Assignment for Data Engineer

Problem Statement: Real-Time Cryptocurrency Data Pipeline Using GCP and Python

A digital marketing agency wants to monitor and analyze real-time cryptocurrency data to identify market trends and support data-driven strategy decisions. The goal is to design an automated data pipeline that extracts pricing data from the dynamic website https://www.investing.com/crypto/ for top 10 crypto currencies, processes it, and makes it available for analysis and visualization through a dashboard.

As a data engineer, your task is to build a real-time data pipeline using Google Cloud Platform (GCP) and Python. The pipeline should begin with a Selenium script that scrapes key cryptocurrency data—including name, symbol, current price, % change, volume, and market cap—from the specified website. This scraping process should run at hourly intervals.

Once the data is scraped, process and clean it using Pandas and NumPy. This includes handling missing values, converting data types, and calculating additional metrics such as percentage change, z-scores, and rolling averages.

After processing, the data should be published to a Google Cloud Pub/Sub topic. A Cloud Function, triggered by that topic, should parse the incoming data and store it in a BigQuery table. To automate the scraping process, set up a Cloud Scheduler job to trigger the Selenium script at regular intervals (every 60 minutes).

Finally, build a Looker Studio dashboard connected to the BigQuery table. The dashboard should include filters such as cryptocurrency name/symbol, time range, and percentage change thresholds, along with visualizations like time-series charts, key performance indicators, and a dynamic summary table to highlight market movements.

o Code repository link : https://github.com/Mohdmaazekram/Cryptocurrency_Assigement o BigQuery dataset and table link (with appropriate access) : https://console.cloud.google.com/bigquery?ws=!1m5!1m4!4m3!1smaazgsheet!2scrypto_currencies!3scrypto_currencies_data o Looker Studio dashboard link : https://lookerstudio.google.com/u/1/reporting/0204258c-2b79-4cda-ac8e-252ebc23627d/page/p_x57mqzj6sd

==

Overview

- **This project implements an automated data pipeline that scrapes real-time cryptocurrency data from Investing.com, processes it using Python, and pushes the cleaned data into BigQuery on Google Cloud Platform (GCP). The pipeline is scheduled to run every hour using Cloud Run Jobs and Cloud Scheduler.**

Tools & Technologies Used In This Project

- **Python with Selenium, Pandas, NumPy, BigQuery and Other libraries.**
- **I Used Selenium WebDriver for dynamic website scraping.**
- **Google Cloud Platform (GCP) and its Services is Use For Automate Data Pipeline that Scrapes Real-Time Data.**
  - Cloud Shell
  - Artifact Registry
  - Cloud Run (Job)
  - Cloud Scheduler
  - BigQuery

Step-by-Step Implementation

1) Upload and Test Code in Cloud Shell

- *Open Cloud Shell in GCP Console.*
- *Create a working directory and upload necessary files Like:*

  Uploading files from my local system into this directory using cloud shell terminal:
  - Dockerfile
  - requirements.txt
  - scraper.py
- *Test Python Script:*
  - Command to Run or Test Python File : python3 scraper.py in cloud shell terminal.
  - Also Install Python Library : pip install -r requirements.txt in cloud shell terminal.

2) Build Docker Image and Push to Artifact Registry

- *Code For Create a Docker repository:*
  - gcloud artifacts repositories create my-repo --repository-format=docker --location=us-central1 --description="Scraper Image Repo"
- *Build and push Docker image in Cloud Shell Terminal:*
  - gcloud builds submit --tag us-central1-docker.pkg.dev/Project_ID/Repositry_name/image_name
  - For Eg: gcloud builds submit --tag us-central1-docker.pkg.dev/local-axis-461918-j2/my-repo/scraper

3) Create Cloud Run Job

```
* First Go to Job Run.
* Second Click on Jobs and Create it.
* Third Select container image from Artifact Registry.
For Eg: my-repo (Repositry_name)
            Scraper (image_name)
                486c99e12b
* execute the job. from the UI button or from cloud shell.
```

4) Grant BigQuery Permissions to the Job

```
SA_EMAIL=$(gcloud run jobs describe scraper --region=us-central1 --
format="value(template.template.serviceAccount)")

gcloud projects add-iam-policy-binding [Project_ID] (local-axis-461918-j2) \
  --member="serviceAccount:$SA_EMAIL" \
  --role="roles/bigquery.dataEditor"

gcloud projects add-iam-policy-binding [Project_ID] (local-axis-461918-j2) \
  --member="serviceAccount:$SA_EMAIL" \
  --role="roles/bigquery.metadataViewer"
```

5) Schedule Cloud Run Job Every 1 Hour with Cloud Scheduler

```
* First Go to Job Run.
* Second Click on Job Next Click on [Container] scraper
* Then Click on Trigger to set according to your condition.
```

6) Updating the Python Code and Redeploying

```
* 1. Edit scraper.py
        nano scraper.py
* 2. Rebuild and push the image to Artifact Registry
        gcloud builds submit --tag us-central1-docker.pkg.dev/maazgsheet/my-repo/scraper

* 3. Update the Cloud Run Job with the new image
        gcloud run jobs update scraper \
          --image=us-central1-docker.pkg.dev/maazgsheet/my-repo/scraper \
          --region=us-central1

* 4. Execute the updated Cloud Run Job
        gcloud run jobs execute scraper --region=us-central1
```

Quick Fix: Limit the Build Context

```
    mkdir scraper-job
    cp scraper.py Dockerfile requirements.txt scraper-job/
    cd scraper-job
    gcloud builds submit --tag us-central1-docker.pkg.dev/local-axis-461918-j2/my-repo/scraper
```

==

Summary of Files Used

- scraper.py : Python script for scraping and pushing to BigQuery

- requirements.txt : Python dependencies
- Dockerfile : Docker image setup for Cloud Run Job

Final Notes

- Cloud Scheduler triggers the Cloud Run Job scraper every hour.
- The job scrapes the top 10 crypto data from Investing.com and appends it to BigQuery.
- BigQuery table used: maazgsheet.crypto_currencies.crypto_currencies_data