UNIVERSIDAD
PANAMERICANA

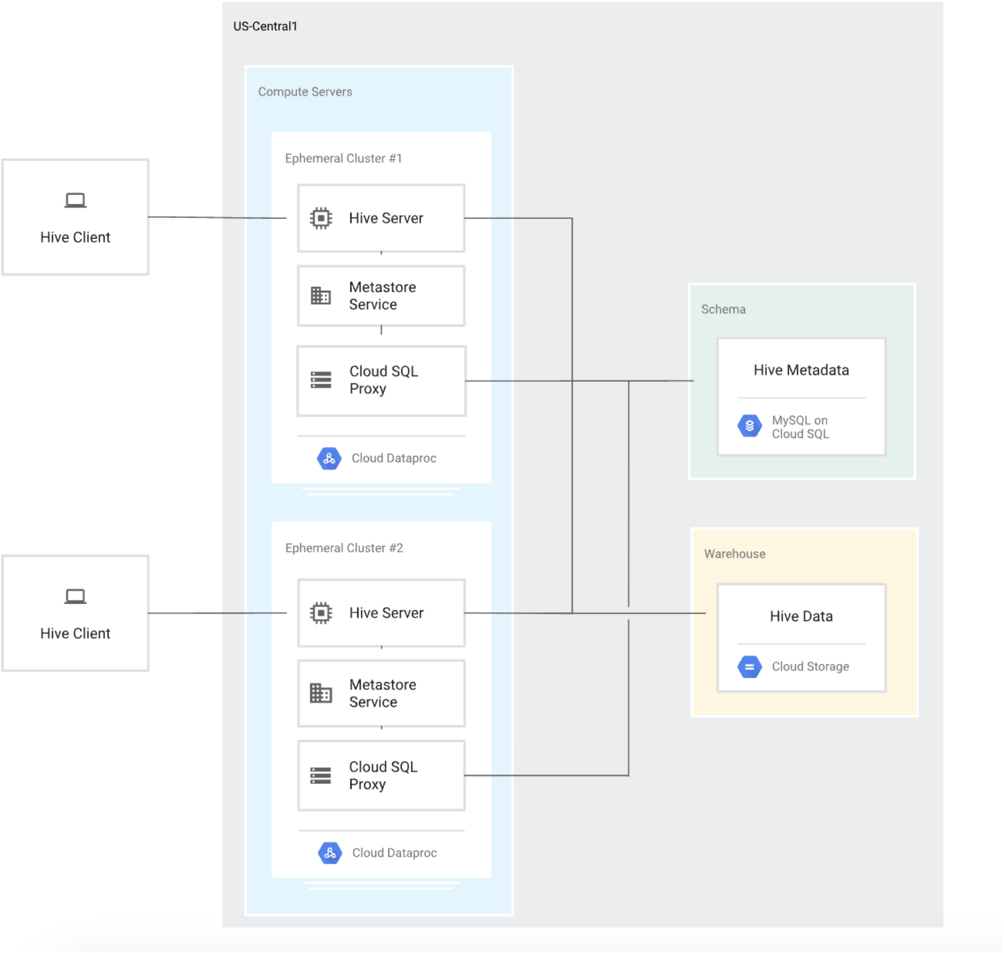
Hadoop Cluster with Hive

Andrés Gregori Altamirano

April 2024

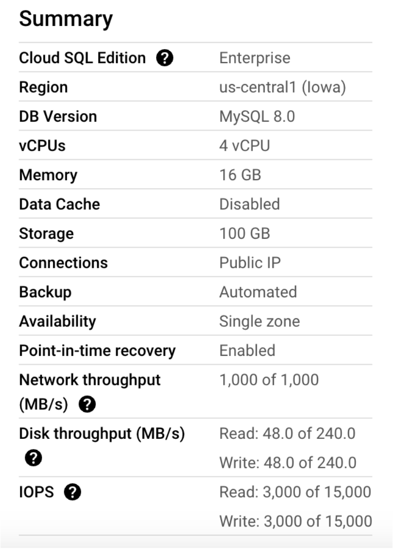
Abstract

During this project, you'll create a Hadoop cluster with a Hive Database in which you'll add your preferred dataset and make an EDA. Your Hive Metastore will be stored in a MySQL instance on Cloud SQL. And your Hive Warehouse will be stored on Cloud Storage. Also, you'll create a document explaining your development process and the possible setbacks you may find on your project. Even though there is a procedure written on this document, you may find your own way into the project and it will be accepted. The following diagram represents having 2 clusters that can communicate through Hive, however, we will focus on creating only one cluster.



Procedure
Note:Be sure to take screenshots of the entire process for evidence.

1. Download the Google Cloud SDK. More info on this link
2. Create a Cloud Storage bucket:
   - Open the console and select Cloud Storage
   - Create a bucket. Note: It is preferable to set the bucket on the same Compute Engine Region you'll use. US-South or US-Central are recommended to avoid latency.
   - Try to tag it so you remember that it will be for the Hive data.
   - Create a new bucket with the initialization actions. Use the same config. for the first bucket.
   - Download the following doc: cloud-proxy
   - Upload it into the new bucket.
3. Create a Cloud SQL Instance.
   - Note: The following characteristics are highly recommended. Cloud SQL is an expensive product within GCP. Please consider this steps.
   - Open Cloud SQL and select Create Instance
   - Select MySQL
   - Configure it to be as the following:



Summary

Cloud SQL Edition	Enterprise
Region	us-central1 (Iowa)
DB Version	MySQL 8.0
vCPUs	4 vCPU
Memory	16 GB
Data Cache	Disabled
Storage	100 GB
Connections	Public IP
Backup	Automated
Availability	Single zone
Point-in-time recovery	Enabled
Network throughput (MB/s)	1,000 of 1,000
Disk throughput (MB/s)	Read: 48.0 of 240.0 Write: 48.0 of 240.0
IOPS	Read: 3,000 of 15,000 Write: 3,000 of 15,000

   - This will take some minutes, when done, proceed to create your Dataproc Cluster
4. Create a Service Account for managing the resources
   - On the side menu go to the IAM & Admin section and click on the tool: IAM.
   - Look for the service account that says: ”Compute Engine default service account” and click on Grant Access
   - Add the following permsissioons: **Cloud SQL Admin, Cloud SQL Service Agent, Dataproc Worker**
5. Create a Dataproc Cluster
   - Enable the Cloud Dataproc API
   - Create a Cluster on Compute Engine

- Chosse a Standard Cluster (1 master, N workers)
- Choose the following image: 2.1 (Ubuntu 20.04 LTS, Hadoop 3.3, Spark 3.3)
- On the nodes you can use 3 Worker Nodes with an N2 Machine each. If N2 Machines are not available at the region, try the E2 Machines.
- On the Customize Cluster look for the initialization actions section and add the following bucket: **gs://your-bucket-with-initialization-actions/cloud-sql-proxy.sh** Change the ${Region} set to the region you are creating your cluster.
- On cluster metadata add the following key: enable-cloud-sql-proxy-on-workers and as value: false
- Add this key, value: key: hive-metastore-instance value: ${PROJECT}: ${REGION}:hive-metastore
- On cluster properties we need to add the directory to the hive metastore to be linked to. So on prefix add: **hive** and on key add: **hive.metastore.warehouse.dir** as value write: **gs://${Warehouse_Bucket}/datasets**
- **Recommended:** Set a schedule deletion time for any problem to occur.
- Enable Project access to your cluster, this will work for further steps.
- Your cluster is good to go! Click create cluster. It should take a while to start.

6. Create your Hive tables

   - For creating your Hive tables you should first copy your datasets as a CSV or Parquet file into the bucket we have set as our Hive Database
   - It is recommended to add a new folder into the datasets folder for storing the CSV files
   - Open the CLI for GCP. You can use it locally or start a shell on the GCP site.
   - Write the following command

```
gcloud dataproc jobs submit hive \
--cluster (YOUR_CLUSTER-NAME) \
--region ${REGION} \
--execute "
  CREATE EXTERNAL TABLE my_table
  (StartDate DATE, user_id STRING, Name STRING)
  STORED AS ROW FORMAT DELIMITED FIELDS
  TERMINATED BY "," STORED AS TEXT FIELD
  LOCATION 'gs://${WAREHOUSE_BUCKET}/datasets/my_table';"
```

   - Notice the command, we are submitting our first job to our Hadoop Cluster!! And we are using MySQL flavored code to create our tables
   - Repeat this step as convenient depending on your datasets

7. Running queries on Gcloud

   - As you saw for creating tables, you can use the Dataproc Jobs API to run Hive queries, just change the execution flag with the query you would like to and it will work. It is the only method in which you don't need an SSH authentication.

8. Running queries on SparkSQL

   - This is one of the most useful ways to run queries with Hive. Through SparkSQL
   - Open an SSH session of the Cluster into the master node:

```
gcloud compute ssh my_big_data_cluster-m
```

   - Once connected start a Spark instance by writing pyspark

- Write the following code:

```
from pyspark.sql import HiveContext
hc = HiveContext(sc)
hc.sql("""YOUR QUERY GOES HERE""").show()
```

9. Run your queries with your data. Here are some important questions you can ask yourself for getting a good EDA. **Extra credit:** Create graphs with Python or R given your query results. You can hand in this in a Jupyter Notebook

   - Are my metrics absolute or relative? Try comparing two different columns, one depending from the other, so we have relative metrics instead of only absolute numbers.
   - Join different tables, don't try to query things in just one table, try to do some joins, understand the shcema and find relationships.
   - Are we considering the whole data? How is my data dispersion? Try to consider doing a histogram out of the columns you are selecting. What do you see?
   - Now that you have relative metrics, is your data telling you something? Now try to compare two relative metrics, is something coming up?

10. Go into the Hive Metastore. **Note:** The results of the Hive Metastore you can write them on your evidence document.

    - Do a SQL Connection to the Hive Metastore with the SDK: *gcloud sql connect hive-metastore –user=root*
    - Write *USE hive_metastore;*
    - Query for the data location and lets confirm that everything is being stored here. Try these queries:

```
SELECT s.* FROM hive.TBLS t
JOIN hive.DBS d
ON t.DB_ID = d.DB_ID
JOIN hive.SDS s
ON t.SD_ID = s.SD_ID
WHERE TBL_NAME = 'your_table_name'
AND d.NAME='default';

SELECT INPUT_FORMAT, LOCATION
FROM SDS s, TBLS t
WHERE s.SD_ID = t.SD_ID AND t.TBL_NAME = 'your_table_name';
```