

Data Engineer Project - Dibimbang

CREDIT DATA PIPELINE

Muhammad Arij Arfina



Content

01

Problems

02

Data Pipeline

03

Requirements & Steps

04

Output



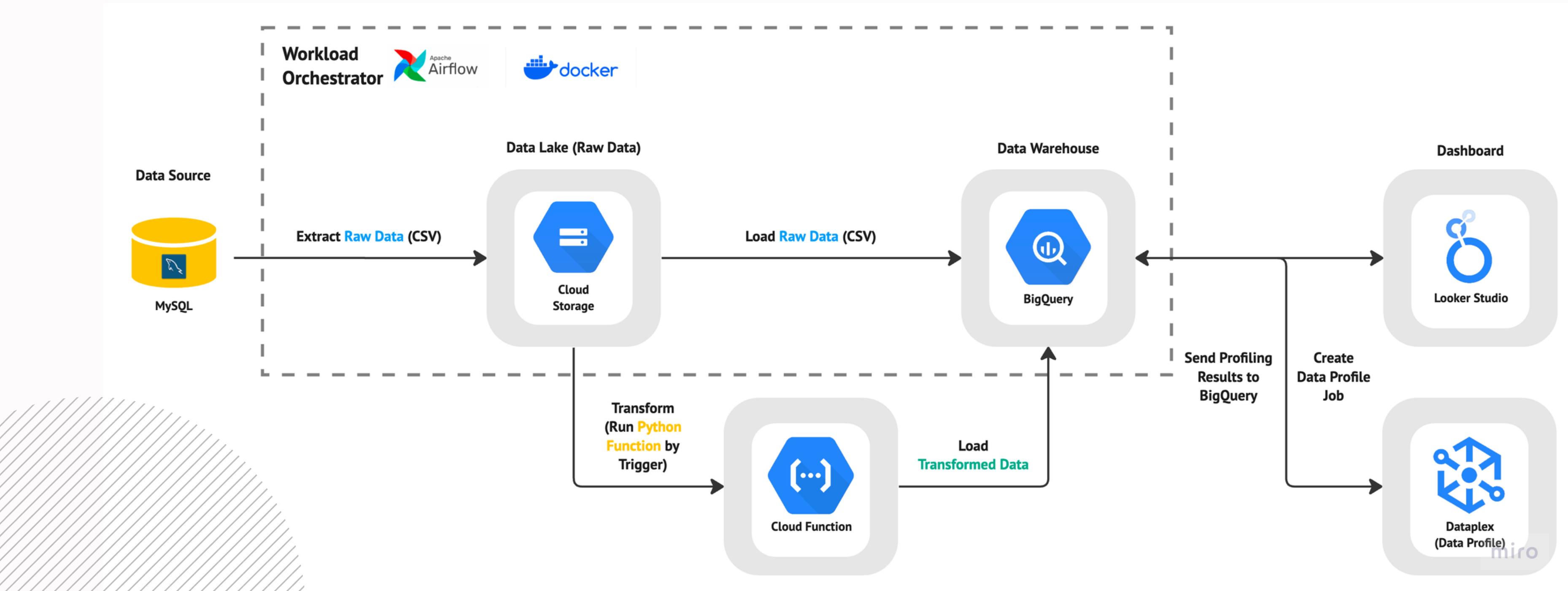


Problems

Credivo, like many modern companies, deals with a lot of data from different sources. This data can help make smart decisions and find important insights, but it's hard to use effectively.

To fix this, Credivo knows it needs a better data pipeline for managing data. This pipeline will organize the data better, making it easier to use and trust. With this pipeline, Credivo wants to give its team the right information at the right time, so they can make better decisions and make the company better overall.

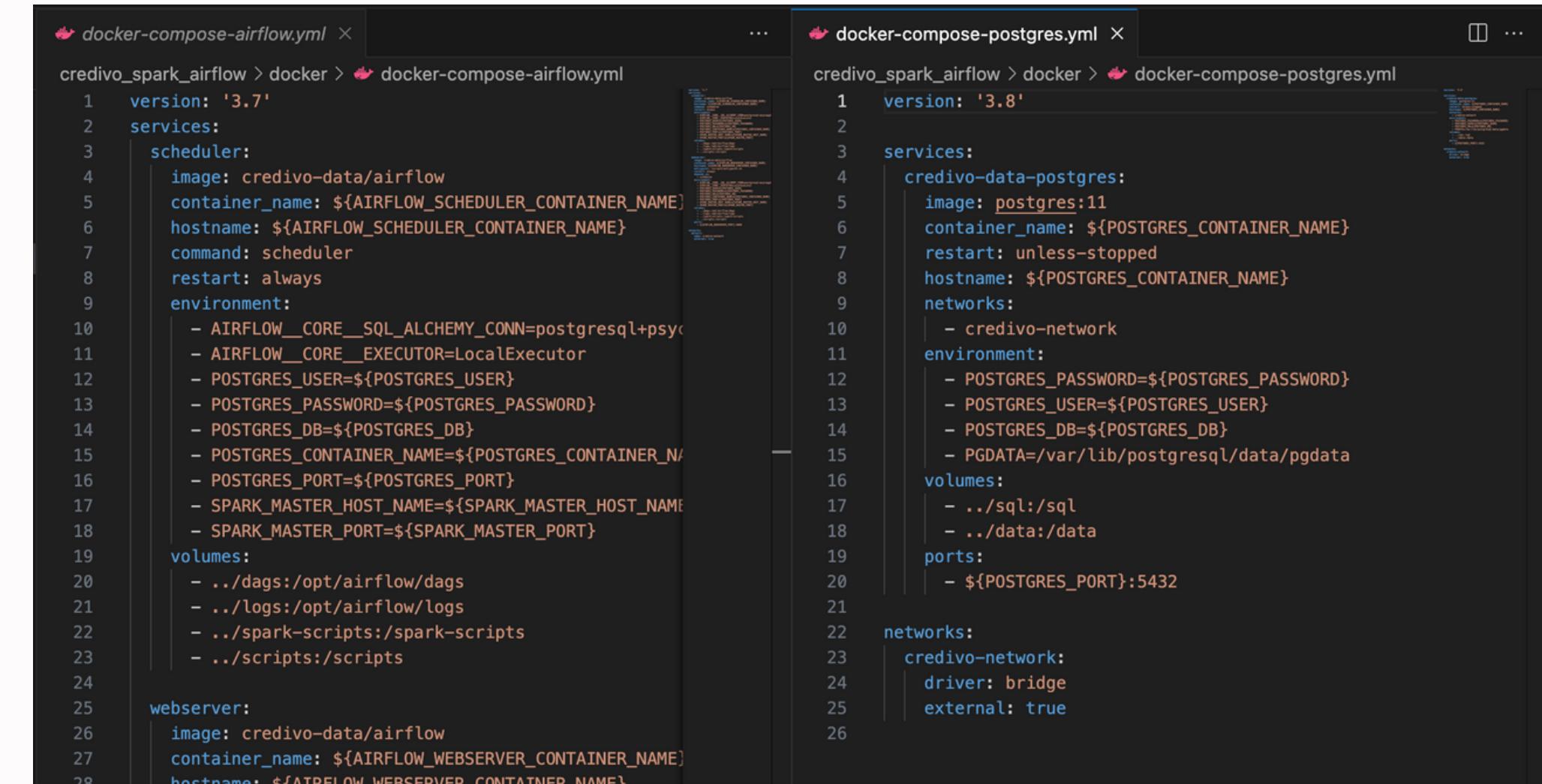
Data Pipeline



Requirement

1. Docker Compose Configuration:

- File:
 - **docker-compose-airflow.yml**
 - **docker-compose-postgres.yml**
 - **docker-compose-jupyter.yml**
- Services:
 - Airflow Scheduler
 - Airflow Webserver
 - Postgres
 - Jupyter



The image shows a code editor with two tabs open, each displaying a Docker Compose configuration file.

File Structure:

- Left tab: `docker-compose-airflow.yml` (version 3.7)
- Right tab: `docker-compose-postgres.yml` (version 3.8)

Content of docker-compose-airflow.yml:

```
1 version: '3.7'
2 services:
3   scheduler:
4     image: credivo-data/airflow
5     container_name: ${AIRFLOW_SCHEDULER_CONTAINER_NAME}
6     hostname: ${AIRFLOW_SCHEDULER_CONTAINER_NAME}
7     command: scheduler
8     restart: always
9     environment:
10    - AIRFLOW__CORE__SQLALCHEMY_CONN=postgresql+psycopg2://airflow:airflow@postgres/airflow
11    - AIRFLOW__CORE__EXECUTOR=LocalExecutor
12    - POSTGRES_USER=${POSTGRES_USER}
13    - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
14    - POSTGRES_DB=${POSTGRES_DB}
15    - POSTGRES_CONTAINER_NAME=${POSTGRES_CONTAINER_NAME}
16    - POSTGRES_PORT=${POSTGRES_PORT}
17    - SPARK_MASTER_HOST_NAME=${SPARK_MASTER_HOST_NAME}
18    - SPARK_MASTER_PORT=${SPARK_MASTER_PORT}
19   volumes:
20    - ./dags:/opt/airflow/dags
21    - ./logs:/opt/airflow/logs
22    - ./spark-scripts:/spark-scripts
23    - ./scripts:/scripts
24
25   webserver:
26     image: credivo-data/airflow
27     container_name: ${AIRFLOW_WEBSERVER_CONTAINER_NAME}
28     hostname: ${AIRFLOW_WEBSERVER_CONTAINER_NAME}
```

Content of docker-compose-postgres.yml:

```
1 version: '3.8'
2 services:
3   credivo-data-postgres:
4     image: postgres:11
5     container_name: ${POSTGRES_CONTAINER_NAME}
6     restart: unless-stopped
7     hostname: ${POSTGRES_CONTAINER_NAME}
8     networks:
9      - credivo-network
10     environment:
11       - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
12       - POSTGRES_USER=${POSTGRES_USER}
13       - POSTGRES_DB=${POSTGRES_DB}
14       - PGDATA=/var/lib/postgresql/data/pgdata
15     volumes:
16       - ./sql:/sql
17       - ./data:/data
18     ports:
19       - ${POSTGRES_PORT}:5432
20
21   networks:
22     credivo-network:
23       driver: bridge
24       external: true
```

Requirement

2. Airflow DAG (Directed Acyclic Graph):

- File:
 - [run]credivo-dag-gcs.py
- Steps:
 - Define an Airflow DAG named **move_mysql_to_gbg** for loading raw data from **MySQL to Google Cloud Storage (Raw Data)** and **BigQuery (Raw Data)**
 - Import Variable and define variable in DAG file such as **BUCKET_NAME**, **SQL_QUERY**, **FILENAME**, **GCP_CONN_ID**, **MYSQL_CONN_ID**, and **DESTINATION_DATASET_TABLE**
 - Use **MySQLToGCSEoperator** to load raw data from **MySQL to Google Cloud Storage**
 - Use **GCSToBigQueryOperator** to load raw data from **Google Cloud Storage to BigQuery**
 - Schedules the DAG to run daily and set dependecies

```
[run]credivo-dag-gcs.py 4, M X
credivo_spark_airflow > dags > [run]credivo-dag-gcs.py > ...
3 import os
4 from airflow.providers.google.cloud.transfers.mysql_to_gcs import MySQLToGCSEoperator
5 from airflow.providers.google.cloud.transfers.gcs_to_bigquery import GCSToBigQueryoperator
6 from airflow.models import Variable
7
8 #dag arguments
9 default_args = {
10     'owner': 'ariq',
11     'start_date': datetime[2024, 1, 1],
12     'retries': 1,
13     'retry_delay': timedelta(minutes=5)
14 }
15
16 #define dag
17 dag = DAG('move_mysql_to_gbg',
18            schedule_interval= "@daily",
19            default_args= default_args,
20            description="Moving Data from MySQL to BigQuery",
21            dagrun_timeout=timedelta(minutes=60)
22 )
23
24 BUCKET_NAME = Variable.get("bucket_name")
25 SQL_QUERY = Variable.get("simple_query_all")
26 FILENAME = Variable.get("gcs_file")
27 GCP_CONN_ID = Variable.get("gcp_conn_id")
28 MYSQL_CONN_ID = Variable.get("mysql_conn_id")
29 DESTINATION_DATASET_TABLE = Variable.get("destination_dataset_table")
```

The screenshot shows the Airflow web interface with the following details:

- Header:** Airflow, DAGs, Cluster Activity, Datasets, Security, Browse, Admin, Docs.
- DAGs Section:**
 - Buttons: All (1), Active (1), Paused (0), Running (16), Failed (0).
 - Table Headers: DAG, Owner, Runs, Schedule.
 - Table Data:

DAG	Owner	Runs	Schedule
move_mysql_to_gbg	ariq	8 (green) 16 (green)	@daily

Requirement

3. Environment:

- File:
 - **.env**

4. Dockerfile:

- File:
 - **Dockerfile.airflow-arm**
 - **Dockerfile.jupyter**

• Descriptions:

- Utilizes the Apache Airflow version **2.7.1**
- Installs dependencies

5. Makefile:

- File:
 - **Makefile**
- Commands:
 - run-all: Starts all the containers
 - stop-all: Stops all the containers

```
credivo_spark_airflow > .env
1 POSTGRES_USER=credivo
2 POSTGRES_PASSWORD=credivo
3 POSTGRES_DB=postgres_db
4 POSTGRES_PORT=5432
5 POSTGRES_CONTAINER_NAME=credivo-postgres
6
7 AIRFLOW_SCHEDULER_CONTAINER_NAME=credivo-airflow-scheduler
8 AIRFLOW_WEBSERVER_CONTAINER_NAME=credivo-airflow-webserver
9 AIRFLOW_WEBSERVER_PORT=8080
10
11 JUPYTER_CONTAINER_NAME=credivo-jupyter
12 JUPYTER_PORT=9999
13
```

```
Dockerfile.airflow-arm
FROM apache/airflow:2.7.1-python3.9@sha256:faddcd177ae17f604f587aa8761bbd398685e00d379a2da7eab1c4ca
USER root
# Install OpenJDK-11
RUN apt update && \
apt-get install -y openjdk-11-jdk && \
apt-get install -y ant && \
apt-get install -y procps && \
apt-get clean;
# Set JAVA_HOME
ENV JAVA_HOME /usr/lib/jvm/java-11-openjdk-arm64
RUN export JAVA_HOME
USER airflow
RUN pip install \
lxml \
pyspark==3.3.2 \
apache-airflow-providers-apache-spark \
requests \
pandas
COPY --chown=airflow:root ./dags /opt/airflow/dags
```

```
Makefile
1 include .env
2 .PHONY: run-all
3
4 run-all: postgres airflow
5
6 .PHONY: stop-all
7
8 stop-all:
9     @docker-compose -f ./docker/docker-compose-airflow.yml stop
10    @docker-compose -f ./docker/docker-compose-postgres.yml stop
11
12 docker-build-arm:
13     @echo '_____
14     @echo 'Building Docker Images ...'
15     @echo '_____
16     @docker network inspect credivo-network >/dev/null 2>&1 || docker network create credivo-network
17     @echo '_____
18     @docker build -t credivo-data/spark -f ./docker/Dockerfile.spark .
19     @echo '_____
20     @docker build -t credivo-data/airflow -f ./docker/Dockerfile.airflow-arm .
21     @echo '_____
22     @docker build -t credivo-data/jupyter -f ./docker/Dockerfile.jupyter .
23     @echo '=====
```

Requirement

6. Google Cloud Bucket

- Description:
 - Create and define bucket name as crdivo_lake
 - Location type : Region
 - Location : asia-southeast2 (Jakarta)
 - Storage Class : Standard (We consider to use standard because of the usage activity is actively used)

7. BigQuery Dataset

- Description:
 - Create New dataset and define name as CREDIVO_DW to store Raw Data and Transformed Data
 - Data location : asia-southeast2 (Jakarta)

The screenshot shows the 'Bucket details' page for a Google Cloud Storage bucket named 'crdivo_lake'. The 'CONFIGURATION' tab is selected. Key configuration details include:

- Location:** asia-southeast2 (Jakarta)
- Storage class:** Standard
- Public access:** Not public
- Protection:** None

The 'OVERVIEW' section provides the following information:

- Created:** January 29, 2024 at 3:01:20 PM GMT+7
- Updated:** February 2, 2024 at 2:32:40 PM GMT+7
- Location type:** Region
- Location:** asia-southeast2 (Jakarta)
- Replication:** –
- Default storage class:** Standard
- Requester Pays:** OFF
- Tags:** None
- Labels:** None
- Cloud Console URL:** https://console.cloud.google.com/storage/browser/crdivo_lake
- gsutil URI:** gs://crdivo_lake

The 'PERMISSIONS' section shows:

- Access control:** Uniform
- Public access prevention:** Enabled via bucket setting
- Public access status:** Not public

The 'PROTECTION' section shows:

- Object versioning:** Off
- Bucket retention policy:** None
- Object retention:** Disabled

CREDIVO_DW

Dataset info

Dataset ID	corporate-digital.CREDIVO_DW
Created	Jan 22, 2024, 4:41:44 PM UTC+7
Default table expiration	Never
Last modified	Jan 22, 2024, 4:41:44 PM UTC+7
Data location	asia-southeast2
Description	
Default collation	
Default rounding mode	ROUNDING_MODE_UNSPECIFIED
Time travel window	7 days
Storage billing model	LOGICAL
Case insensitive	false
Labels	
Tags	

Requirement

8. Cloud Function

- Description:
 - Create new function named `credivo_transform` to transform data that stored in Google Cloud Storage to BigQuery
 - Region : `asia-southeast2`
 - Memory allocated : `2 GiB`
 - CPU : `1`
 - Timeout : `60 seconds`
 - Minimum instances : `0`
 - Maximum instances: `100`

9. Service Account for Credentials

- Description:
 - Create and define service account name as `local-airflow-mysql-gcs` to generate google cloud credential key

The screenshot shows the 'Function details' page for a 2nd gen function named 'credivo_transform'. The function was deployed at Feb 3, 2024, 3:36:24 PM. The URL is https://asia-southeast2-corporate-digital.cloudfunctions.net/credivo_transform. The 'DETAILS' tab is selected, showing the following configuration:

Setting	Value
Last deployed	February 3, 2024 at 3:36:24 PM GMT+7
Region	asia-southeast2
Memory allocated	2 GiB
CPU	1
Timeout	60 seconds
Minimum instances	0
Maximum instances	100
Concurrency	1
Service account	compute@developer.gserviceaccount.com
Build Worker Pools	—
Container build log	37af18a7-aa3f-4bb9-8648-478fc8bff7ec

The 'Networking Settings' section shows 'Ingress settings' set to 'Allow all traffic'.

The screenshot shows the 'local-airflow-mysql-gcs' service account's 'KEYS' tab. The 'KEYS' tab is selected, showing a warning message: "Service account keys could pose a security risk if compromised. We recommend using a public key certificate instead. Learn more about authenticating service accounts on Google Cloud [here](#)". Below the message, there is a button to "Add Key". A table lists the existing key:

Type	Status	Key
Google Cloud Platform API Key	Active	0b6117488a7b8b62b4be3c249e6ad5db1bd604f5

Requirement

10. Data Profiling Scan on Google Cloud

Dataplex:

- Description:
 - Create and define data profile name as **DP - CRDV APPLICATION TEST**
 - Set profiling scope to **Entire Data** and sampling size to **All Data**
 - Send the profile test to BigQuery Dataset named CREDIVO_DW and name it as DP_{Table_Name} (In this case, we use application_test data to run the whole process)
 - Set repeat check on Daily

Data Profiling [+ CREATE DATA PROFILE SCAN](#) [+ CREATE MULTIPLE PROFILE SCANS](#) [REFRESH](#)

Analyze the profile of your Dataplex managed data by configuring and scheduling checks over your data.

Filter	Display name	Last run	Labels	Table name	Incremental column	Schedule	⋮
<input type="button" value="DP - CRDVO"/>	DP - CRDVO APPLICATION TEST PROFILE	18 hours ago	None	raw_application_test		Every day at 12:00 AM Etc/GMT+8	<input type="button"/>

Output

Buckets > crdovo_lake > raw_application_test

LIVE OBJECT

VERSION HISTORY

DOWNLOAD **EDIT METADATA** **EDIT ACCESS** **DELETE**

Overview

Type	text/csv
Size	25.4 MB
Created	Feb 3, 2024, 5:07:38 PM
Last modified	Feb 3, 2024, 5:07:38 PM
Storage class	Standard
Custom time	—
Public URL	Not applicable
Authenticated URL	https://storage.cloud.google.com/crdovo_lake/raw_application_test
gsutil URI	gs://crdovo_lake/raw_application_test

Permissions

Public access: Not public

Protection

Version history: —

Retention expiration time: None

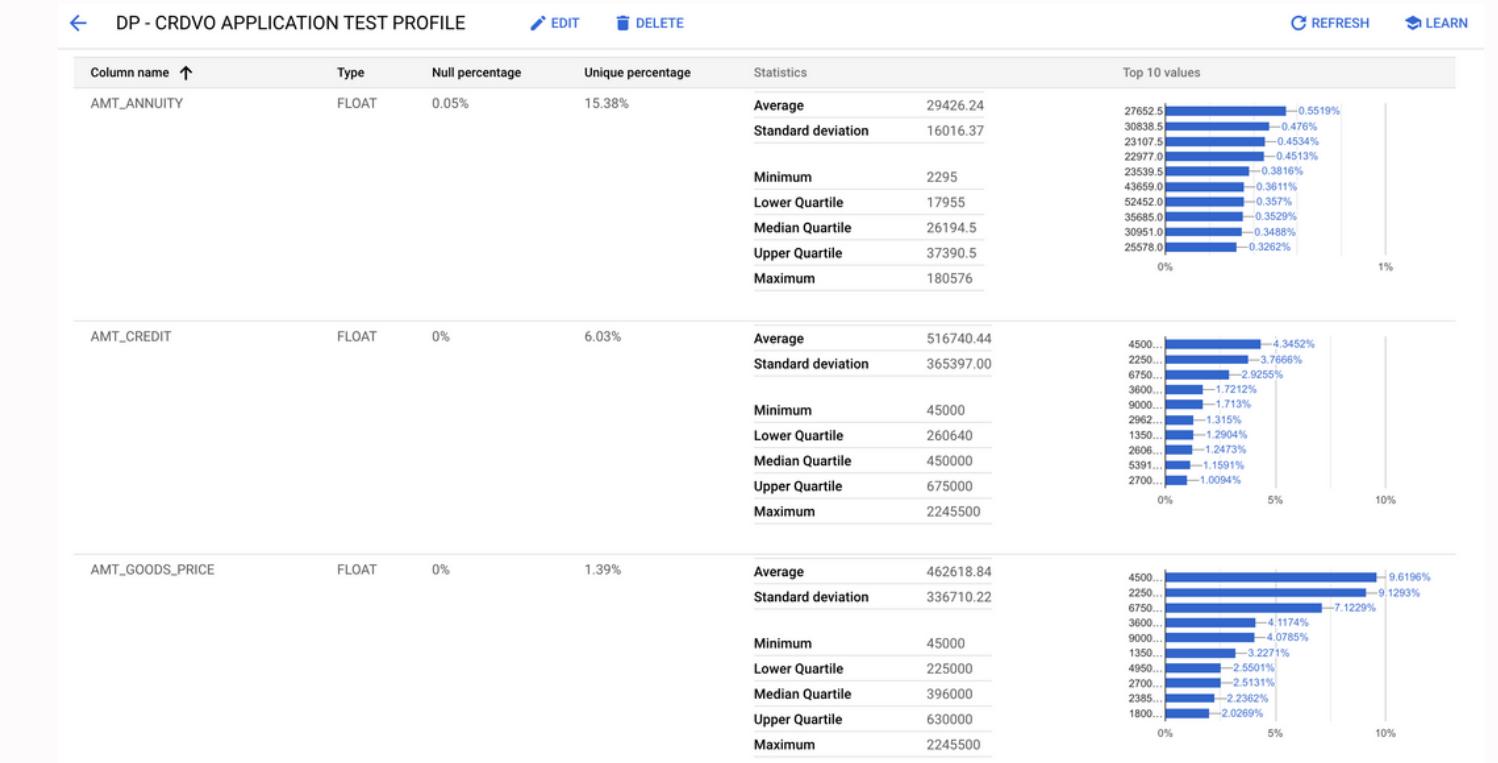
Object retention retain until time: None

Bucket retention retain until time: None

Hold status: None

Encryption type: Google-managed

Data Lake in Google Cloud Storage



Data Profiling in Dataplex

Explorer

application_test_clean

SCHEMA

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
SK_ID_CURR	INTEGER	NULLABLE	-	-	-	-	-
NAME_CONTRACT_TYPE	STRING	NULLABLE	-	-	-	-	-
CODE_GENDER	STRING	NULLABLE	-	-	-	-	-
FLAG_OWN_CAR	STRING	NULLABLE	-	-	-	-	-
FLAG_OWN_REALTY	STRING	NULLABLE	-	-	-	-	-
CNT_CHILDREN	INTEGER	NULLABLE	-	-	-	-	-
AMT_INCOME_TOTAL	FLOAT	NULLABLE	-	-	-	-	-
AMT_CREDIT	FLOAT	NULLABLE	-	-	-	-	-
AMT_ANNUITY	FLOAT	NULLABLE	-	-	-	-	-
AMT_GOODS_PRICE	FLOAT	NULLABLE	-	-	-	-	-
NAME_TYPE_SUITE	STRING	NULLABLE	-	-	-	-	-
NAME_INCOME_TYPE	STRING	NULLABLE	-	-	-	-	-
NAME_EDUCATION_TYPE	STRING	NULLABLE	-	-	-	-	-
NAME_FAMILY_STATUS	STRING	NULLABLE	-	-	-	-	-
NAME_HOUSING_TYPE	STRING	NULLABLE	-	-	-	-	-

SUMMARY

application_test_clean

corporate-digital.CREDIVO_DW

Last modified: Feb 3, 2024, 5:18:44PM UTC+7

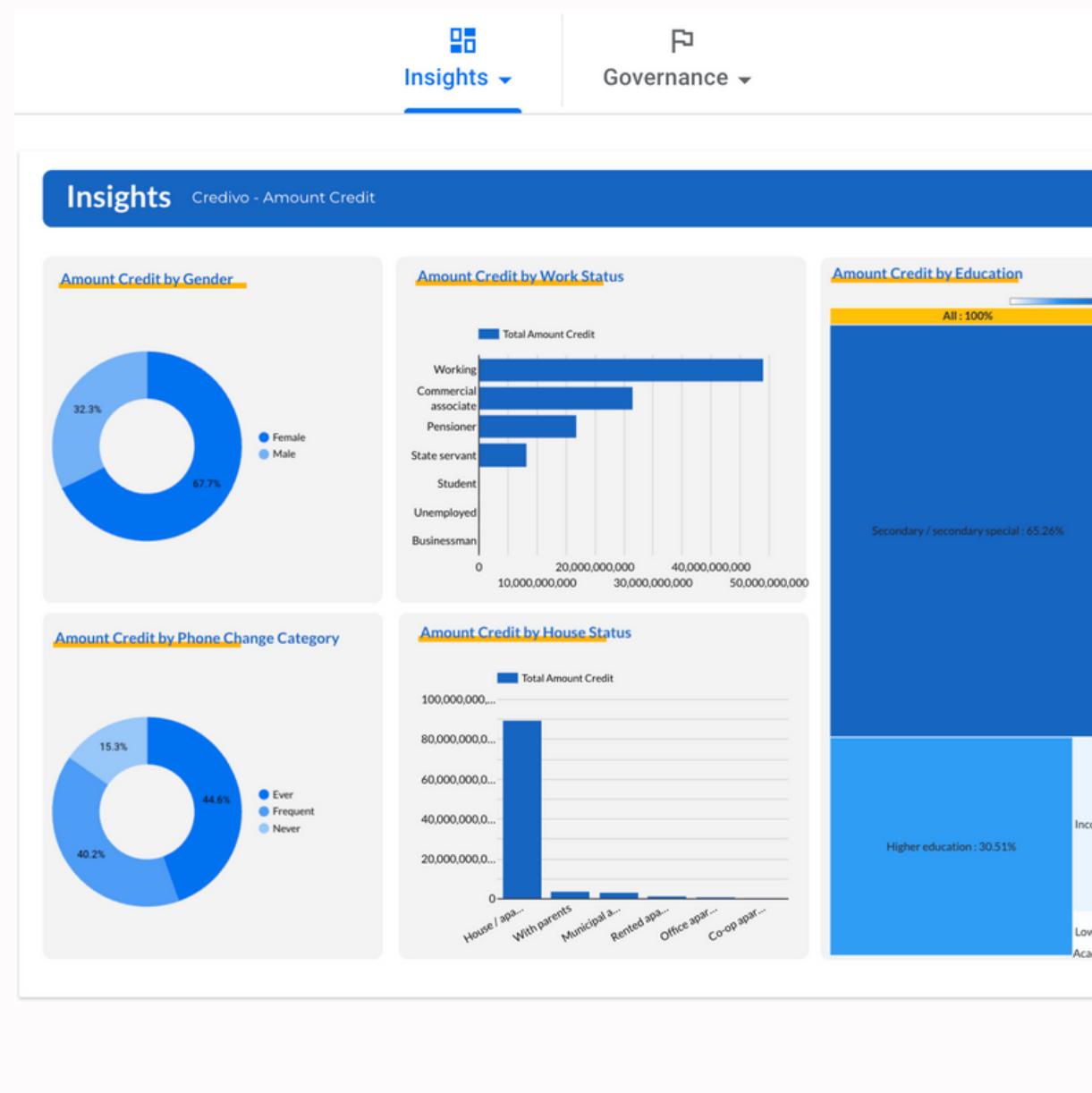
Data location: asia-southeast2

VIEW ROW ACCESS POLICIES

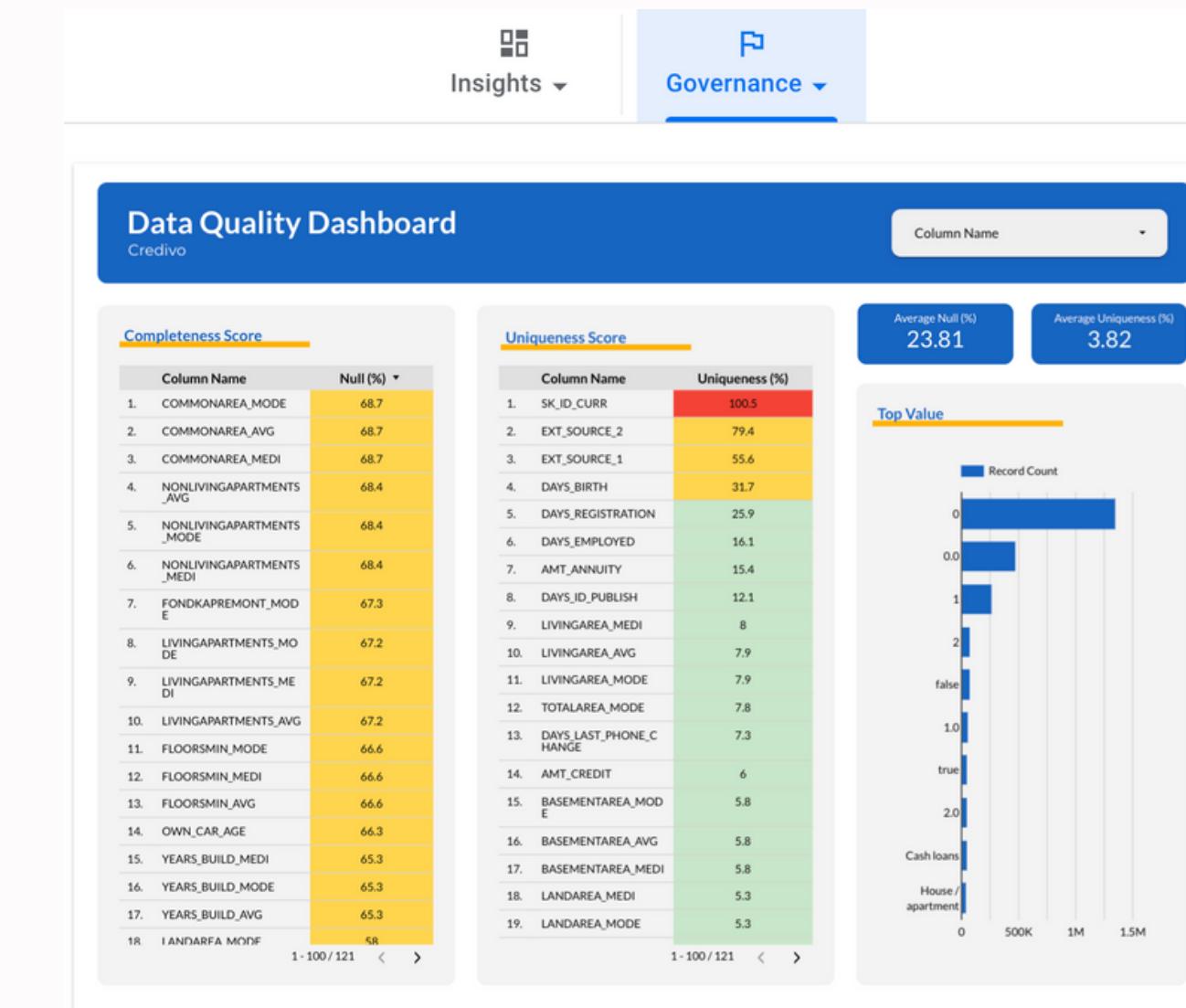
Raw and Clean Data in BigQuery

Output

Insight Dashboard



Data Quality Dashboard



The background image shows a modern office space with a high ceiling featuring exposed pipes and ductwork. Large windows on the left provide natural light. A central atrium is filled with various types of green plants and trees. The floor is made of light-colored wood. In the foreground, there are several desks with black office chairs, and a large sofa area in the background.

THANK YOU