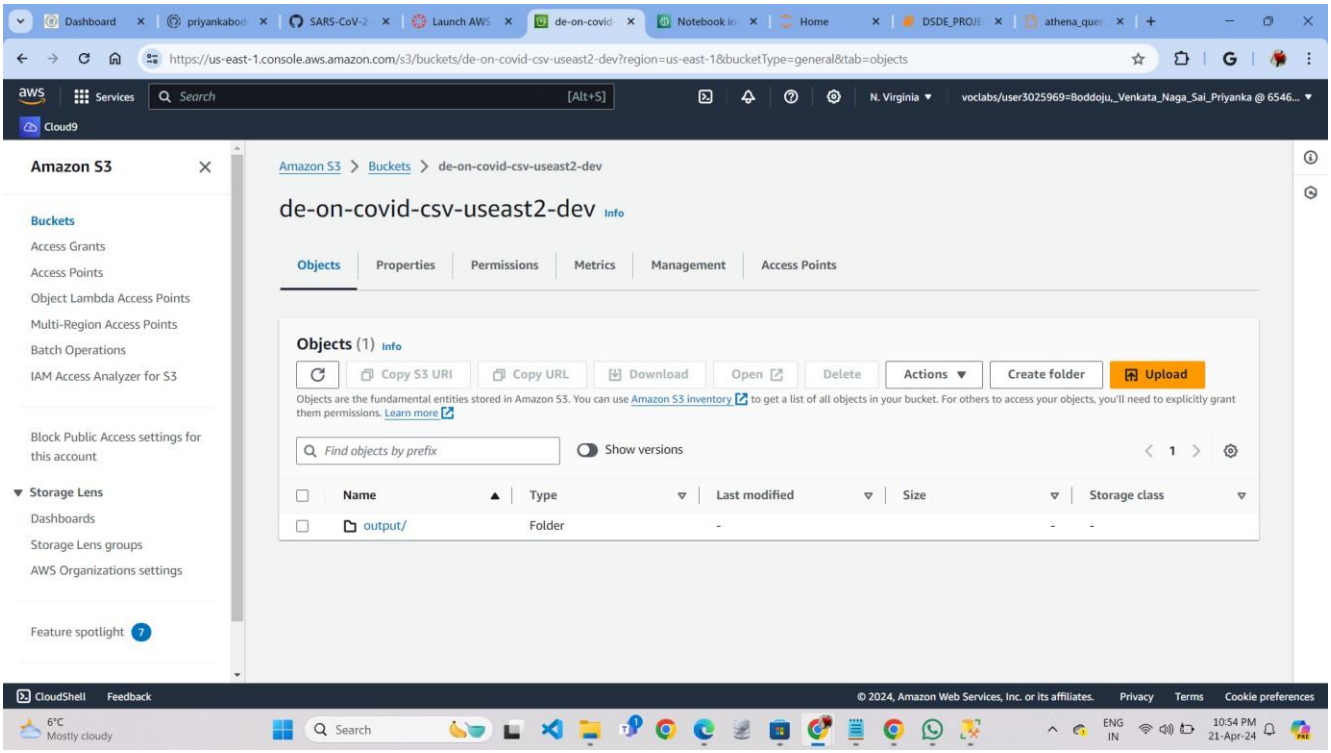


Create AWS S3 Bucket

1. Create a virtual environment.
   python3 -m venv venv
   source venv/bin/activate
2. Install Boto3
   pip install boto3
3. to check if installation successful, run python [type python3] then follow these steps:
   - import boto3
   - s3 = boto3.resource('s3')
   - for bucket in s3.buckets.all():
     print(bucket.name)
4. To be able to upload the data to S3; make sure the data file actually exist in your instance
5. Create python file.
   touch uploadtos3.py
   vim uploadtos3.py

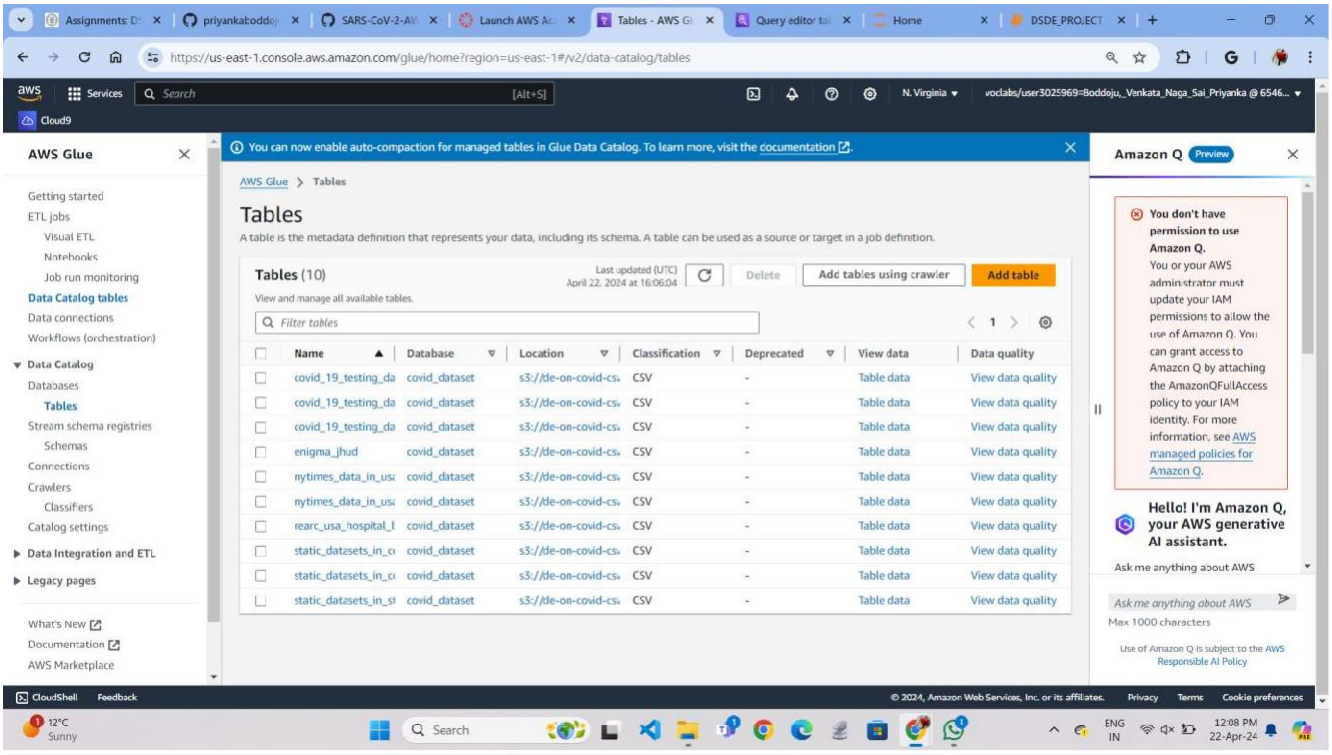
6. Paste the entire code given below.
7. Save and quit (:wq)
8. Run the file.
   Python uploadtos3.py
9. Go to S3[Choose the bucket created by code] and confirm that files has been uploaded.

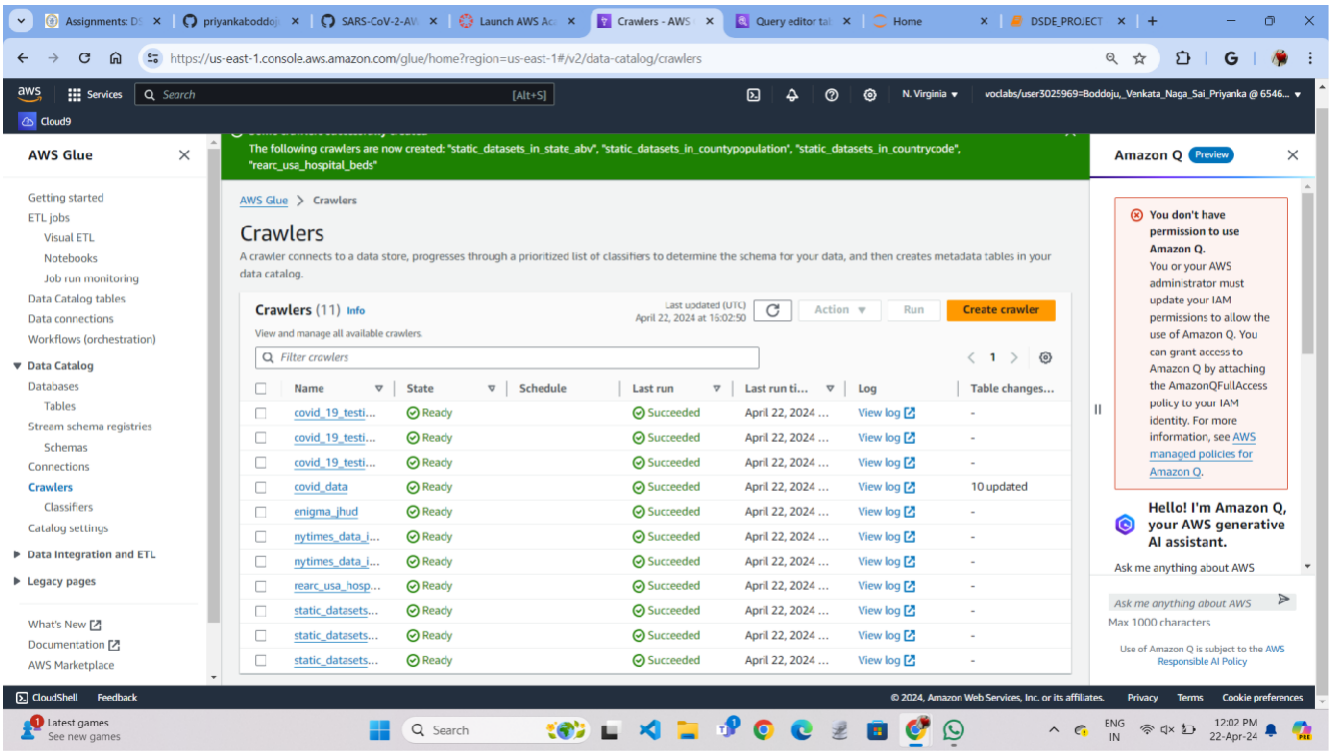




Using AWS Glue:

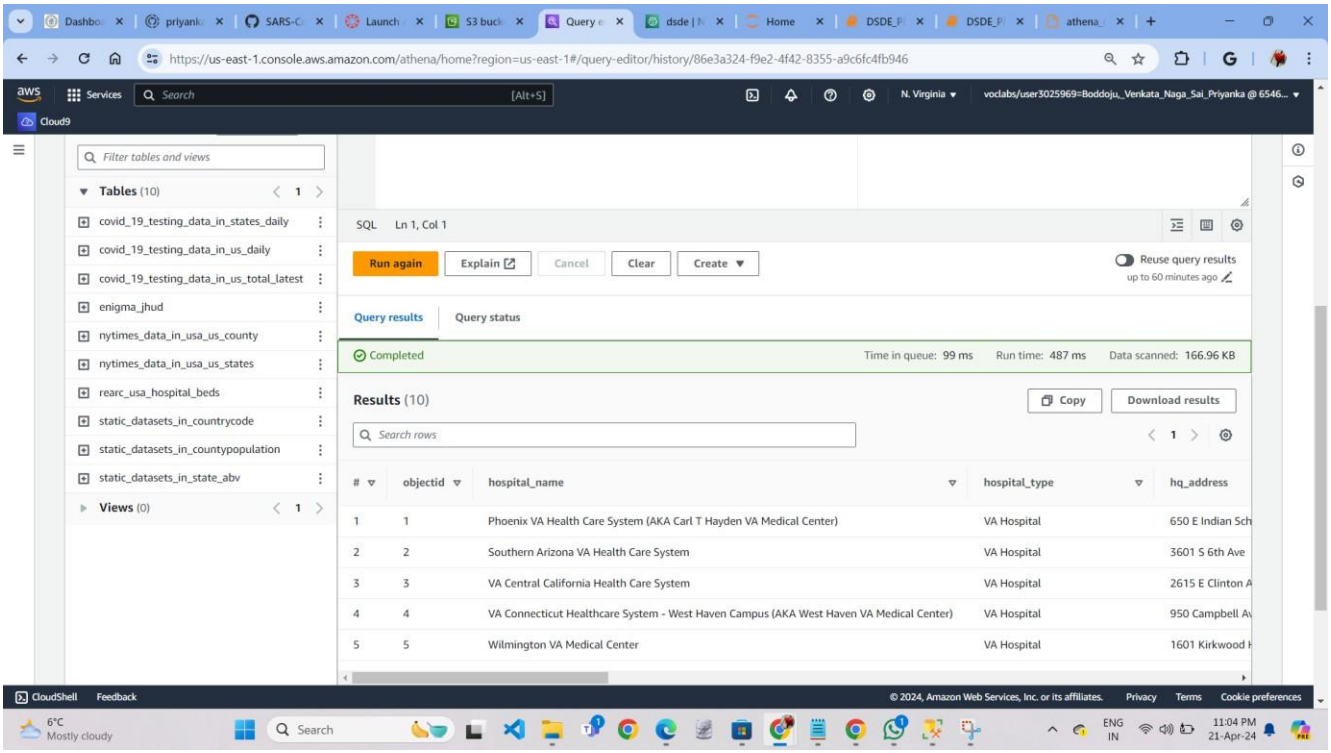
1. In a project leveraging AWS Glue and its crawlers
2. Data is automatically discovered, cataloged, and cleaned.
3. ETL workflows are built visually, reducing manual coding.
4. With serverless execution, resources scale based on workload.
5. Integration with AWS services ensures seamless data pipelines.
6. Enabling efficient data ingestion, transformation, and analysis.


https://us-east-1.console.aws.amazon.com/glue/home?region=us-east-1#/v2/data-catalog/tables
AWS Glue
You can now enable auto-compaction for managed tables in Glue Data Catalog. To learn more, visit the documentation.
AWS Glue > Tables
Tables
A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.
Tables (10)
Delete
Add tables using crawler
Add table
Filter tables
Name
Database
Location
Classification
Deprecated
View data
Data quality
Amazon Q
Preview
You don't have permission to use Amazon Q.
Hello! I'm Amazon Q, your AWS generative AI assistant.
Ask me anything about AWS
Max 1000 characters
Use of Amazon Q is subject to the AWS Responsible AI Policy


https://us-east-1.console.aws.amazon.com/glue/home?region=us-east-1#/v2/data-catalog/crawlers
AWS Glue
The following crawlers are now created: "static_datasets_in_state_abv", "static_datasets_in_countypopulation", "static_datasets_in_countrycode", "rearc_usa_hospital_beds"
AWS Glue > Crawlers
Crawlers
A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.
Crawlers (11) Info
Action
Run
Create crawler
Filter crawlers
Name
State
Schedule
Last run
Last run ti...
Log
Table changes...
Amazon Q
Preview
You don't have permission to use Amazon Q.
Hello! I'm Amazon Q, your AWS generative AI assistant.
Ask me anything about AWS
Max 1000 characters
Use of Amazon Q is subject to the AWS Responsible AI Policy

Using AWS Athena:

In our project, AWS Athena plays a crucial role in facilitating data analysis by providing a serverless querying service for data stored in Amazon S3. With Athena, we can run SQL queries directly against our S3 data without managing any infrastructure, allowing for quick and efficient exploration of our datasets. By integrating with AWS Glue, Athena automates schema discovery, making it easy to get started with querying our data. Its seamless integration with other AWS services and support for various data formats enable us to derive valuable insights and make data-driven decisions effectively.



Database Creation

CREATE DATABASE IF NOT EXISTS covid_dataset

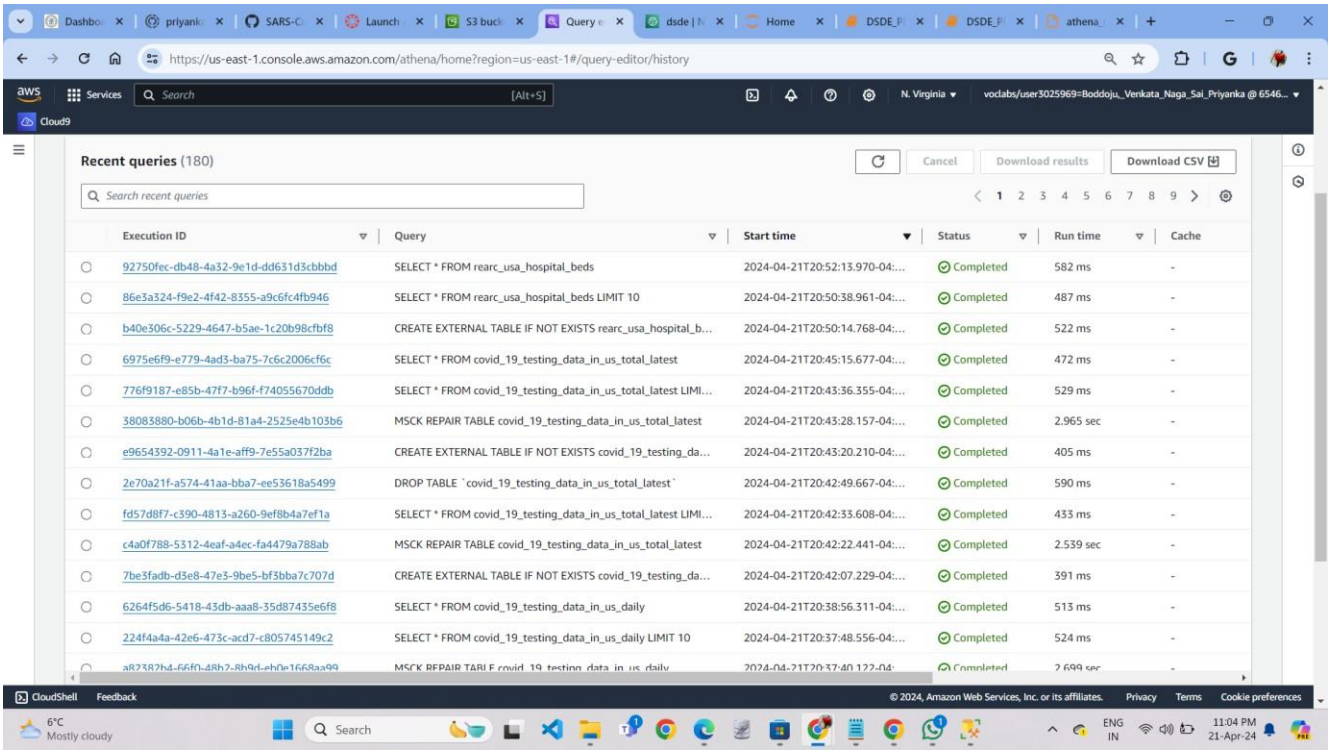
Creating Table

CREATE EXTERNAL TABLE IF NOT EXISTS covid_dataset.enigma_jhu (date DATE, country STRING, country_code STRING, subregion1 STRING, subregion1_code STRING, subregion2 STRING, subregion2_code STRING, aggregate_level INT, new_confirmed INT, new_deceased INT, new_recovered INT, new_tested INT, total_confirmed INT, total_deceased INT, total_recovered INT, total_tested INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION 's3://de-on-covid-csv-useast2-dev-test-bucket/' TBLPROPERTIES ('skip.header.line.count'='1')

Loading Table

MSCK REPAIR TABLE covid_dataset.enigma_jhu

```
CREATE EXTERNAL TABLE IF NOT EXISTS     covid_dataset.nytimes_data_in_usa_us_states
( date STRING, state STRING, fips STRING, cases INT, deaths INT )
ROW FORMAT DELIMITED          FIELDS TERMINATED BY ','
LOCATION 's3://de-on-covid-csv-useast2-dev-test-bucket/us_states/'
TBLPROPERTIES       ('skip.header.line.count'='1')
```



Conclusion

This project presents an efficient and impactful solution for extracting, transforming, and analyzing COVID-19 data, leading to valuable insights critical for informed decision-making and public health interventions. By automating data processes and establishing a centralized database, we have enabled comprehensive analysis of key pandemic parameters, providing stakeholders with timely and accurate information on COVID-19 trends.

The integration of a robust visualization component has facilitated the generation of informative reports and interactive dashboards, empowering individuals to gain a deeper understanding of the pandemic's impact. This visual representation of data supports proactive measures by enabling personal risk assessment and informing the adoption of appropriate preventive measures.

Moving forward, our next steps involve broadening the scope of COVID-19 data sources to include demographic, socioeconomic, and public health indicators. This comprehensive approach will enhance our understanding of the pandemic's multifaceted impact and guide targeted interventions to mitigate its effects on vulnerable populations.

Furthermore, we are committed to enhancing COVID-19 data visualization by developing intuitive, interactive dashboards with advanced charting and mapping tools. Incorporating user feedback and stakeholder input will ensure that our visualizations provide clear, compelling, and actionable insights for informed discussions on pandemic response strategies.

In conclusion, this project underscores the importance of leveraging data-driven approaches and advanced visualization techniques to combat the COVID-19 pandemic effectively. By continuously expanding our data sources and refining our visualization capabilities, we aim to empower stakeholders with the knowledge and tools necessary to navigate the evolving challenges posed by the pandemic and safeguard public health and well-being.