**Objective**

The goal in this assignment is to create an ingest to ETL pipeline using CSV files and Google BigQuery. *Schedule a recurring cron job to batch update the data (No points will be deducted if this part is skipped) 10/12/22*.

**Step 1:** **Create a GCP project**

Once logged in to the Google Cloud console, you will create a new project by naming it and making sure that it has a billing profile attached to it. Once created, make sure that you select that project so that you are working under it, you should see a page that prompts you that “You’re working in <name of project>”.

**Step 2: Activate Cloud shell**

Click on the cloud shell icon on the right hand side of the page otherwise you can search for it on the search bar. If you were able to open it up via the cloud shell icon in the project page it should automatically be configured to that project on the shell. However, if it’s not, you will need to configure it through the following commands.

| **gcloud** projects list |
| --- |

| **gcloud** config set project <name **of** project> |
| --- |

**Step 3 : Clone Github repository**

Clone the github repo from GCP and work from the data ingest examples of Chapter 2

**Chapter Ex:**

| **git clone https://github.com/GoogleCloudPlatform/data-science-on-gcp** |
| --- |

**Project:**

| **git clone https://github.com/luperodriguez/MSDS434Project** |
| --- |

To check that the repo successfully cloned into your project, list out folders in that project and confirm that data-science-on-gcp is there.

| ls |
| --- |

Change directories into the folder that examples from Chapter 2

| **cd** data-science-on-gcp/ |
| --- |

| ls |
| --- |

| **cd** 02\_ingest/ |
| --- |

**Step 4 : Make a Data directory**

Create a data directory that will hold the examples from chapter 2

| **mkdir** data |
| --- |

Copy the download shell script into the data directory

| **cp** download.sh data |
| --- |

Change into the data directory

| **cd** data |
| --- |

Confirm that the download shell script file was successfully copied into the data directory.

| ls |
| --- |

**Step 5: Run a Query**

Run a query that will filter and download the data for the year ‘2015’ by Month

| **for** **MONTH** **in** ‘seq 1 12’; do  **bash** download.sh 2015 $**MONTH** **done** |
| --- |

Confirm that it downloaded 12 csv files (one for each month of 2015) were successfully copied into the data directory.

| ls |
| --- |

| **cp** 201501.csv 201501.bck |
| --- |

Remove the rest of the .csv file to save on usage

| **rm** \*.csv |
| --- |

**Step 6: Read file into Python**

| python |
| --- |

Once Python loads into the powershell, create a variable function with the following command

***Chapter Ex:***

| **f = open(“201501.bck”).readline()** |
| --- |

***Project:***

| **f = open('Redfin\_National.bck').readline()** |
| --- |

Perform some exploration of the data rows and then write a csv file of the first 5000 rows

| **f** [0] **f** [1] |
| --- |

| **fout** = open("201501","w") **for** i **in** range(5000):   fout.write(f[i]) |
| --- |

| ls |
| --- |

| **cat** 201501.csv |
| --- |

**Step 7: Create a cloud storage bucket**

On the google console page, search for bucket cloud storage. From there create a bucket and provide it a name. Click on the create button

Once the bucket is set up you can then do a google storage (gs) utility copy from the cloud shell terminal. In order to copy all the files in the current directory to the storage bucket we previously created, we will use the command gsutil cp \* gs://<YOUR\_BUCKET\_NAME>.One important aspect to note is that when performing an upload of multiple large files, using the flag -m, to perform a parallel (multi-threaded/multi-processing) copy, will significantly improve the performance.

| **gsutil** -m cp \*.csv gs://<**YOUR\_BUCKET\_NAME**> |
| --- |

***Project:***

| **gsutil** -m cp \*.csv gs://demo-4 |
| --- |

**Step 8: Create a new dataset in BiqQuery**

Create a new dataset that will hold your table

| **bq** mk demo4ongcp |
| --- |

**Step 9: Create a new table in BiqQuery**

Create a new table where the data will be loaded into using the following command

| **bq** mk -t demo4ongcp.national |
| --- |

**Step 10: Load Data into BiqQuery**

We will now load the data using another bq command. We also need to create a schema while loading the data. For that we will use the auto-detect command along with bq. We will be giving the path of the Cloud storage bucket and the destination table details in the command.

**Chapter Ex:**

| **bq** load --autodetect --source\_format=CSV **demo4ongcp**.flight\_auto gs://<name **of** bucket>bucket/201501.csv |
| --- |

***Project:***

| **bq** load --autodetect --source\_format=CSV demo4ongcp.national gs://demo-4/Redfin\_National.csv |
| --- |

