**Setup**

1. **Kafka**
   * **Add Network Tags**
     + Add network tags to apply firewall rule to specific VM (Ex: kafkavm)
   * **Firewall Rule (Restart VM for Rule to Apply):**
     + VPC Network -> Firewall -> Create Firewall Rule
     + Name: “kafkaportvm”
     + Network: “default”
     + Priority: 1
     + Direction of Traffic: Ingress
     + Targets: Specific Target Tags -> Input target tags (Ex: kafkavm)
     + Source filter: IPV4 ranges, 0.0.0.0/0
     + Protocols and Ports: TCP 9092
   * **Promote Ephemeral External IP to Static IP for Kafka VM**
     + Tutorial: <https://cloud.google.com/compute/docs/ip-addresses/reserve-static-external-ip-address#promote_ephemeral_ip>
   * **Docker + Docker Compose**
     + Install on VM (Debian): <https://docs.docker.com/engine/install/debian/>
   * **Python + PIP**
     + Install on VM (Debian): sudo apt-get -y install python3-pip
     + Check PIP version: pip --version
     + Install Requirements: pip install -r requirements.txt
   * **Files needed here:** “docker-compose.yml”, “requirements.txt”, “producer\_reddit\_comments.py”, “producer\_reddit\_submissions.py”
   * **Create Environmental Variables Files**
     + Create .env file: nano .env
     + Enter variables: KAFKA\_ADDRESS=external IP of VM
   * **Confluent Kafka**
     + Dockerhub: <https://hub.docker.com/r/confluentinc/cp-kafka>
     + Source for Docker Compose File: <https://raw.githubusercontent.com/confluentinc/cp-all-in-one/7.2.0-post/cp-all-in-one/docker-compose.yml>
     + Check Docker Version in VM: docker --version
     + **Docker compose in VM does not have “-“**: docker compose version
     + Docker Compose
       1. Start: docker-compose --env-file .env up
       2. Stop: docker-compose down
     + URL for Accessing Confluent Control Center: <http://localhost:9021/clusters>
   * **Run in Detach Mode using Screen**
     + **Install Screen:**
       1. sudo apt-get update
       2. sudo apt-get -y install screen
     + **Check Screen Version:** screen --version
     + **Run Screen:** Screen -S (screen name) (command)
       1. **Command:** screen -s kafka sudo docker compose --env-file .env up
     + **Detach from Screen:** CTRL + A, D
     + **List all screens:** screen -ls
     + **Reattach screens:** screen -r (screen name)
     + **Kill Particular Screens:** screen -x -s (screen name) quit
     + **Kill All Screens:** pkill screen
     + **Source:** <https://medium.com/codebase/how-to-keep-multiple-linux-terminals-running-in-background-screen-ccf2e53b0d22>
   * **GCP VM Ports**
     + **Connect to Port (through Cloud Shell) to Preview Confluent UI:**
       1. Connect to instance: gcloud compute ssh kafkavm --zone=us-central1-c -- -L 9021:localhost:9021
       2. Click on Web Preview -> Change Port to 9021 -> Preview
       3. **Graphical user interface

          Description automatically generated**Source: <https://stackoverflow.com/questions/63098587/accessing-localhost-of-gcp-instance-from-local-machine>
   * **Run Producers**
     + python3 producer\_reddit\_comments.py --KAFKA\_ADDRESS=external IP of kafka VM
     + python3 producer\_reddit\_submissions.py --KAFKA\_ADDRESS=external IP of kafka VM
   * **Run Producers using Screen**
     + screen -S comments python3 producer\_reddit\_comments.py --KAFKA\_ADDRESS=external IP of kafka VM
     + screen -S submissions python3 producer\_reddit\_submissions.py --KAFKA\_ADDRESS=external IP of kafka VM
2. **Spark**
   * **Dataproc**
     + **JAR Files Source:** <https://mvnrepository.com/artifact/org.apache.spark/spark-sql-kafka-0-10_2.12/3.1.3>
     + Create Dataproc Spark Cluster with Spark 3.13
     + I just did 1 Master Node and no workers
   * **Submit PySpark Job**
     + Need to upload the “sparkStreaming.py” file to sparkclusterbucket
     + gcloud dataproc jobs submit pyspark gs://sparkclusterbucket/sparkStreaming.py --cluster sparkclustervm --properties spark.jars.packages=org.apache.spark:spark-sql-kafka-0-10\_2.12:3.1.3 --region us-central1 -- --KAFKA\_ADDRESS=external IP of kafka VM
3. **Cloud Functions (Detect On Create of New File in Bucket)**
   * Make sure (fedex\_reddit\_dataset) exists, and is clear of the tables before starting
   * Tutorial Example: <https://cloud.google.com/functions/docs/tutorials/storage>
   * Creating Functions
     + Function Name: loadParquetComments
     + Region: us-central-1
     + Trigger -> Cloud Storage -> On (finalizing/Creating) file in storage bucket
     + Select Desired Bucket
     + Memory: 512 MB, Max Instances = 1
     + Code:
       1. Runtime: Python 3.9
       2. Requirements: Put the Bigquery Requirements
       3. Copy paste the “function\_comments.py” or “function\_submissions.py” file
     + Deploy
     + Test and try uploading dummy file “parquet.txt” to check if function is working properly
4. **Terraform**
   * Tutorial: <https://cloud.google.com/docs/terraform/get-started-with-terraform>
   * Reset Terraform State = Can delete the terraform state files
   * Can utilize Cloud Shell to run Terraform Commands directly
   * **Execution Steps**
     + terraform init = Initalize & install
     + terraform plan = Match against previous state
     + terraform apply = Apply changes to the cloud
     + terraform destroy = Remove stack from cloud (everything)
     + terraform destroy -target RESOURCE\_TYPE.NAME = Destroy specific resource
   * **Text

     Description automatically generatedTerraform destroy except resources**
   * **After Running Terraform**
     + Still need to install dependencies (Docker, PIP, etc.) as specified above, and upload the .yml and .py files for docker to VM

**Dashboard Link**: <https://datastudio.google.com/reporting/afbed74b-8f18-4012-bc91-ddbc66e4574c>

**Architecture Diagram**

Data lake

Data Warehouse

consumer

producer

**PRAW Python File**

**Data Warehouse**

Table

Description automatically generatedSubmissions

Comments

Table

Description automatically generated