



GCP Data Engineering Bootcamp

Your Take Off to Google Cloud

Creator: Chetan Dixit
Version: 1.0
Release: Oct-2021



All code & commands need to be stored in code repository (create a private GitHub repository and organize the code)



<https://docs.github.com/en/get-started/quickstart/create-a-repo>



Evaluate code on RRR (readability, reusability, repeatability)



Evaluate code on Error and Exception handling, unit tests.



Least the total cost for bootcamp more the score



Make sure you have logging enabled in your code and logs for each activity could be found through cloud logging and monitoring.

Refresh the Concepts

- Read about Cloud Computing basics
- <https://searchcloudcomputing.techtarget.com/definition/public-cloud>
- Reference sites for Python & SQL
- <https://www.tutorialspoint.com/python/index.htm>
- <https://realpython.com/tutorials/basics/>
- <https://docs.python.org/3/tutorial/venv.html>
- <https://www.python.org/dev/peps/pep-0008/>
- <https://www.tutorialspoint.com/sql/index.htm>
- <https://www.tutorialspoint.com/sql/sql-select-query.htm>
- <https://www.tutorialspoint.com/sql/sql-where-clause.htm>
- <https://www.tutorialspoint.com/sql/sql-group-by.htm>

Setting up GCP Account

- Free Google Cloud Account
 - Create your free google cloud account claim your \$300 credit
 - <https://cloud.google.com/free/>
- Paid Google Account (In case you are not eligible for Free account)
 - Check here to setup billing
<https://cloud.google.com/billing/docs/concepts>
 - You don't need Organization, Folders
 - <https://cloud.google.com/billing/docs/how-to/payment-methods>
 - <https://console.cloud.google.com/>

01 **GCP Basics**

Create Google Cloud Project

- Create a separate project for this bootcamp. At the end of bootcamp assessment this project would be deleted to avoid any accidental charges.
- <https://cloud.google.com/resource-manager/docs/creating-managing-projects#console>

Understanding IAM

- IAM overview
- <https://cloud.google.com/iam/docs/overview>
- Read about basic roles
- <https://cloud.google.com/iam/docs/understanding-roles/#basic>
- Read about Service Accounts
- <https://cloud.google.com/iam/docs/service-accounts>
- Check IAM on your project
<https://console.cloud.google.com/home>
- <https://console.cloud.google.com/iam-admin/iam>
- Create two service accounts for Google Cloud Storage Admin and Bigquery Admin roles respectively use them in your subsequent activities.

Activate Google Cloud Shell

- <https://cloud.google.com/shell/docs>
- <https://cloud.google.com/shell>
- Set up default project, region, zones. Choose us-central1 as default region
- Set up Environment Variables
PROJECT_ID, ZONE

Cloud SDK – gcloud & python

- Install Python 3 on your laptop if its not already present.
- Setup gcloud & python sdk on your laptop
- Check & setup gcloud in your Cloud Shell
- <https://cloud.google.com/sdk/docs/install>
- <https://cloud.google.com/sdk/gcloud>
- <https://cloud.google.com/sdk/docs/initializing>

Cloud logging and monitoring

- <https://cloud.google.com/products/operations>
- <https://cloud.google.com/products/operations#section-6>
- <https://cloud.google.com/blog/products/management-tools/cloud-monitoring-improves-custom-dashboard-creation>
- Make sure all your Python Code has logging.
- For all activities, find out your logs in cloud logging.

02 **Network and Compute**

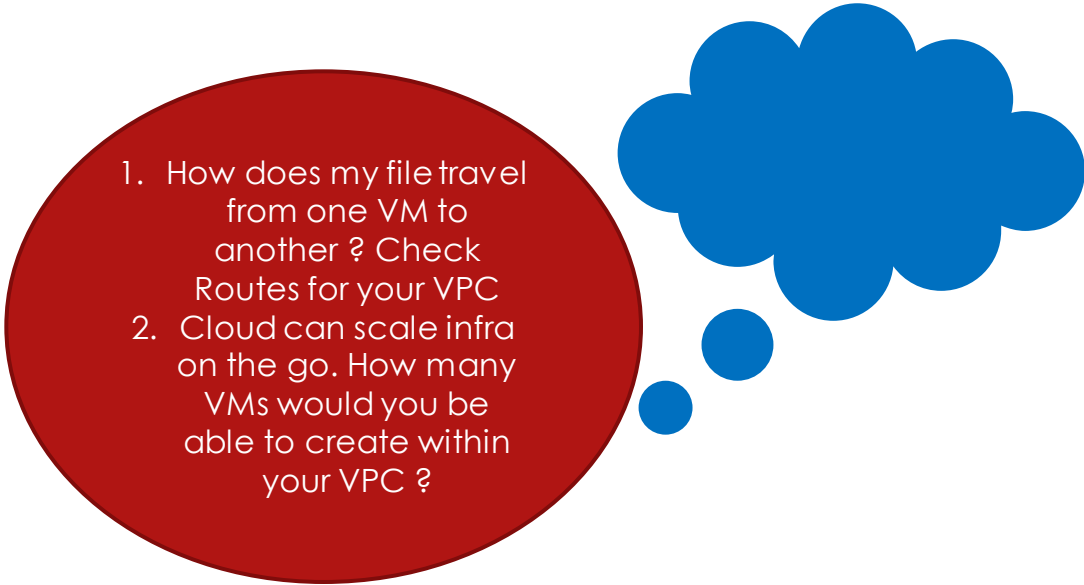
VPC

<https://cloud.google.com/vpc/docs/concepts>

- ▶ Create a VPC Network and two subnets
 - ▶ Use Console
 - ▶ Subnet 1 IP Address Range `10.0.0.0/13`
 - ▶ Subnet 2 IP Address Range `10.8.0.0/13`
 - ▶ Region US Central 1
 - ▶ Other settings as default
 - ▶ Create Firewall rule no https, http allowed

▶ Create two VMs

- ▶ Use gcloud on cloud shell - Create one f1-micro VM in each subnet, No external IP
- ▶ Login to VM in subnet1, create a random text file with text data.
- ▶ Copy the above text file onto other VM in subnet 2
- ▶ Verify the copied file on VM in subnet 2
- ▶ Preserve the screenshots of verification and shutdown the VM

- 
1. How does my file travel from one VM to another ? Check Routes for your VPC
 2. Cloud can scale infra on the go. How many VMs would you be able to create within your VPC ?

03 **Storage & Databases**

Files

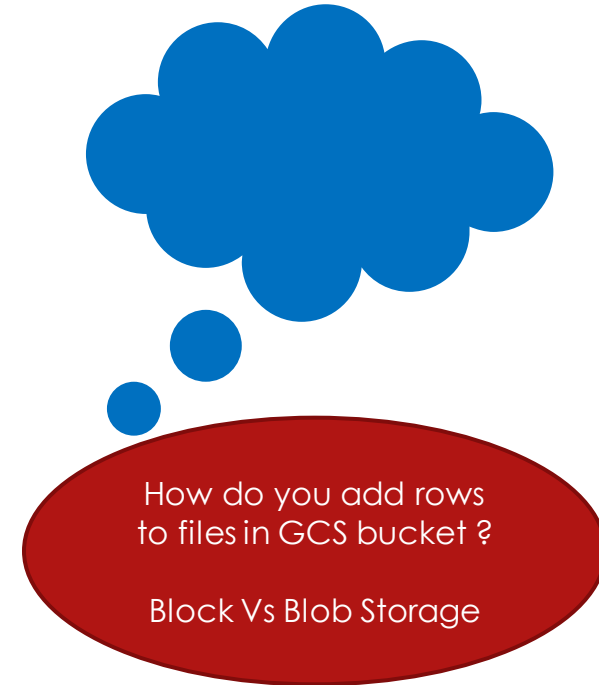
<https://docs.python.org/3/tutorial/inputoutput.html#reading-and-writing-files>

- ▶ Files created in this activity would be needed in next set of activities
- ▶ Create a python program on your laptop to generate a CSV file with 10 columns and 10000 rows. Columns need to be a mix of number, text, , Boolean date and timestamp types. At least one of each type should be present. Save a copy of file in separate folder.
- ▶ Modify your program to add another 5000 lines to your file, and execute to add these rows to file created in above step.
- ▶ You now have two files 1st with 10K rows and 2nd with 15K rows in different folders. They should be having same name.

Google Cloud Storage

<https://cloud.google.com/storage/docs/concepts>

- ▶ Use gcloud and gsutil
- ▶ Create a standard class bucket in region us-central1
- ▶ Enable object versioning on the bucket.
- ▶ Set object lifecycle – objects to be deleted in 48 hours
- ▶ Upload a file (10K rows created in Files section) to bucket
- ▶ Upload the modified file (15K rows created in Files section)
- ▶ Download the 1st version of the file from bucket into a separate folder
- ▶ Verify the downloaded file and 10K original file contents are same.
- ▶ Need to produce proof that they are byte by byte same.



BigQuery – Part 1

<https://cloud.google.com/bigquery/docs>

- ▶ Use bq from Cloud Shell
- ▶ Create a dataset: location US, default table expiration 30 Days
- ▶ Create a table by uploading version1 (10K rows) of file in GCS bucket. Make sure table is partitioned on one of your datetime columns
- ▶ Create another table by uploading version2 (15K rows) of file in GCS bucket. Make sure table is partitioned on one of your datetime columns
- ▶ Write an SQL to generate a result like:

This should be default settings throughout the bootcamp whenever you create dataset

What if you had to do this check on a daily basis ?

Partition	NumOfRowsInTable1	NumOfRowsInTable2	Diff of rows 2 - 1
-----------	-------------------	-------------------	--------------------

BigQuery – Part 2

- ▶ Use a python program with service account
- ▶ Source Data: bigquery-public-data, Dataset->austin_bikeshare, table->bikeshare_trips
- ▶ Create a table partitioned on trip date and clustered on station name in your dataset which would contain following information:
Hourly summary for trip duration and number of trips by station -> trip_date, trip_start_hour, start_station_name, trip_count, total_trip_duration_minutes
- ▶ Create a view on top of above table to show following information:
Highest trips in hour and its station name -> trip_date, trip_start_hour, max_trips, station_name

BigQuery – Part 3

- ▶ Use Bigquery Web Console
- ▶ Source Data: bigquery-public-data, Dataset->baseball, table->schedules
- ▶ Create a Bigquery view to generate following result: For all closed games, find average attendance for weekday and weekend for each home team.

BigQuery – Part 4

GCS + BigQuery + Python

- ▶ Download data from:
- ▶ <https://www.kaggle.com/ashishgup/netflix-rotten-tomatoes-metacritic-imdb>
- ▶ Tasks:
 - ▶ Load raw data from your laptop to GCS bucket using Python.
 - ▶ Create a separate dataset. Load data from GCS to BigQuery table, identify right data types to represent each column into BigQuery and load data using Python. You can use cloud shell to run your python code. Use a service account.
 - ▶ SQL: Find out the number of titles in each Country Availability grouped by Runtime
 - ▶ SQL: Find out Number of Titles against each actor. Should cover all actors available in data.
 - ▶ SQL : Find out the number of Titles for each Genre. Should cover all genres available in data.
 - ▶ SQL: Find out the number of Titles available in each country by Genre.
 - ▶ SQL: Find out top 3 Box Office grossers for each year: Release Year, Title, Box Office, Actors, Genre

Pub Sub – Part 1

<https://cloud.google.com/pubsub/docs>

- ▶ Use gcloud & Cloud Shell
- ▶ Create a topic “topic1” and two pull subscriptions “sub1” and “sub2” for this topic
- ▶ Publish a message -> { "id": 1, "name": "abc"} to the topic
- ▶ Publish 4 more similar messages by incrementing id and changing the value of name.

Pub Sub – Part 2

<https://cloud.google.com/pubsub/docs>

- ▶ Use Python program, use Service Account
- ▶ Use topics and subscriptions created earlier
- ▶ You can execute your Python program from Cloud Shell
- ▶ Read messages from “topic1” using “sub1” subscription and write them into a file -> file1

Pub Sub – Part 3

<https://cloud.google.com/pubsub/docs>

- ▶ Use Python program, use Service Account
- ▶ Use topics and subscriptions created earlier
- ▶ You can execute your Python program from Cloud Shell
- ▶ Publish five more messages to the “topic1” by incrementing id and changing the value of name.

Pub Sub – Part 4

<https://cloud.google.com/pubsub/docs>

- ▶ Use Python program, use Service Account
- ▶ Use topics and subscriptions created earlier
- ▶ You can execute your Python program from Cloud Shell
- ▶ Read messages from “topic1” using “sub2” subscription and write them into a file-> file2
- ▶ What is the difference between file1 (created in part 2) and file2 ? Why?

Cloud SQL -PostgreSQL

- ▶ Use Web Console
- ▶ Create a PostgreSQL-12 Instance “mypginstance”, us-central1 region, single zone, Lightweight, 1 vCPU, 3.75GB, SSD Storage, 20GB Capacity, Backups Disabled,
- ▶ Note down your password
- ▶ Create a database named myorg
- ▶ Create following tables and insert data into these tables. Hint: In your Cloud Shell use `gcloud sql connect & psql`
- ▶ Employee: emp_id, name, dept_id, salary, joining_date, leaving_date, is_active
- ▶ Department: dept_id, dept_name, dept_head_id
- ▶ Project: proj_id, proj_name, dept_id, proj_start_date, proj_end_date
- ▶ Project_staff: proj_id, emp_id, role_name, start_date, end_date
- ▶ Populate atleast 15 employee, 3 department, 3 project, 10 project_staff rows. All data should follow primary key & foreign key constraints

Big Table

▶ Video Learning

- ▶ <https://www.youtube.com/watch?v=P4q4nqJAamo>
- ▶ <https://www.youtube.com/watch?v=MglAys6RxKE>
- ▶ Handle massive workloads with Cloud Bigtable database service → <https://goo.gle/3kkbRYU>
- ▶ QuickStart for Bigtable using Cloud Shell → <https://goo.gle/3klbzkv>
- ▶ Cloud Bigtable in action (NEXT 2017) → <https://goo.gle/2WmRe6d>
- ▶ Cloud Bigtable Documentation → <https://goo.gle/3kCTNLa>
- ▶ Bigtable vs. BigQuery - what's the difference? → <https://goo.gle/3IRHnhQ>

04 Ingestion

Dataflow – Part 1

<https://cloud.google.com/dataflow/docs>

<https://cloud.google.com/dataflow/docs/samples>

- ▶ Use your earlier python code to generate a CSV file “f100k.csv” with 100K rows.
- ▶ Create a new regional (us-central1) GCS bucket and Upload file “f100k” to GCS bucket
- ▶ Create a new BigQuery dataset “dfbatch” with location US, default table expiration 30 Days, create a table “t100k” with schema for your file “f100k.csv”
- ▶ Create a dataflow batch pipeline using Python to load data from file “f100k.csv” in GCS bucket to BigQuery Table “t100k”. Use VPC network created at the start of the bootcamp and machine type as n1-standard-2 and not more than 3 instances while launching pipeline.

Dataflow – Part 2

<https://cloud.google.com/dataflow/docs>

<https://cloud.google.com/dataflow/docs/samples>

- ▶ Use Web console and create a pubsub topic “dftopic” and subscription “dfsub”
- ▶ Use your python program from PubSub – Part 3 to publish Json messages to this topic. Do not execute yet.
- ▶ Create a new BigQuery dataset “dfstream” with location US, default table expiration 30 Days. Create a new table “tstream” with schema for your messages.
- ▶ Create a python streaming dataflow pipeline to consume data from subscription “dfsub” and write to Bigquery Table “tstream”.
- ▶ Launch your pipeline and check for successful running.
- ▶ Execute your python program to publish 5 messages every 5 seconds for 5 minutes.
- ▶ Monitor the pipeline for pub sub messages ack, unack, received etc
- ▶ After 5 minutes ensure your pipeline processes all messages before you shutdown (drain) your pipeline. Explore pipeline shutdown options.

Python + Compute Engine

- ▶ Launch a ubuntu/Linux/centos compute engine n1-standard2 in your VPC. Make sure you have Python 3 and python sdk and related packages for postgresql installed
- ▶ Create a BigQuery dataset named “pgdataset” with location US, default table expiration 30 Days.
- ▶ Write a python program to read all 4 tables from database “myorg” in Cloud SQL Instance “mypginstance” and create tables in Bigquery dataset “pgdataset” with data from those tables.

Cloud Composer

<https://cloud.google.com/composer/docs>

- ▶ Create an Avro file (with at least 1000 records) and upload it on GCS bucket using python program
- ▶ <https://avro.apache.org/docs/1.10.2/gettingstartedpython.html>
- ▶ Create a DAG to upload Avro file data from GCS bucket to a Bigquery table. (create a separate dataset for this activity)
- ▶ <https://cloud.google.com/composer/docs/quickstart>

Cloud Function

<https://cloud.google.com/functions/docs>

- ▶ Code a python Cloud Function to read data from GCS bucket file and load data into a Bigquery table. Create a separate bucket and bigquery dataset for this activity.
- ▶ Cloud function should trigger on a file arrival at GCS bucket.
- ▶ Upload at least 5 files, each with at least 10K rows, simultaneously to your GCS bucket.

Dataproc

<https://cloud.google.com/dataproc/docs>

- ▶ Create a Dataproc spark cluster from console
- ▶ Name bootcamp, Region us-central1, Zone us-central1-c
Configure nodes, for Master node - Machine type 2 vCPUs (n1-standard-2), 2 Worker nodes - Machine type 2 vCPUs (n1-standard-2), Choose VPC that you created earlier for your spark cluster.
- ▶ Submit an example spark job. Spark examples are usually at <file:///usr/lib/spark/examples/jars/spark-examples.jar>
- ▶ Find out the details of Job what it has done and download/screengrab the Job Details/Logs

Datafusion

<https://cloud.google.com/data-fusion/docs>

- ▶ Video Learning
- ▶ <https://www.youtube.com/watch?v=xjsNWh1TLKo>
- ▶ <https://www.youtube.com/watch?v=kehG0CJw2wo> (start from 17th Minute)
- ▶ <https://cloud.google.com/data-fusion/docs/quickstart>

05 Exploration & Consumption

Dataprep

<https://cloud.google.com/dataprep/docs/concepts>

- ▶ Use the table from dataset created in BigQuery – Part 4
- ▶ Setup your Dataprep environment it may take a while to get the setup running.
- ▶ Create a new flow – “Netflix Data Exploration”
- ▶ Import Data - use table from BigQuery – Part 4
- ▶ Edit Recipe
- ▶ Explore to find out number of columns, data types
- ▶ What is the most common value in Genre
- ▶ Explore top 3 “Country Availability” with number of titles
- ▶ Explore “how many distinct values in each column”
- ▶ What is the Maximum and Minimum Box Office Amount.
- ▶ Can you change data types of columns? How ?
- ▶ Can you remove duplicates from data ? How ?

Data Studio

<https://developers.google.com/datastudio>



Microsoft Excel
Worksheet

- ▶ Refer to embeded Excel Sheet
- ▶ Generate data using Python where possible.
- ▶ Generate data for 20 Customers, 50 Products, 1000 Orders, 10000 order_quantity. Have it in CSV or JSON or Excel (your choice).
- ▶ Create 4 respective tables in a new dataset named "mydukan"
- ▶ Load data to Bigquery tables from above files using Python
- ▶ Create an Invoice report in Data Studio – refer to tab "Data Studio Invoice" for details. Yellow cells represent data to be filled rest are column headers. If there are 1000 Orders then report should have 1000 Order Details. Ability to filter on Order ID and/or Customer ID
- ▶ **Pro Level- go for it:** Generate order (10 or so) and order_quantity (related to orders) files upload it to GCS. Make sure these files have customer and products from your master data. Write a Python Cloud Function to be triggered on file upload to load that data in BigQuery tables. View the same data in your Data Studio Report and get the PDF. Modify your program to generate files every 2 mins (for 10 mins) and load to GCS and see results in your Data Studio report and get the PDFs