Batch Load Data with Apache Airflow and Spark

by Andrea Zhang
2022 Aug

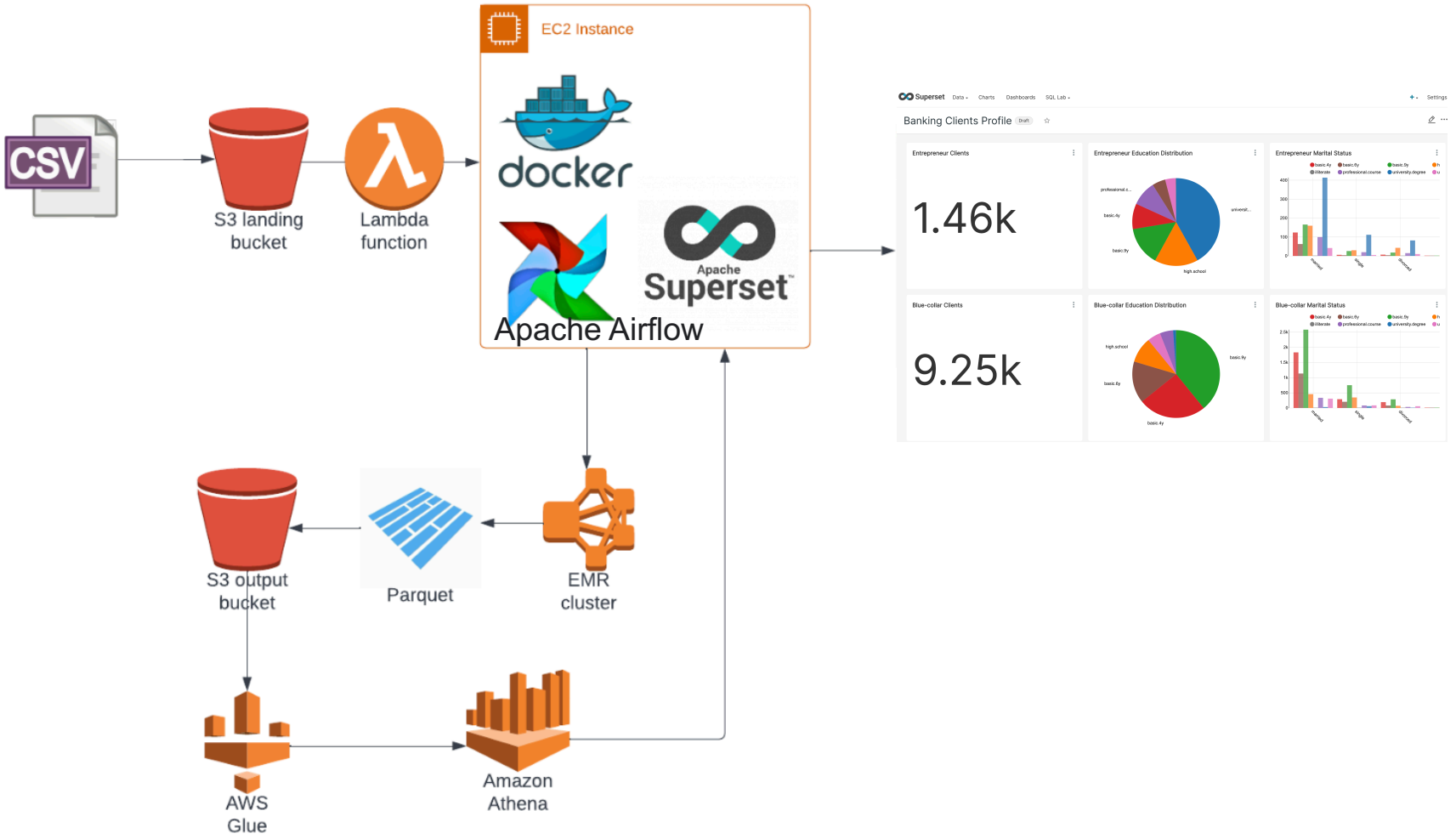
Project Motivation

Objective: The primary focus for this project is to build a batch processing application that partition a given file based on the given format and save into several distributed files in S3 bucket for analytics and visualization.

For this project, multiple big data applications were used, including Docker, Airflow, Lambda function, Spark (EMR), AWS Glue, AWS Athena and visualization layer represented by Superset. The data source used in the project is public banking data.

Conceptual Data Streaming Architecture

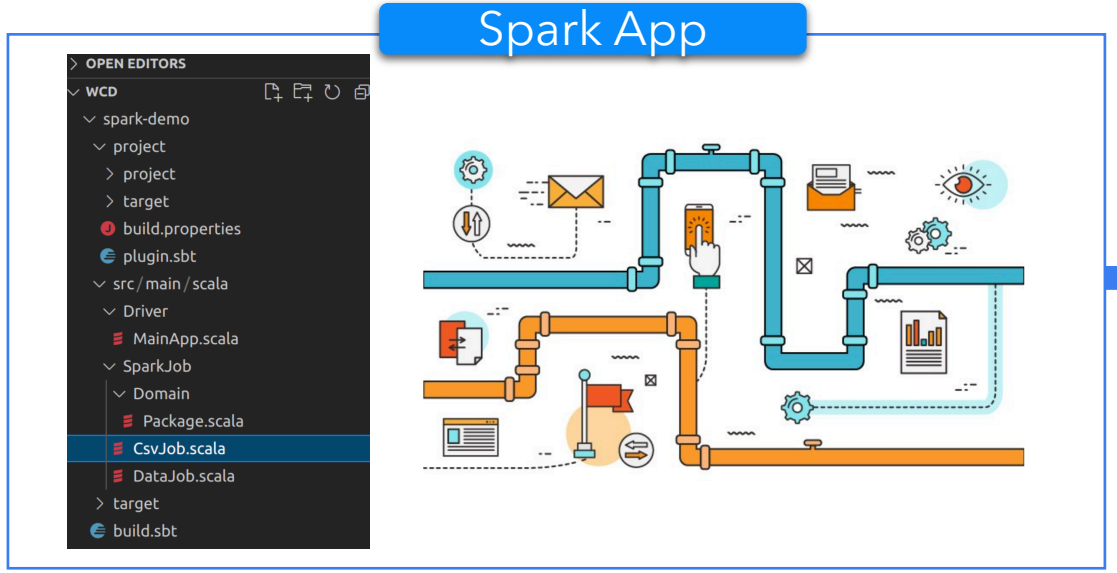
Data Collection | Transformation | Storage | Visualization



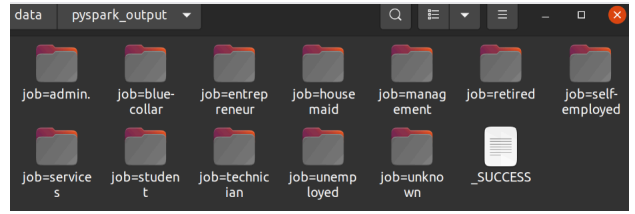
Data Transformation Spark

Compiling Spark Jobs Built in Scala. The Spark application transforms CSV files into multiple parquet files partitioned by the job type column, and export parquet files to the S3 bucket.

input



output



Partitioned by Job Type

Deploy a Spark job:

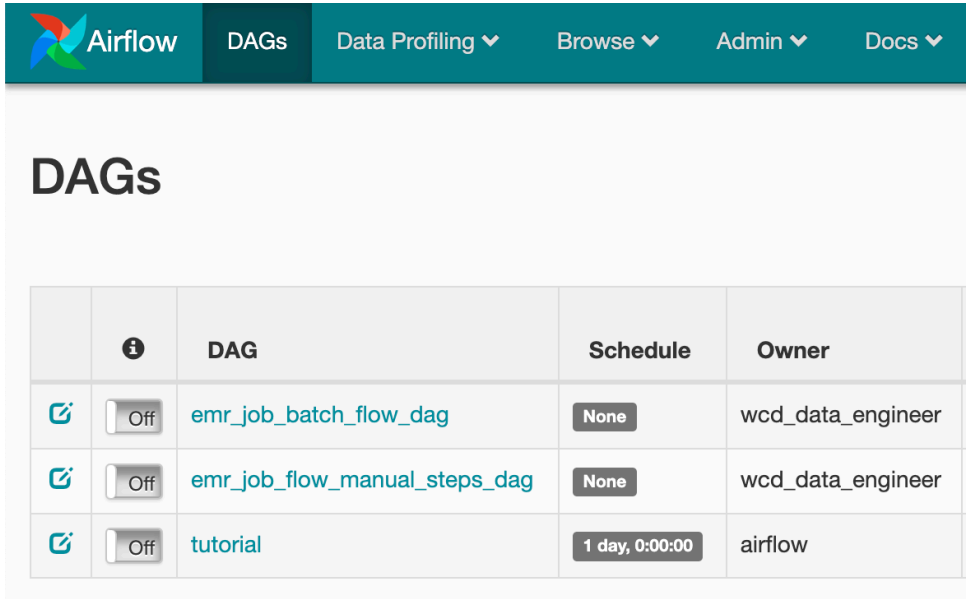
- Get the latest project files
- Download dependencies
- Compile and Package Spark application
- Upload .jar file to an S3 bucket
- Spark-submit using the s3 file to the Spark (EMR cluster) master node

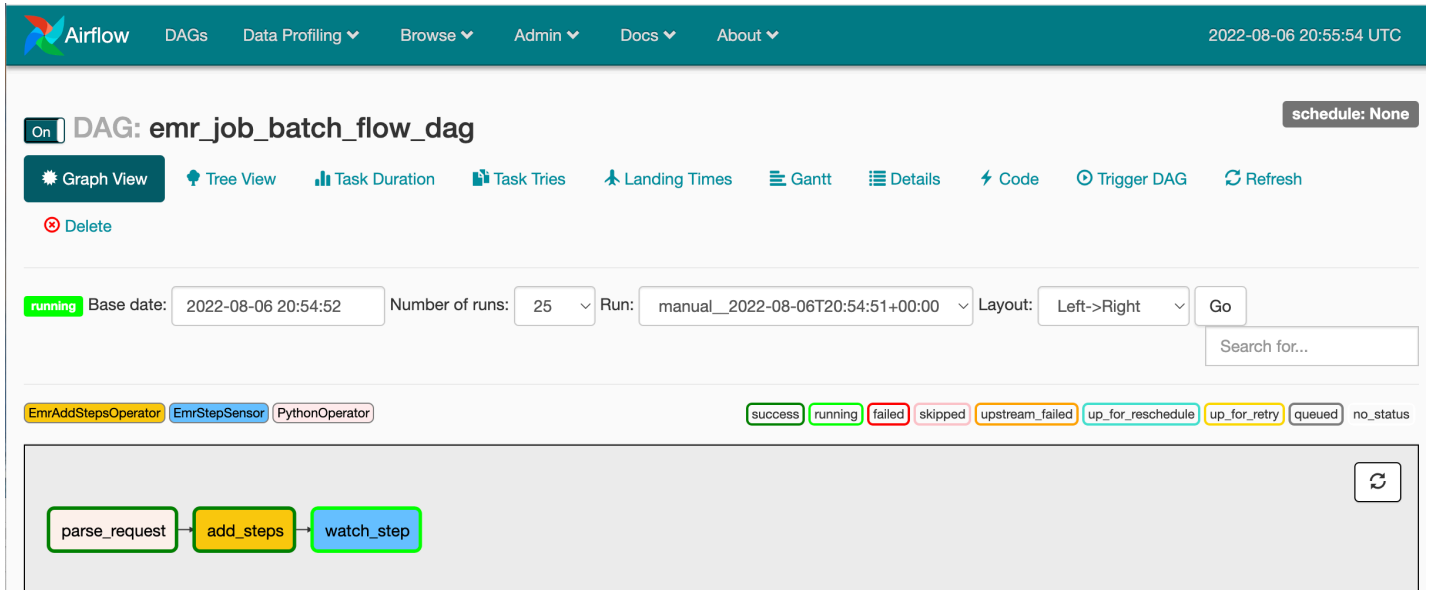
Data Transformation – Apache Airflow

Airflow enable the execution of a collection of task in a more organized way using DAG.

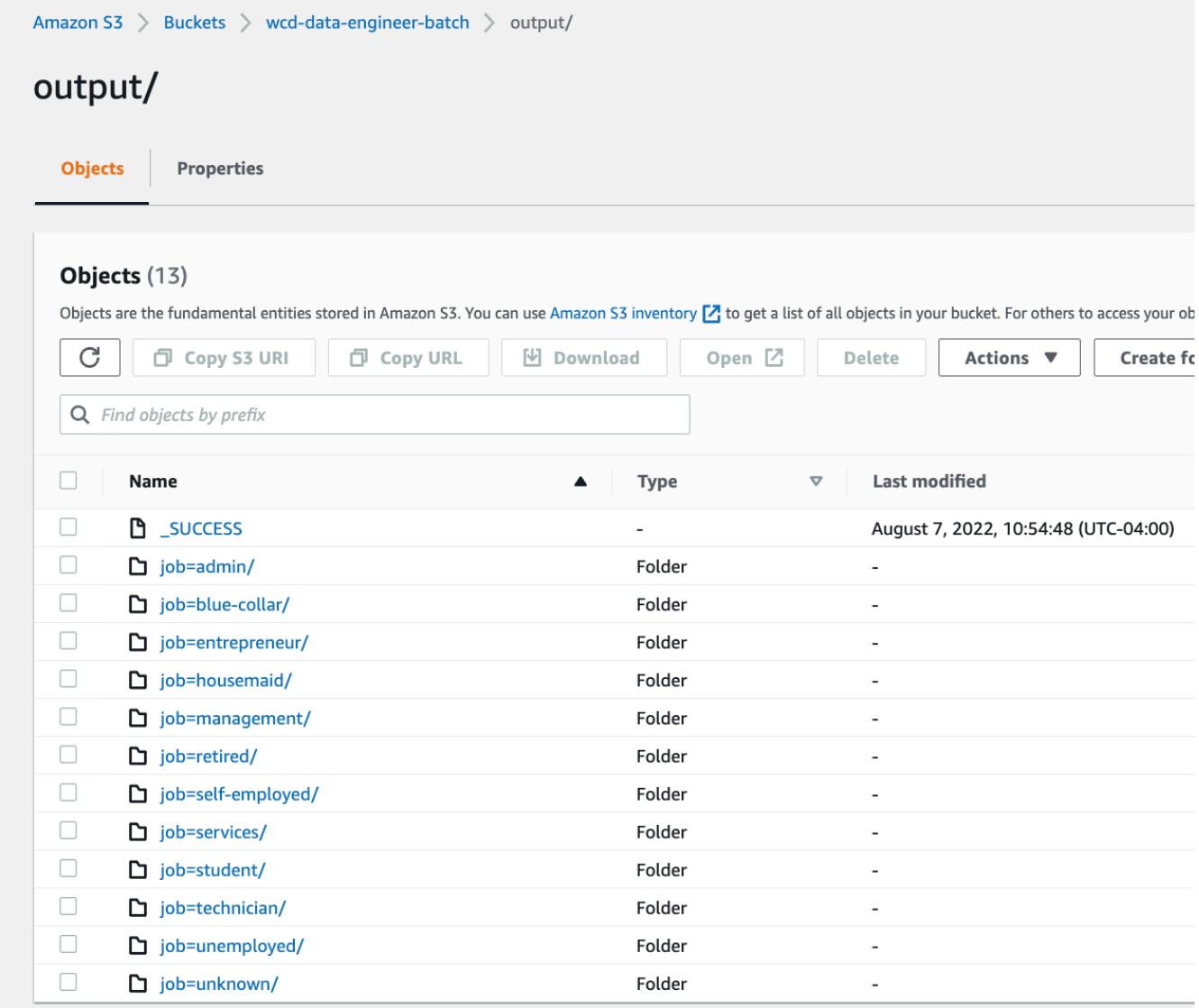
- ➢ Define DAG in Python script and execute the script in a Docker container

- ✓ Schedule workflows
- ✓ Monitor workflows
- ✓ Scalability

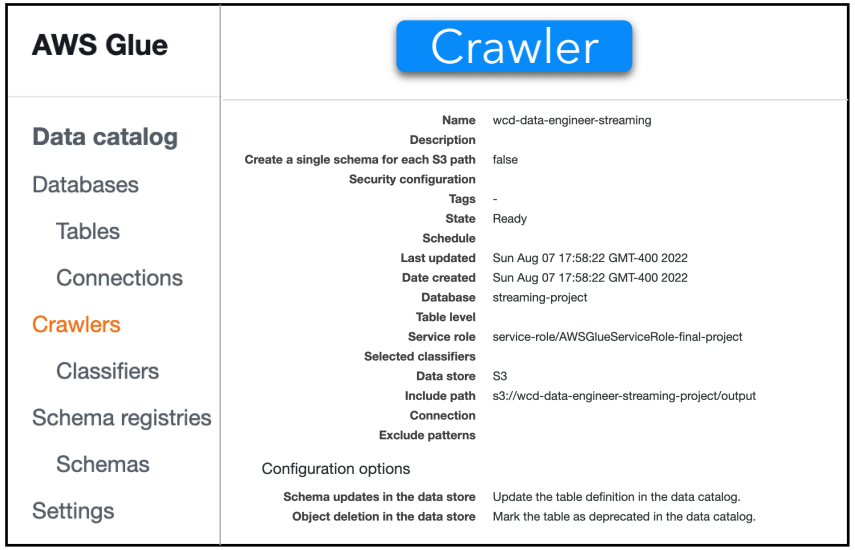


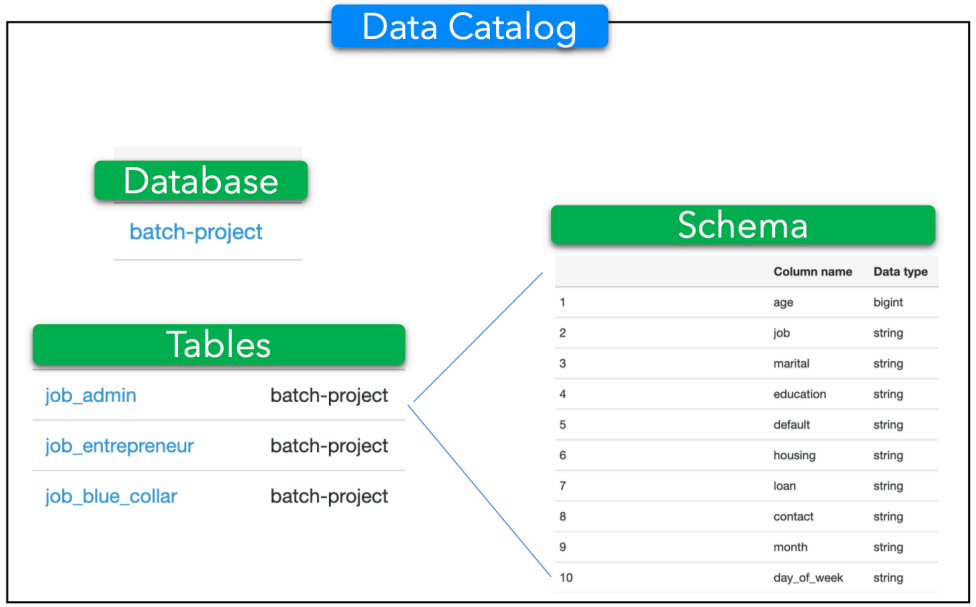


Spark Output – Parquet Files



Data Integration – AWS Glue





Glue provides serverless data integration that prepare and combine data for analytics.

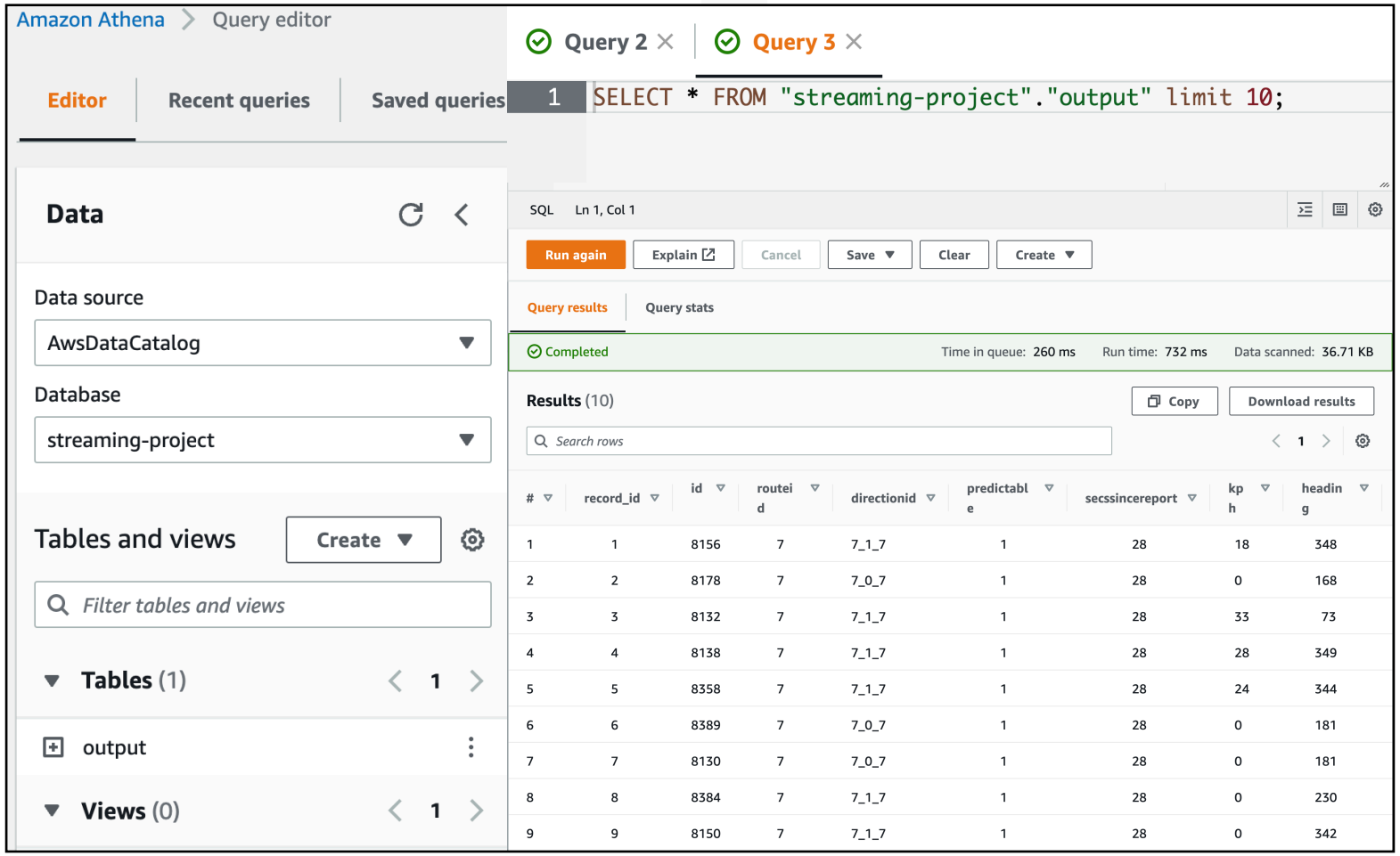
The Glue data Catalog serves as a central metadata repository.

✓ Discover and search across multiple data sets using Glue data Catalog.

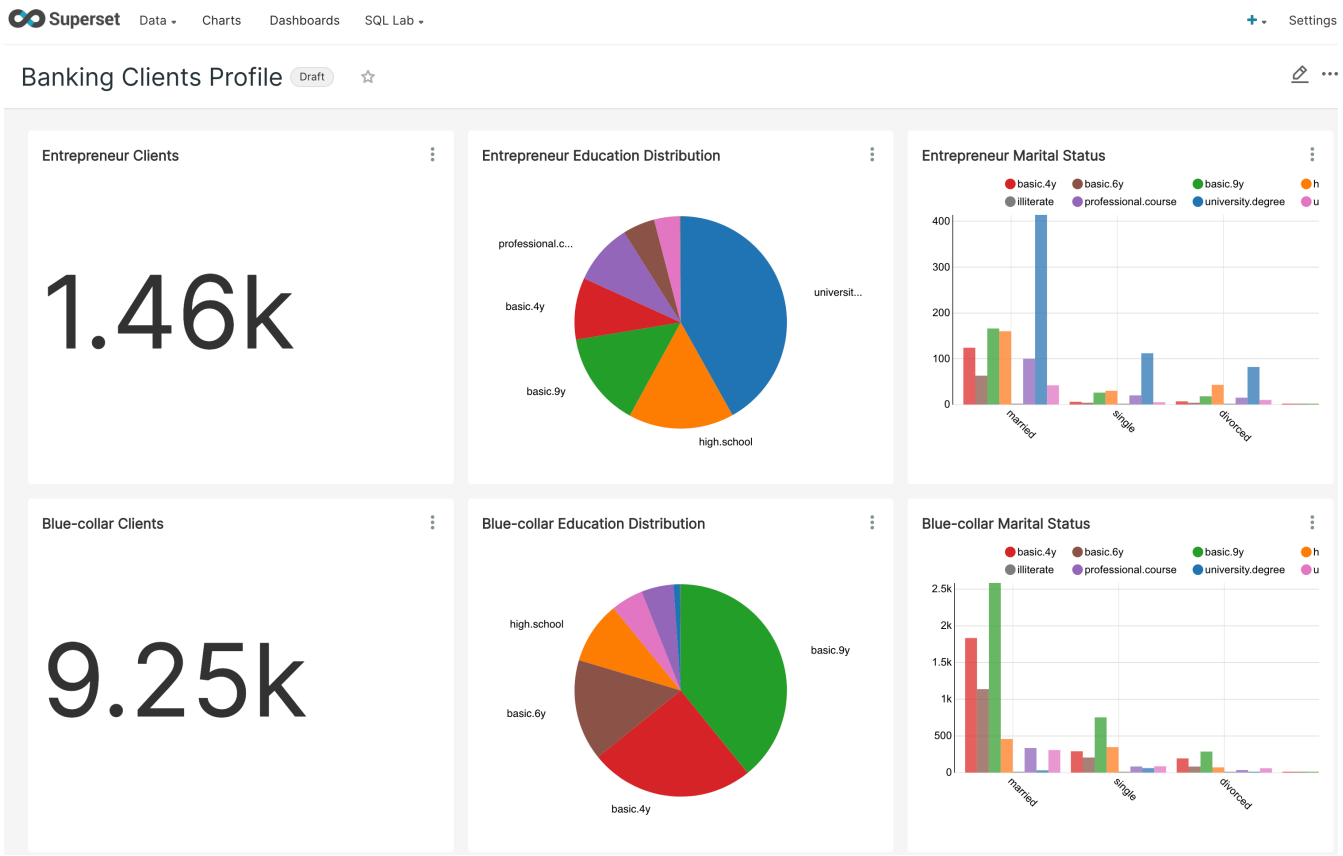
✓ Once the data is cataloged, it is immediately available for search and query using Amazon Athena.

Data Analytics – AWS Athena

Connect Glue Data Catalog and point to the streaming project database and enable querying the bus data instantly.



Data Visualization - Superset

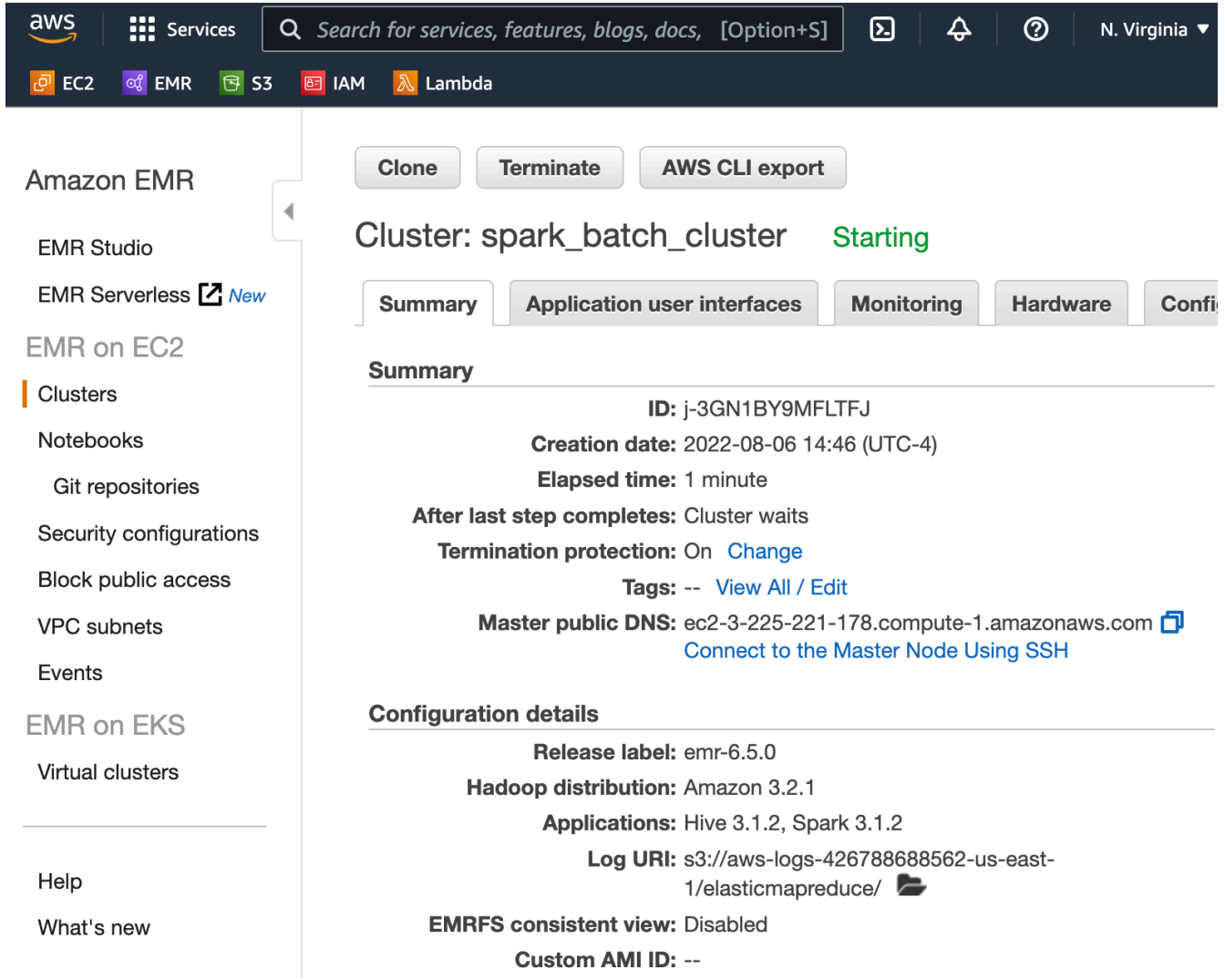


Ingesting Data From Kafka with Spark Streaming

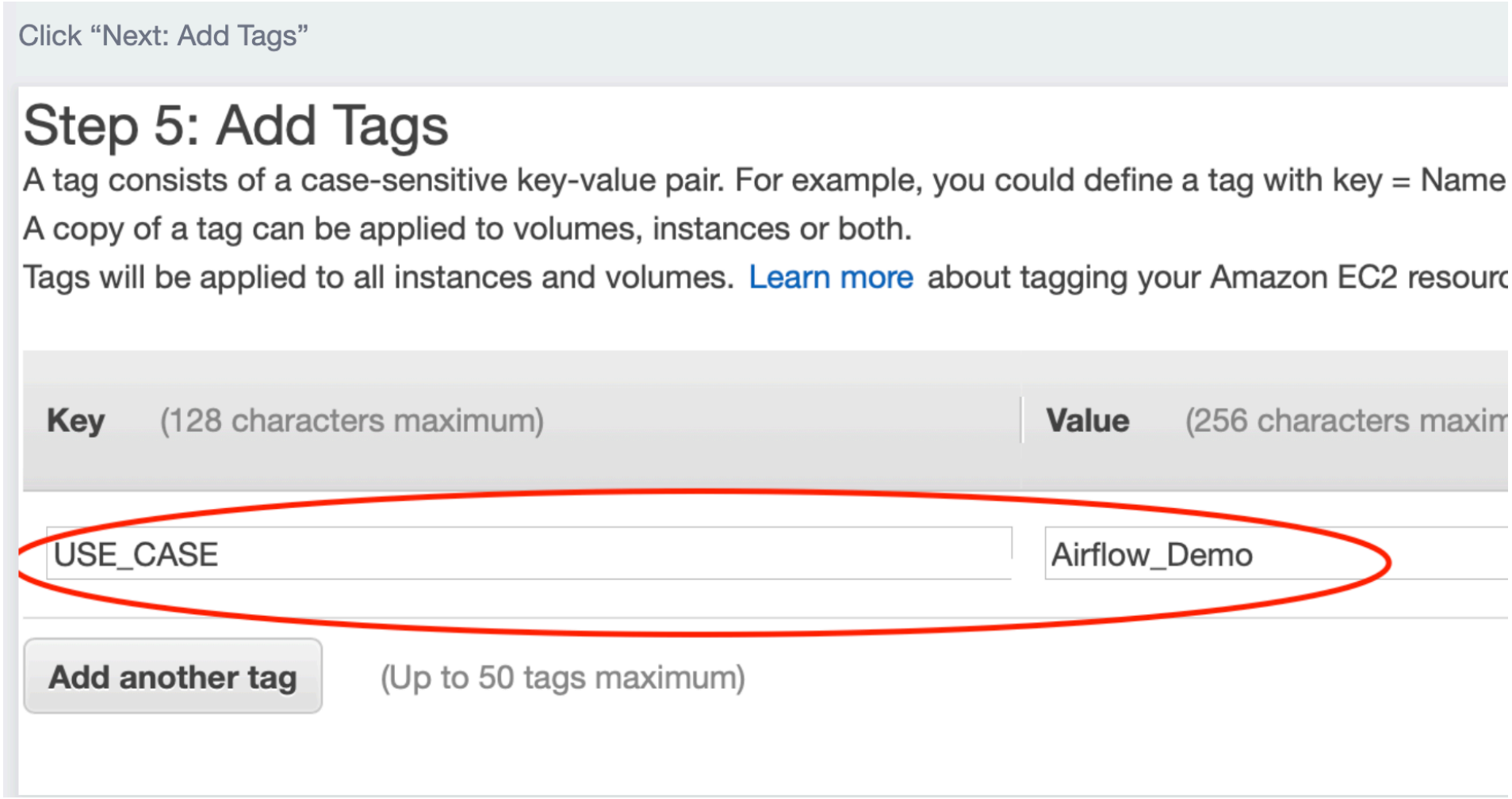
- ➢ Create an AWS EMR and enable big data tools such as Apache spark, Hive, and Hudi.
- ➢ Write a Spark job using Scala and assembly into an application jar file
- ✓ Scalable
- ✓ Fault-tolerant
- ✓ Micro batches of data coming from the Kafka topic

Appendix

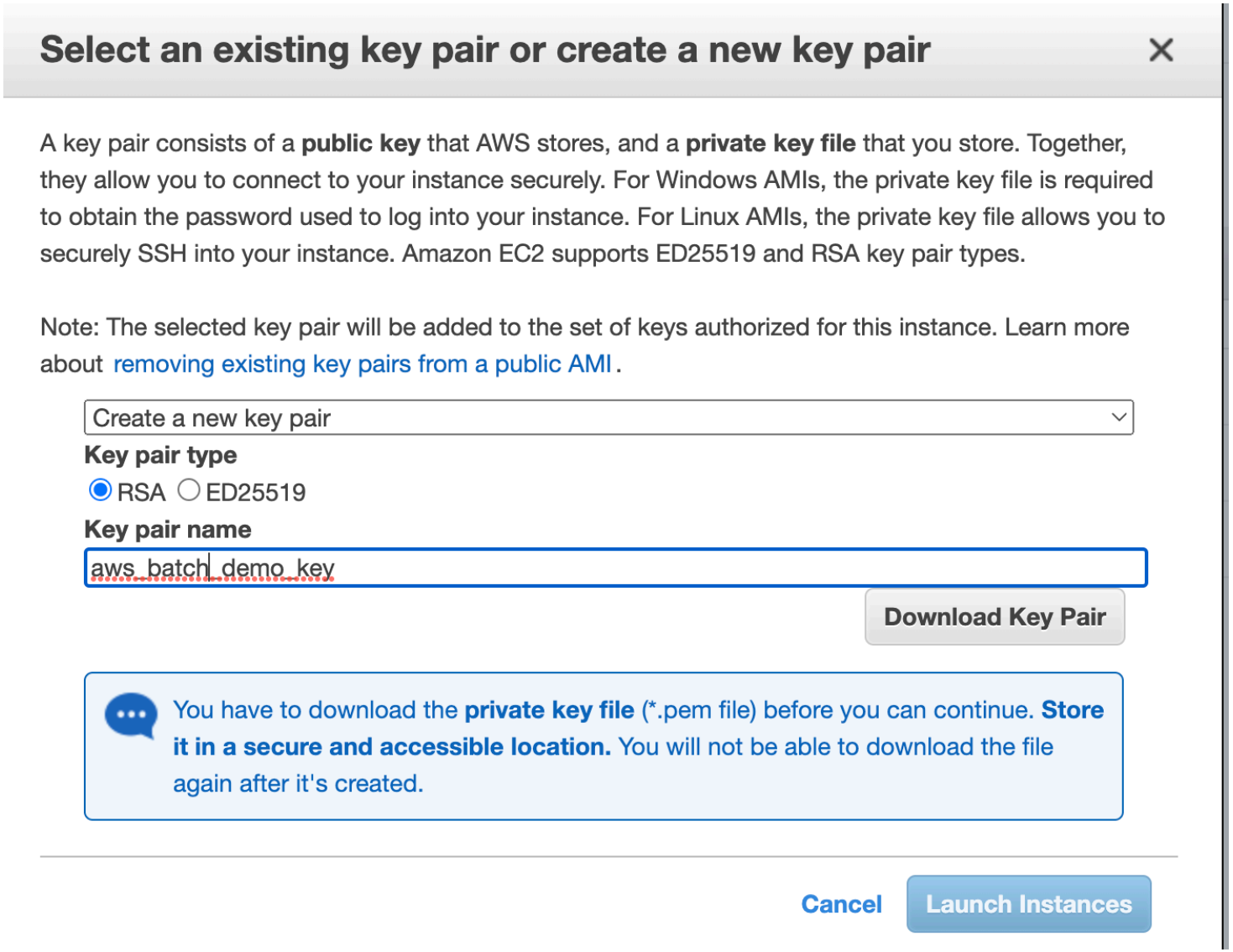
AWS – Set Up Spark Cluster (EMR)



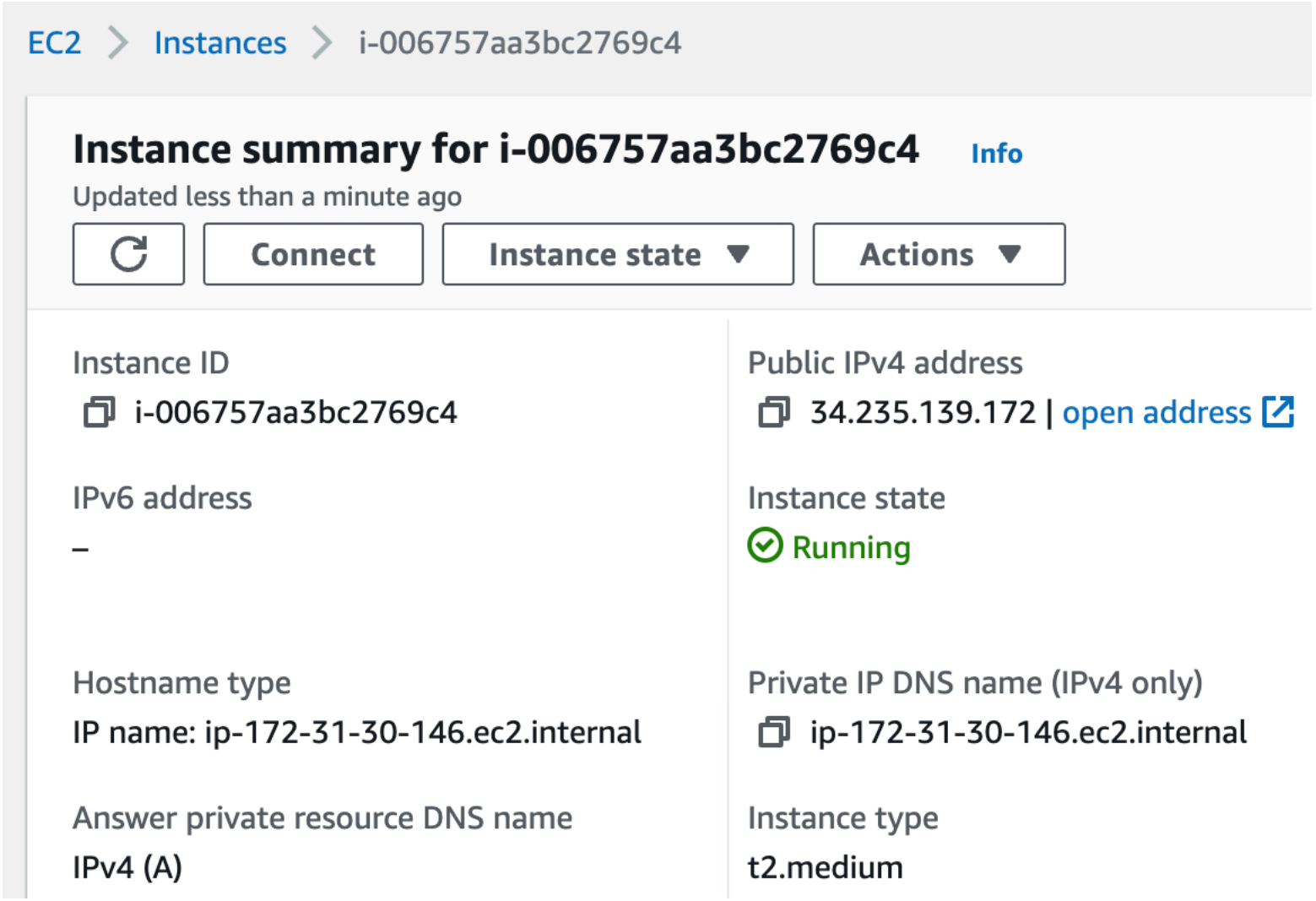
AWS – Set Up Airflow Connection



AWS – Configure Network (Key Pair)

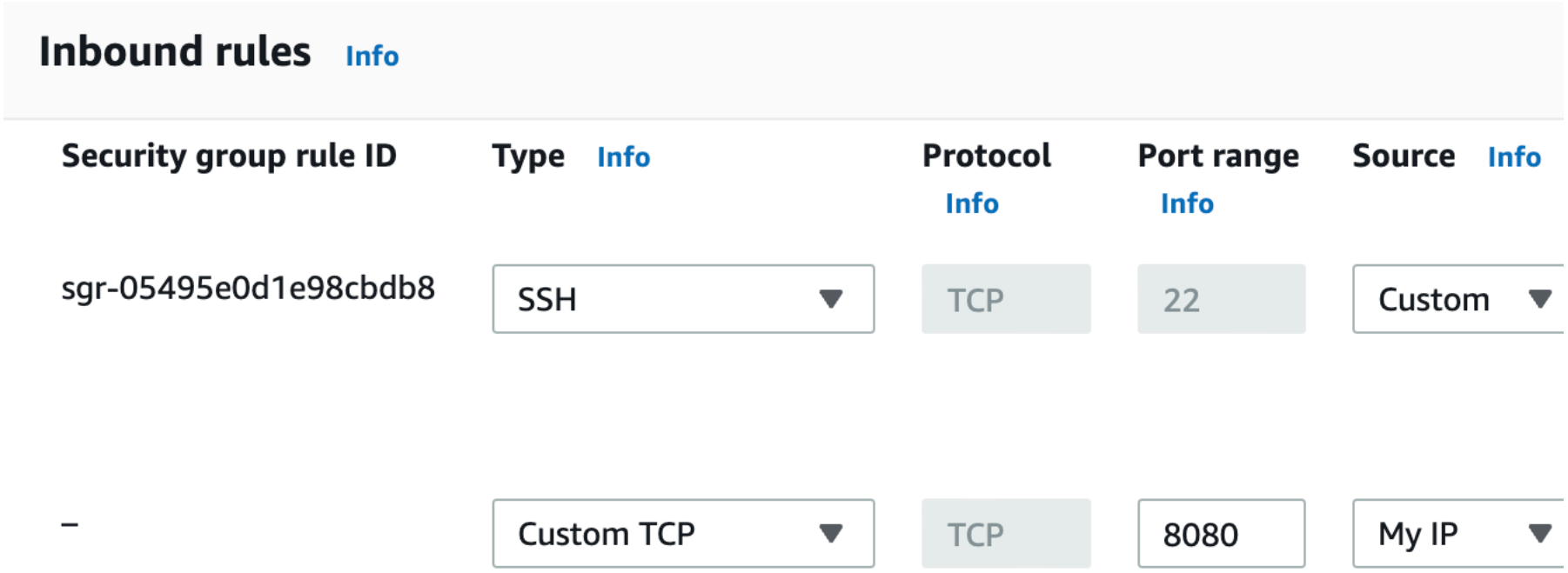


AWS - Set Up EC2 Instance



Update Network Inbound Rules

In EC2, white list my IP address and open port 8080. The airflow image will be connected using port 8080.



SSH to EC2 Using Command Line

```
(base) #####@yings-iMac ~ % mv ~/Downloads/aws_batch_demo_key.pem ~/.ssh
(base) #####@yings-iMac ~ % chmod 400 ~/ssh/aws_batch_demo_key.pem
chmod: /Users/#####/ssh/aws_batch_demo_key.pem: No such file or directory
(base) #####@yings-iMac ~ % chmod 400 ~/.ssh/aws_batch_demo_key.pem
(base) #####@yings-iMac ~ % ssh -i ~/.ssh/aws_batch_demo_key.pem ec2-
user@34.235.139.172
The authenticity of host '34.235.139.172 (34.235.139.172)' can't be established.
ED25519 key fingerprint is SHA256:mbdnrLT6zmkRuAdG3a6LExeJW7zwU9jrACSxIL3D3XE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.235.139.172' (ED25519) to the list of known hosts.


       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-30-146 ~]$
```

Airflow DAG Runs