



# Combating COVID-19 Outbreaks

WQD7005 Data Mining  
Master of Data Science | University of Malaya

**Part C:** Processing of Data Assignment

Group Members:

Azwa bin Kamaruddin (WQD170089)

Kok Hon Loong (WQD170086)



UNIVERSITY  
OF MALAYA

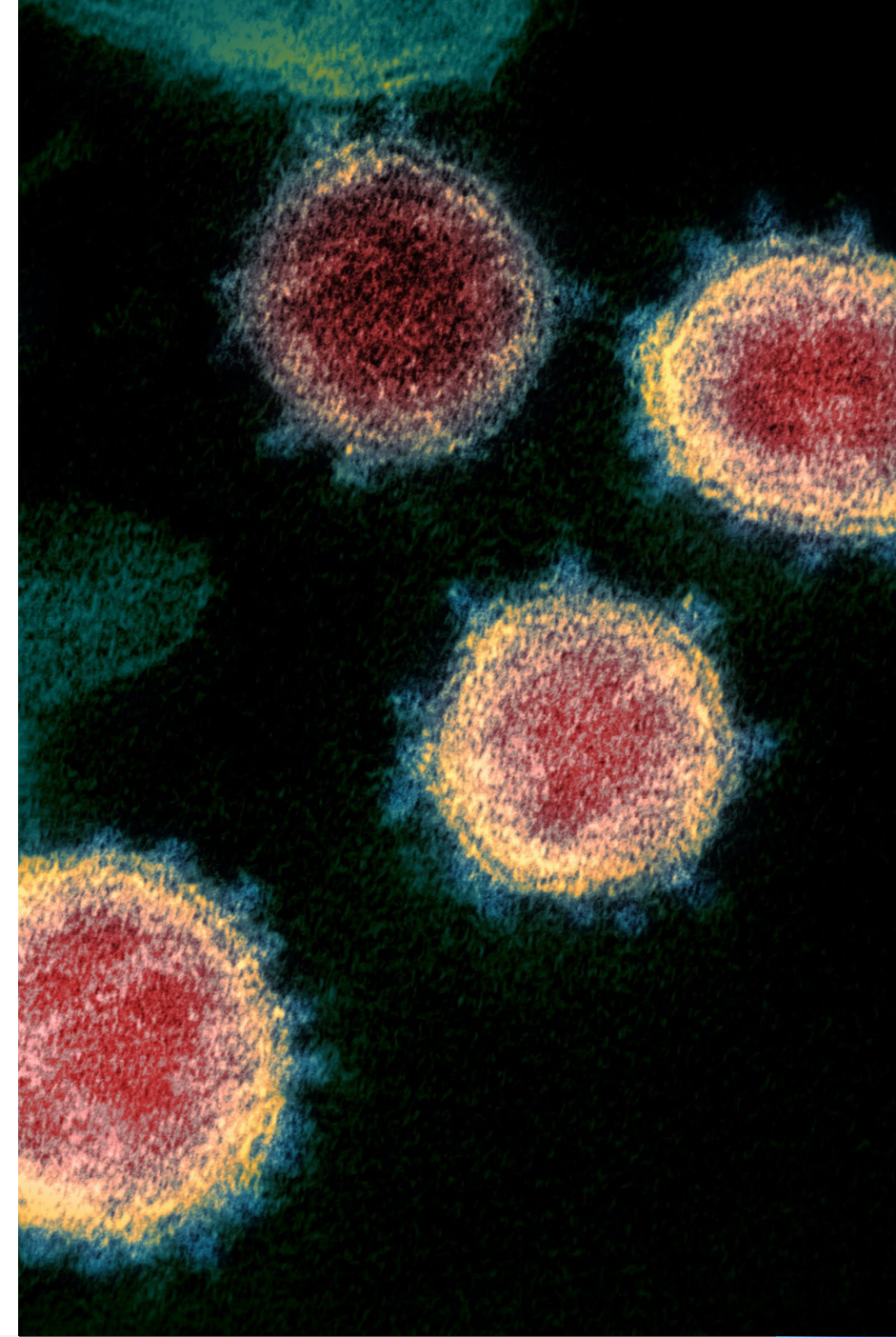


# Assignment Milestones

WQD7005 Data Mining | Semester 2 Session 2019/2020



- Part A: (Group)
  - Web Crawling of Real-time Data
- Part B: (Group)
  - Management of Data using Hadoop Data Warehouse or Data Lake
- **Part C: (Group)**
  - Accessing and Processing of Data from Hadoop Data Warehouse or Data Lake using Python
- Part D: (Individual)
  - Interpretation and Communication of Data Insights
- Part E: (Group)
  - Deployment of the Data Mining Results on Web (Flask) and Mobile Application (Kivy)



# Assignment Background

From the previous assignment milestones, our group have accomplished the following tasks:

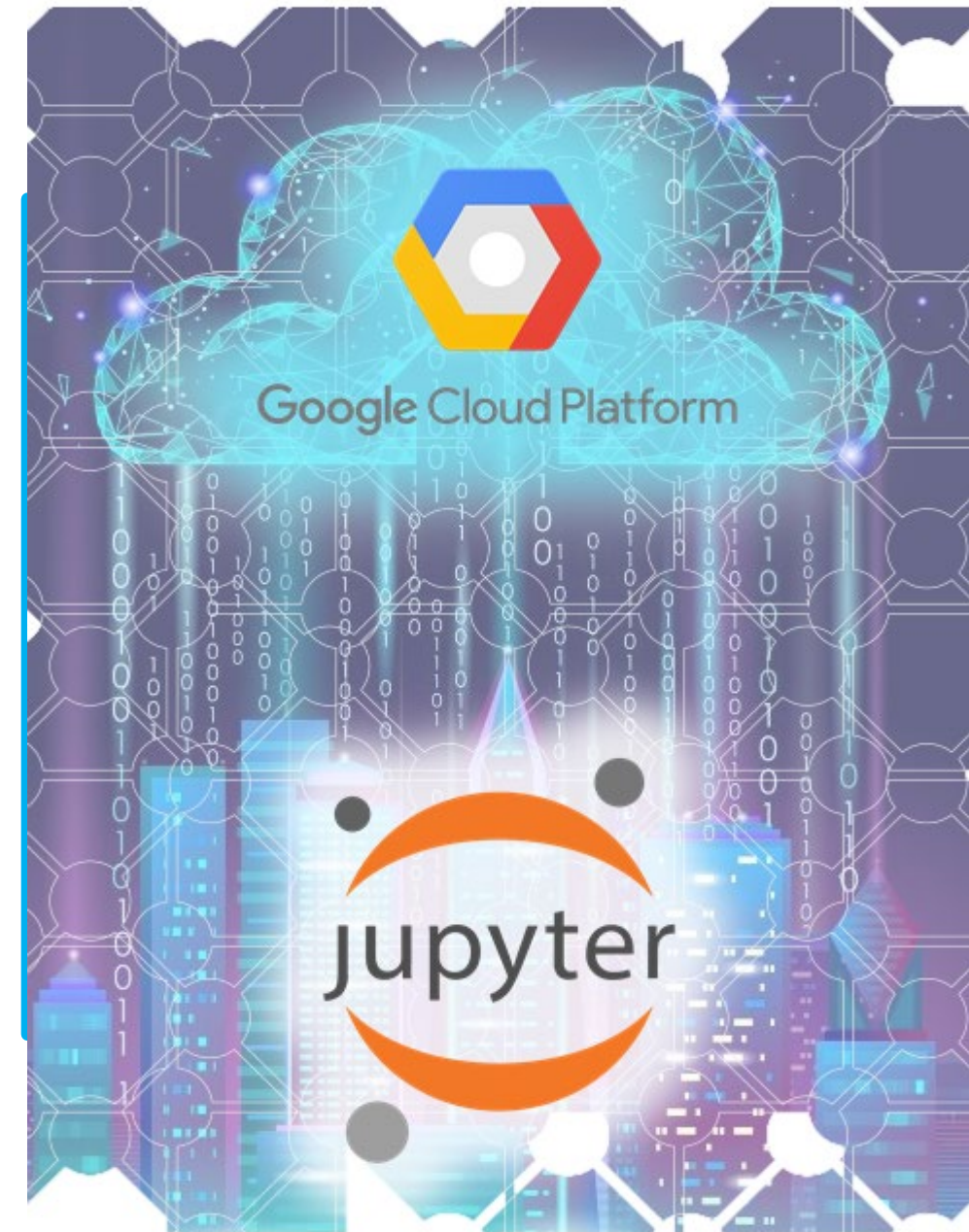
- I. We have demonstrated our method in acquiring the near real-time data using the web crawling approach using Python programming and BeautifulSoup package.
- II. Illustrating the steps in storing the data using cloud storage (a.k.a. data lake) and also to the Hadoop data warehouse leveraging on the DataProc configuration and implementation using Google Cloud Platform (GCP), which the data was then being accessed using Hive.

For this assignment milestone, our group will be focusing on the following:

- ▣ Using **Python** that is coded on **Jupyter Notebook** web application in accessing the varieties of the stored data types (i.e. CSV, JSON) from the cloud storage.

Our group Python code is also uploaded in our group assignment GitHub at the link below:

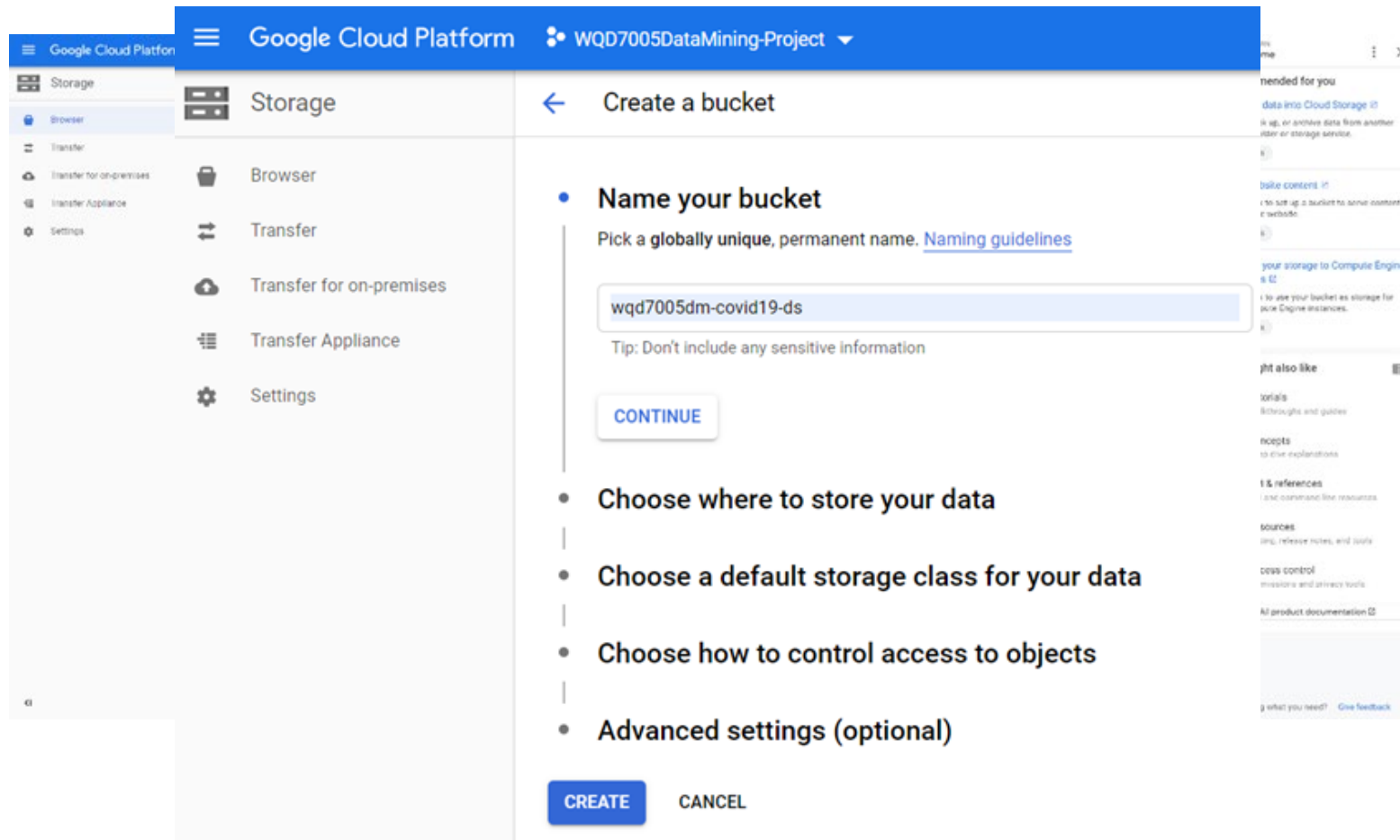
- <https://github.com/scholarazwa/wqd7005-assignment>





# Accessing and Processing the Datasets

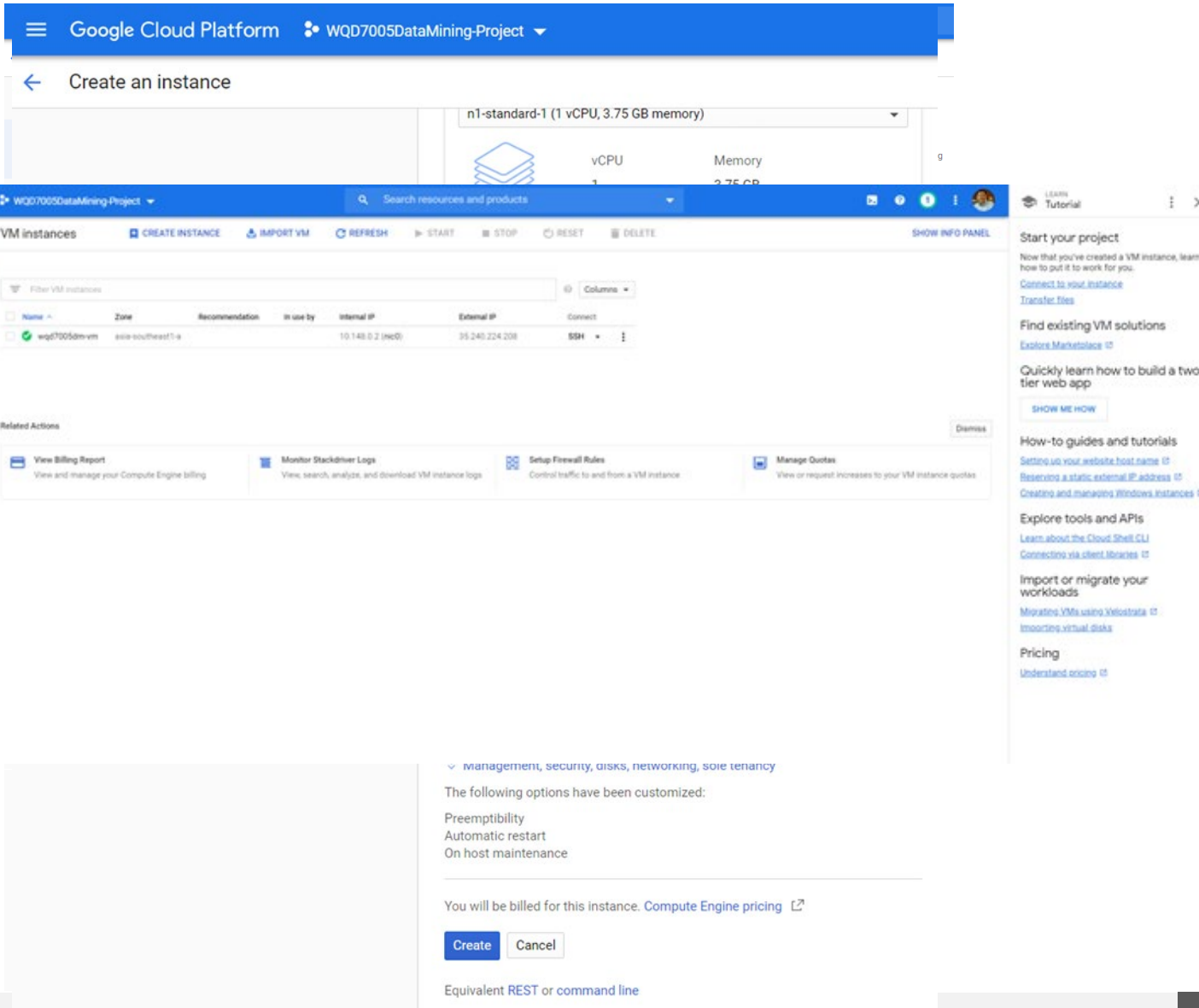
creating Cloud Storage (Data Lake) in Google Cloud Platform (GCP)



- Scroll to the **Storage** section in GCP **Navigation menu**, then click on **Browser**.
- Click on **Create Bucket**, to create the storage container to store the data captured from web crawling.
- Name the storage container as **wqd7005dm-covid19-ds**.

# Accessing and Processing the Datasets

## creating Virtual Machine Instances in Google Cloud Platform (GCP)



The screenshot displays the Google Cloud Platform (GCP) console interface. At the top, the navigation bar shows 'Google Cloud Platform' and the project 'WQD7005DataMining-Project'. Below this, the 'Create an instance' page is visible, showing a dropdown for 'n1-standard-1 (1 vCPU, 3.75 GB memory)'. The main content area shows the 'VM instances' page with a table of existing instances:

Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
wqd7005dm-vm	asia-southeast1-a			10.148.0.2 (nbd)	35.240.224.208	SSH

Below the table, there are 'Related Actions' such as 'View Billing Report', 'Monitor Stackdriver Logs', 'Setup Firewall Rules', and 'Manage Quotas'. At the bottom, a dialog box is open, showing the 'management, security, disks, networking, sole tenancy' section. It lists customized options: 'Preemptibility', 'Automatic restart', and 'On host maintenance'. It also states 'You will be billed for this instance. Compute Engine pricing' and includes 'Create' and 'Cancel' buttons.

- In the **Compute** section, select **Compute Engine** and then click on **VM instances**.
- Click on the **Create** button in the new pop-up window.
- Use the minimal settings and named the VM instances as **wqd7005dm-vm**.
- Our group chose to use **Ubuntu** as the OS with **minimal setting**. Then click on the **Create** button to provision the **VM instances** to install **Anaconda** with **Jupyter Lab** for **Python**.

# Accessing and Processing the Datasets

installing Anaconda and Jupyter Lab in Google Cloud Platform (GCP) VM instances

```
wqd170086@wqd7005dm-vm: /tmp - Google Chrome
ssh.cloud.google.com/projects/wqd7005datamining-project/zones/asia-southeast1-a/instances/wqd7005dm-vm?authuser=1&hl=en_U...
Connected. host fingerprint: ssh-rsa 0 F8:36:7A:C9:04:C8:10:40:E7:50:3B:99:2C:A3
wqd170086@wqd7005dm-vm:/tmp$ curl -O https://repo.anaconda.com/archive/Anaconda3-2020.02-Linux-x86_64.sh
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 521M  100 521M    0     0  220M      0  0:00:02  0:00:02 --:--:-- 220M
wqd170086@wqd7005dm-vm:/tmp$ ls -al
total 534136
drwxrwxrwt  9 root      root      4096 Apr 22  07:42 .
drwxr-xr-x 23 root      root      4096 Apr 22  07:41 ..
drwxrwxrwt  2 root      root      4096 Apr 22  07:40 .ICE-unix
drwxrwxrwt  2 root      root      4096 Apr 22  07:40 .Test-unix
drwxrwxrwt  2 root      root      4096 Apr 22  07:40 .X11-unix
drwxrwxrwt  2 root      root      4096 Apr 22  07:40 .XIM-unix
drwxrwxrwt  2 root      root      4096 Apr 22  07:40 .font-unix
-rw-rw-r--  1 wqd170086 wqd170086 546910666 Apr 22  07:42 Anaconda3-2020.02-Linux-x86_64.sh
drwx----- 3 root      root      4096 Apr 22  07:41 systemd-private-ef235d0d438e42299c2236edcb8d7b75-chrony.se
rvice-DI4VcX
drwx----- 3 root      root      4096 Apr 22  07:40 systemd-private-ef235d0d438e42299c2236edcb8d7b75-systemd-r
esolved.service-q8SLul
wqd170086@wqd7005dm-vm:/tmp$ sha256sum Anaconda3-2020.02-Linux-x86_64.sh
2b9f088b2022edb474915d9f69a803d6449d5fdb4c303041f60ac4aefcc208bb Anaconda3-2020.02-Linux-x86_64.sh
wqd170086@wqd7005dm-vm:/tmp$ bash Anaconda3-2020.02-Linux-x86_64.sh

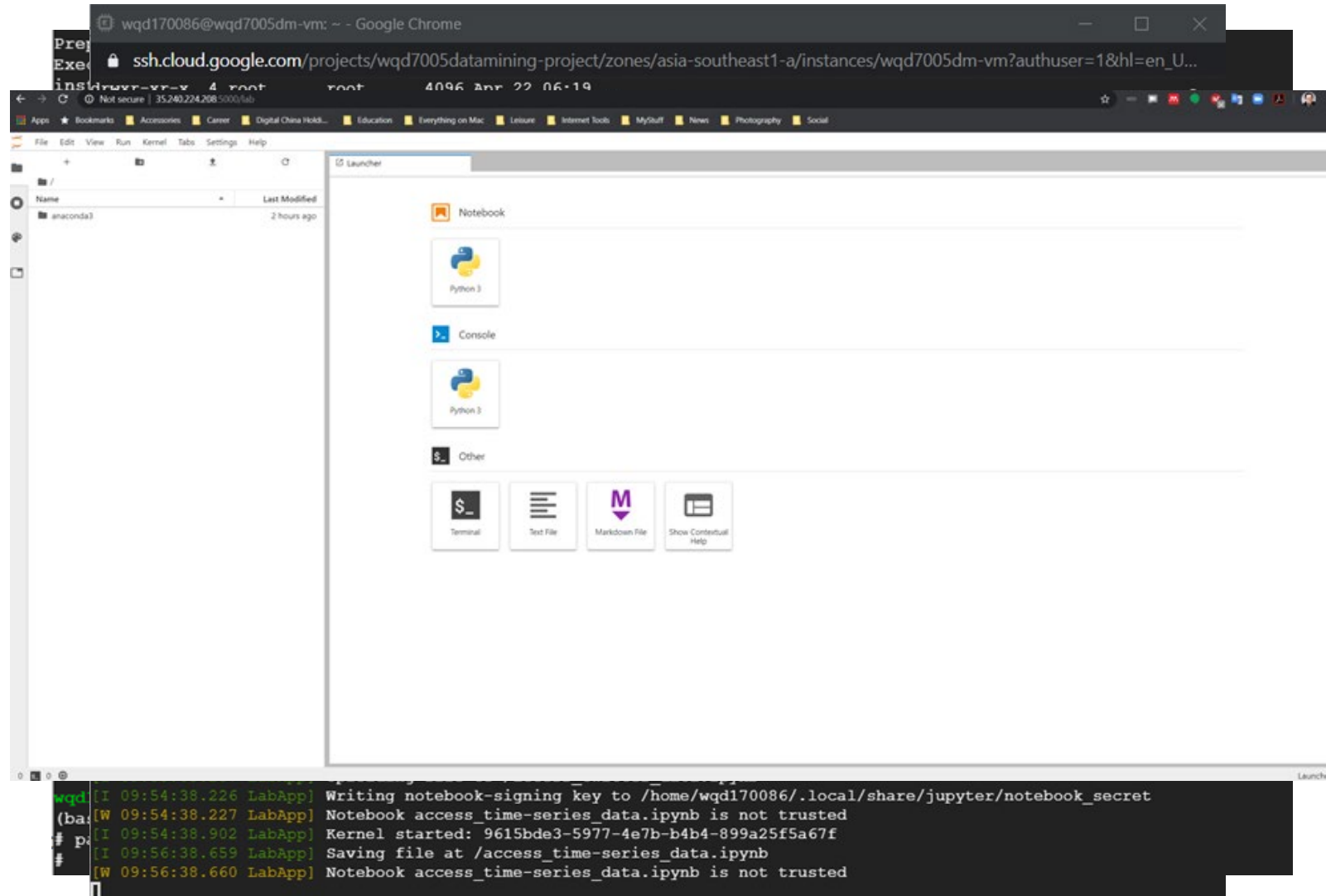
Welcome to Anaconda3 2020.02

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>> 
```

- Click on the **SSH** pull down menu, select **Open in browser window**.
- The **shell command prompt** running on Ubuntu Linux will appear.
- To install Anaconda, use the web browser to connect and view the Anaconda Distribution site @ <https://www.anaconda.com/distribution/>
- Choose the latest Linux OS version and from the command prompt, type the following command to download the bash script: `curl -O https://repo.anaconda.com/archive/Anaconda3-2020.02-Linux-x86\_64.sh`
- Use the **SHA-256 checksum** command before running the Anaconda **installation bash script**.

# Accessing and Processing the Datasets

to run Jupyter Lab in Google Cloud Platform (GCP) to start coding Python



- To activate the installation, type **source ~/.bashrc**.
- To test the installation and activation, type **conda list**.
- Ensure to install the following packages if this is the **first time running** Anaconda and Jupyter Lab
  - To support importing storage in GCP: **pip install --upgrade google-cloud-storage**
  - To support importing Twitter module: **pip install tweepy**
- To start Jupyter Lab from the Ubuntu command prompt (based on the **port #** set up during configuration), type **jupyter-lab --ip=0.0.0.0 --port=5000 --no-browser**
- To access to the Jupyter Lab notebook, use a web browser to key in the token generated from the above command and the static IP address configures, i.e. **http://<Static IP addr>:<Port>#/?token=f9823a614febd8fb5997ab9bb7d98536db463d4de8000c4d**

**Note:** The token to use is different each time the jupyter lab is executed.



# Accessing and Processing the Datasets

using Python on Jupyter Lab notebook in Google Cloud Platform (GCP) VM instances

```
crawl_time-series_data.ipynb X access_time-series_data.ipynb X crawl_twitter_data.ipynb X access_twitter_data.ipynb X
Python 3

# This code is to ACCESS the stored crawled tweets in Google Cloud Storage data lake.

[2]: import pandas as pd
import json
from google.cloud import storage
from io import BytesIO, StringIO

[3]: client = storage.Client()
bucket = client.get_bucket('wqd7005dm-covid19-ds')
blob = bucket.get_blob('twitter_data.json')
data = blob.download_as_string()

[4]: # Read the JSON file from the data lake:
sstr(data, 'utf-8')
df = StringIO(s)

[5]: # Load the JSON file here:
json.load(df)

[5]: {'statuses': [{'created_at': 'Thu Apr 23 05:50:21 +0000 2020',
'id': 1253199510706302976,
'id_str': '1253199510706302976',
'text': 'All that Corona has taught me is that if a Zombie Virus ever existed we would be immediately fucked',
'truncated': False,
'entities': {'hashtags': [],
'symbols': [],
'user_mentions': [],
"urls": []},
'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
'source': '<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>',
'in_reply_to_status_id': None,
'in_reply_to_status_id_str': None,
'in_reply_to_user_id': None,
'in_reply_to_user_id_str': None,
'in_reply_to_screen_name': None,
'user': {'id': 902969151500767232,
'id_str': '902969151500767232',
'name': 'krist n',
'screen_name': 'el_nene_sato',
'location': '',
'description': 'he/him. bi. chaotic good. D&D enthusiast.',
'url': None,
'entities': {'description': {'urls': []}},
'protected': False,
'followers_count': 301,
'friends_count': 659,
'listed_count': 1,
'created_at': 'Wed Aug 30 18:59:56 +0000 2017',
'favourites_count': 45699,
'profile_image_url': 'http://t1.gstatic.com/profileimg/u/902969151500767232/300x300.jpg'}}]}
```

- In this assignment, our group have created 4 sets of Python programming codes:
  - a. Crawl for time-series data (**csv** data sets)
  - b. Access time-series data
  - c. Crawl for Tweets from Twitter on relevant topics (**json** data set)
  - d. Access Tweets data
- For this assignment, we have demonstrated the following:
  - I. Using web crawling method with Python codes to acquire varieties of data sets;
  - II. Accessing the data sets acquired using Python programming from our Cloud Storage (i.e. Data Lake)

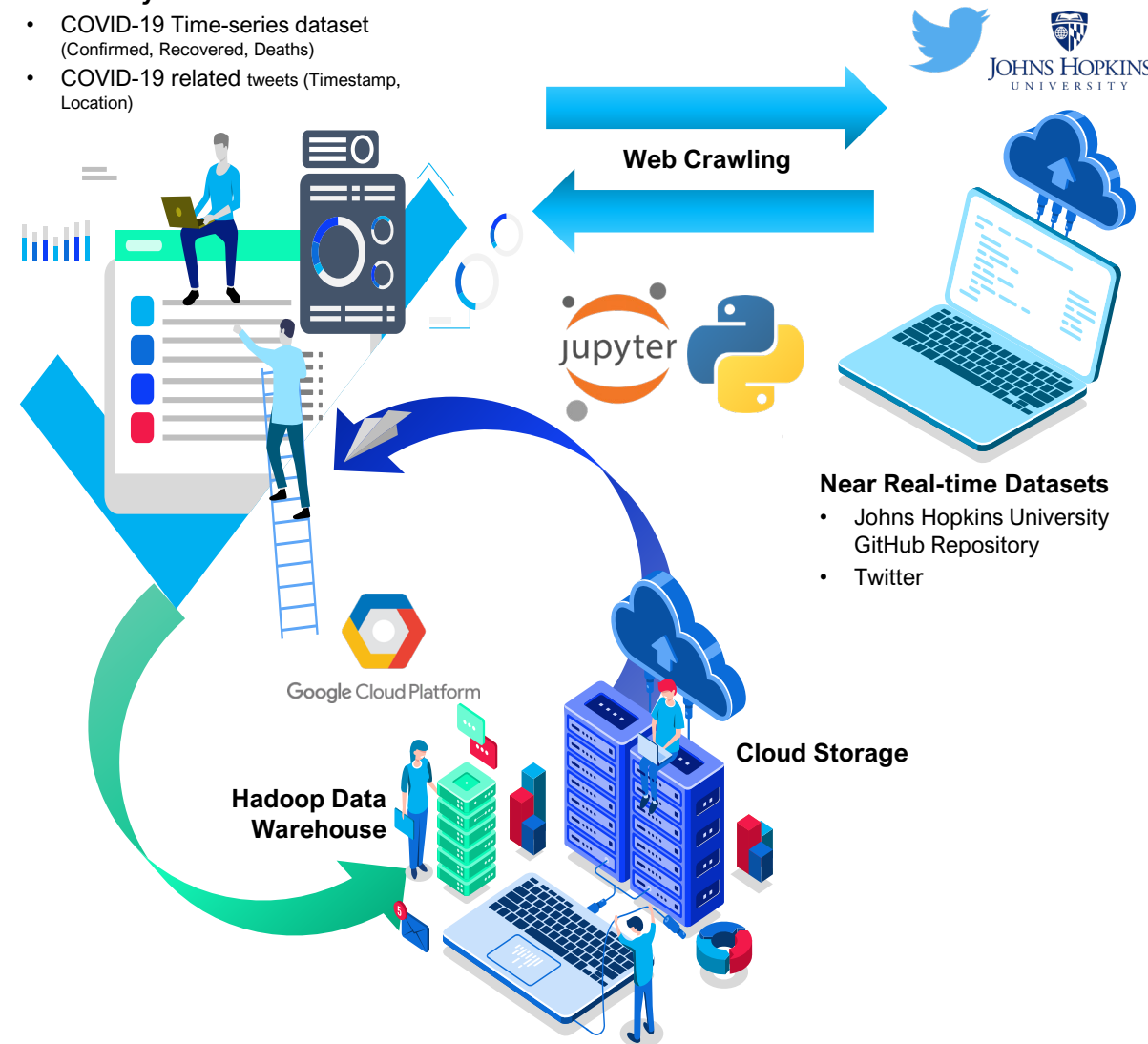


# Summary

- Within a typical data mining project, there are usually many different types of operations management software with many different data storage as well.
- As the number of data sources multiplies, having data scattered all over in various formats prevents the data analysts from seeing the full and clear picture of their current state.
- This creates the necessity for integrating data in a unified storage system where data is collected, reformatted, and ready for use.

## Data Acquisition & Exploratory Data Analysis

- COVID-19 Time-series dataset (Confirmed, Recovered, Deaths)
- COVID-19 related tweets (Timestamp, Location)







# Thank You

Azwa Kamaruddin & Hon-Loong Kok (HL)

