**Data Engineering – Final Assessment**

Contents

[**Data Dictionary** 2](#_Toc137751991)

[**Tasks** 3](#_Toc137751992)

[**Data Discovery , Processing , Loading Task** 3](#_Toc137751993)

[Task 1 3](#_Toc137751994)

[a. Move the dataset onto S3 3](#_Toc137751995)

[b. Create Glue database and 2 crawlers to crawl CSV and JSON folders using AWS CLI 3](#_Toc137751996)

[c. Run the Crawlers from CLI 6](#_Toc137751997)

[d. Confirm the data has been able to query from Athena 8](#_Toc137751998)

[Task 2 10](#_Toc137751999)

[a. Deploy GLUE locally 10](#_Toc137752000)

[b. Develop a GLUE JOB to convert the CSV files to PARQUET from Local Development 11](#_Toc137752001)

[c. Create crawler using AWS CLI on the parquet file 17](#_Toc137752002)

[d. Run the crawler from CLI and confirm data from Athena 18](#_Toc137752003)

[Task 3 20](#_Toc137752004)

[a. Run basic Analytical queries on the dataset using Athena from AWS Console 20](#_Toc137752005)

[**Data Versioning Task** 21](#_Toc137752006)

[Task 1 Create a repository in github and upload the dataset 21](#_Toc137752007)

[a. Create a repository in github.com 21](#_Toc137752008)

[Task 2 Generate the Reports and upload them to github 22](#_Toc137752009)

[a. Clone the repository that you created in task 1 22](#_Toc137752010)

[b. Create a branch for this activity ,Create a file with all the queries from the above tasks 22](#_Toc137752011)

[c. List the version history ( commit history) , status of the local repository 23](#_Toc137752012)

[d. Commit the file to the branch 23](#_Toc137752013)

[e. Push the changes to the github 24](#_Toc137752014)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pushpalatha Anbarasu**

# **Data Dictionary**

Register on AWS Data Exchange for COVID-19 data set

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

# **Tasks**

# **Data Discovery , Processing , Loading Task**

## Task 1

### a. Move the dataset onto S3

My S3 bucket with Data set: Coronavirus Disease (COVID-19) - Testing DataA screenshot of a computer

Description automatically generated

### b. Create Glue database and 2 crawlers to crawl CSV and JSON folders using AWS CLI

aws glue create-database --database-input Name="de-project-db" (creating database)

aws glue get-databases (listing the databases)A screenshot of a computer

Description automatically generated

**Create crawler for CSV files**

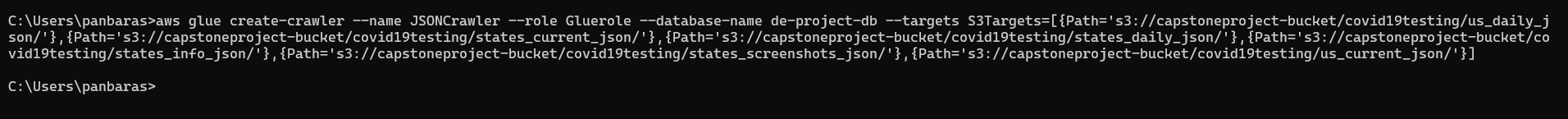
aws glue create-crawler --name CSVCrawler --role Gluerole --database-name de-project-db --targets S3Targets=[{Path='s3://capstoneproject-bucket/covid19testing/states\_current\_csv/'},{Path='s3://capstoneproject-bucket/covid19testing/states\_daily\_csv/'},{Path='s3://capstoneproject-bucket/covid19testing/states\_info\_csv/'},{Path='s3://capstoneproject-bucket/covid19testing/states\_screenshots\_csv/'},{Path='s3://capstoneproject-bucket/covid19testing/us\_current\_csv/'},{Path='s3://capstoneproject-bucket/covid19testing/us\_daily\_csv/'}]



aws glue get-crawler --name CSVCrawler A screenshot of a computer screen

Description automatically generated with medium confidence

**Create crawler for JSON files**

aws glue create-crawler --name JSONCrawler --role Gluerole --database-name de-project-db --targets S3Targets=[{Path='s3://capstoneproject-bucket/covid19testing/us\_daily\_json/'},{Path='s3://capstoneproject-bucket/covid19testing/states\_current\_json/'},{Path='s3://capstoneproject-bucket/covid19testing/states\_daily\_json/'},{Path='s3://capstoneproject-bucket/covid19testing/states\_info\_json/'},{Path='s3://capstoneproject-bucket/covid19testing/states\_screenshots\_json/'},{Path='s3://capstoneproject-bucket/covid19testing/us\_current\_json/'}] 

aws glue get-crawler --name JSONCrawler A screenshot of a computer

Description automatically generated with medium confidence

### c. Run the Crawlers from CLI

**Start the CSVCrawler**

aws glue start-crawler --name CSVCrawler

A black screen with white text

Description automatically generated with low confidence

aws glue get-crawler-metrics (6 tables got created by CSVcrawler)A screenshot of a computer screen

Description automatically generated with medium confidence

**Start the JSONCrawler**

aws glue start-crawler --name JSONCrawler

A black background with white text

Description automatically generated with low confidence

aws glue get-crawler-metrics (6 tables got created by JSONcrawler)A screenshot of a computer

Description automatically generated

Log from consoleA screenshot of a computer

Description automatically generated

### d. Confirm the data has been able to query from Athena

Query the table created from CSV dataA screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

Query the table created from JSON dataA screenshot of a computer

Description automatically generated

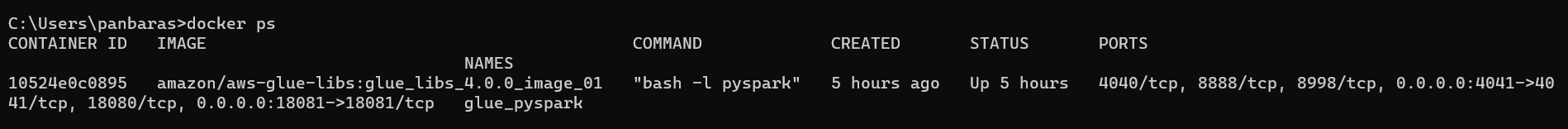
A screenshot of a computer

Description automatically generated with medium confidence

## Task 2

### a. Deploy GLUE locally

**Docker image for AWS Glue version 4.0**

****

docker run -it -v %WORKSPACE\_LOCATION%\aws:/home/glue\_user/.aws -v %WORKSPACE\_LOCATION%:/home/glue\_user/workspace/ -e AWS\_PROFILE=%PROFILE\_NAME% -e DISABLE\_SSL=true --rm -p 4041:4041 -p 18081:18081 --name glue\_pyspark amazon/aws-glue-libs:glue\_libs\_4.0.0\_image\_01 pyspark

A black screen with white text

Description automatically generated with low confidence

Visual Studio code with AWS glue image container

A screenshot of a computer

Description automatically generated with medium confidence

### b. Develop a GLUE JOB to convert the CSV files to PARQUET from Local Development

**Pyspark script for GLUE job:**

**csv\_to\_parquet.py (uploaded this script in S3 location)**

import sys

from awsglue.transforms import \*

from awsglue.utils import getResolvedOptions

from awsglue.context import GlueContext

from pyspark.context import SparkContext

from pyspark.sql import SparkSession

# Create a SparkContext

sc = SparkContext()

# Create a GlueContext and SparkSession

glueContext = GlueContext(sc)

spark = glueContext.spark\_session

# Get the job parameters

args = getResolvedOptions(sys.argv, ['JOB\_NAME', 'input\_path', 'output\_path'])

# Get the input and output paths from Glue job parameters

input\_path = args['input\_path']

output\_path = args['output\_path']

# Read the CSV files from the input path using Spark DataFrame API

input\_data = spark.read.csv(input\_path, header=True, inferSchema=True)

# Convert CSV to Parquet

input\_data.write.parquet(output\_path)

# Stop the SparkContext

sc.stop()

**Glue job creation**

aws glue create-job --name "CSVToParquetJob" --role "arn:aws:iam::667409444611:role/project-glue-service-role" --command '{"Name": "glueetl", "ScriptLocation":"s3://capstoneproject-bucket/covid19testing/csv\_to\_parquet.py"}'

Glue job “CSVTOParquerJob” got created using CLI and details of it can be seen using “aws glue get-jobs”

A screenshot of a computer program

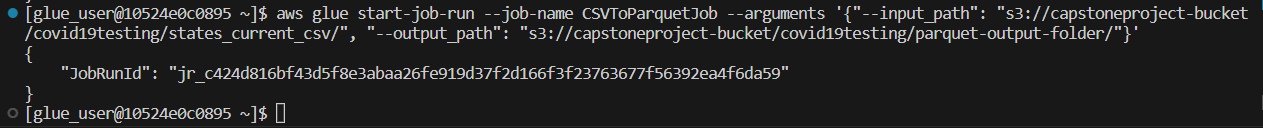
Description automatically generated with medium confidence

Created job shown in AWS GLUE StudioA screenshot of a computer

Description automatically generated

**Run the job**

aws glue start-job-run --job-name CSVToParquetJob --arguments '{"--input\_path": "s3://capstoneproject-bucket/covid19testing/states\_current\_csv/", "--output\_path": "s3://capstoneproject-bucket/covid19testing/parquet-output-folder/"}'



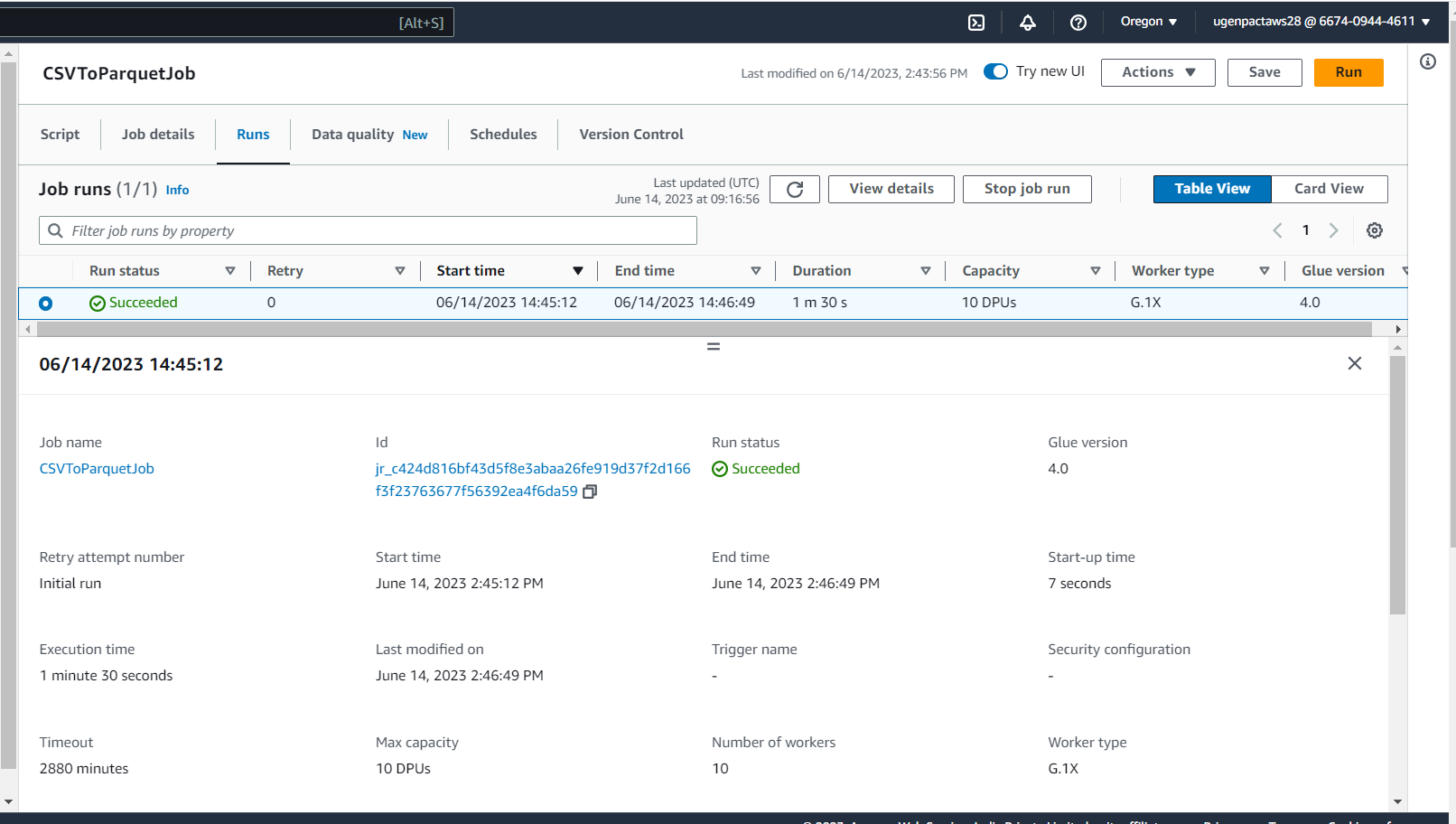
**Job run status**

aws glue get-job-run --job-name CSVToParquetJob --run-id jr\_c424d816bf43d5f8e3abaa26fe919d37f2d166f3f23763677f56392ea4f6da59

A computer code on a black background

Description automatically generated with low confidence

Job run status from AWS GLUE STUDIO



Converted parquet file got populated in S3 output folder A screenshot of a computer

Description automatically generated with medium confidence

**Run the same job for multiple csv files in the source**

S3 folder with multiple csv files

**A screenshot of a computer

Description automatically generated**

aws glue start-job-run --job-name CSVToParquetJob --arguments '{"--input\_path": "s3://capstoneproject-bucket/covid19testing/dataset/csv/", "--output\_path": "s3://capstoneproject-bucket/covid19testing/parquet-output-multiple\_files/"}' A computer code on a black background

Description automatically generated with low confidence

**Job run status**

aws glue get-job-run --job-name CSVToParquetJob --run-id jr\_c8487edfe774e2afd6cbb70baeca84e40daac56ae8d9241e5dae677c850f163e A picture containing text, screenshot, software, font

Description automatically generated

Job run status from AWS GLUE STUDIO A screenshot of a computer

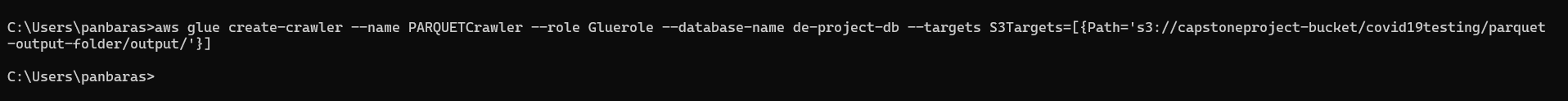
Description automatically generated

Converted Parquet files got populated in destination A screenshot of a computer

Description automatically generated

### c. Create crawler using AWS CLI on the parquet file

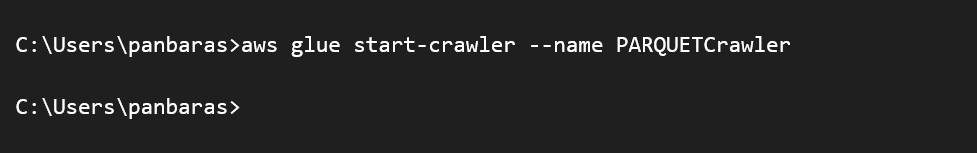
**Create PARQUETCrawler**

aws glue create-crawler --name PARQUETCrawler --role Gluerole --database-name de-project-db --targets S3Targets=[{Path='s3://capstoneproject-bucket/covid19testing/parquet-output-folder/output/'}] ****aws glue get-crawler --name PARQUETCrawler (details of crawler)A screenshot of a computer program

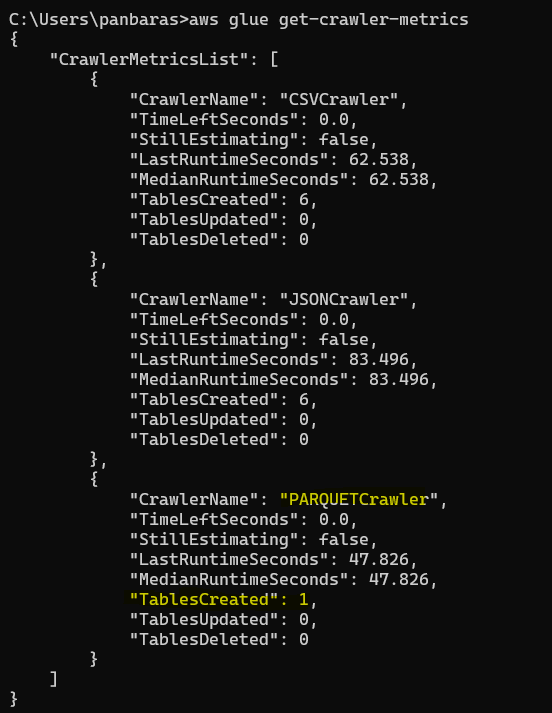
Description automatically generated with medium confidence

### d. Run the crawler from CLI and confirm data from Athena

aws glue start-crawler --name PARQUETCrawler



aws glue get-crawler-metrics (1 table got created by PARQUETcrawler)

****

Log from console A screenshot of a computer

Description automatically generated

Query data from Athena**A screenshot of a computer

Description automatically generated**

A screenshot of a computer

Description automatically generated with medium confidence

## Task 3

### a. Run basic Analytical queries on the dataset using Athena from AWS Console

--Top 5 state with highest number of positive results A screenshot of a computer

Description automatically generated with medium confidence

--State with highest number of people hospitalized A screenshot of a computer

Description automatically generated with medium confidence

--Total number of positive and negative cases per day A screenshot of a computer

Description automatically generated with medium confidence

# **Data Versioning Task**

## Task 1 Create a repository in github and upload the dataset

### a. Create a repository in github.com

A screenshot of a computer

Description automatically generated with medium confidence

Uploaded datasets to repository

A screenshot of a chat

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

## Task 2 Generate the Reports and upload them to github

### a. Clone the repository that you created in task 1

git clone <https://github.com/lathatirupur/Capstone_Project.git>

A computer screen with white text

Description automatically generated with low confidence

### b. Create a branch for this activity ,Create a file with all the queries from the above tasks

git branch capstone\_project\_activity

git checkout capstone\_project\_activity

A screen shot of a computer program

Description automatically generated with low confidence

### c. List the version history ( commit history) , status of the local repository

Added a file in local repositoryA screenshot of a computer

Description automatically generated

Display the commit history of the repository

git log

A picture containing text, screenshot, font

Description automatically generated

Status of the local repository

git status

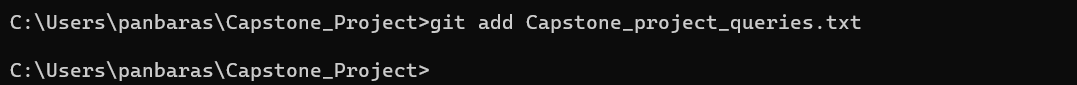
A screenshot of a computer screen

Description automatically generated with low confidence

### d. Commit the file to the branch

Add the file to the staging area

git add Capstone\_project\_queries.txt



git status

A screen shot of a computer program

Description automatically generated with low confidence

Commit the changes

git commit -m "Add project queries file"

A screenshot of a computer error

Description automatically generated with low confidence

### e. Push the changes to the github

git push origin activity-branch

A screenshot of a computer program

Description automatically generated with medium confidence

Version history

A screen shot of a computer

Description automatically generated with low confidence

Change got pushed to activity-branch in github

A screenshot of a computer

Description automatically generated with medium confidence