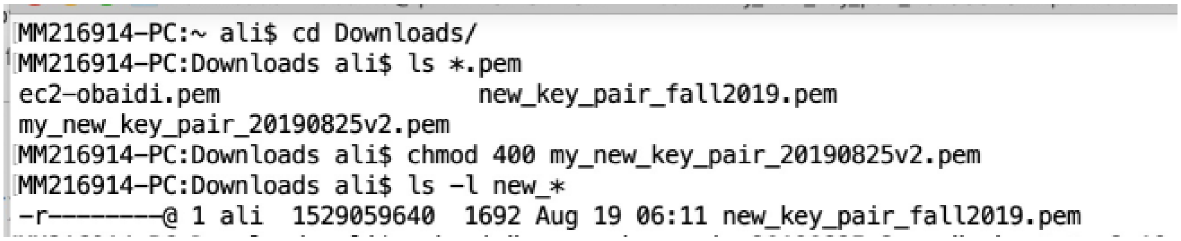
AWS/Dashboard/Database Guide

Part 1: Downloading the Key (*THIS IS ONLY AN INITIAL STEP FOR USERS WHO DO NOT HAVE THE CORRECT KEY OR HAVE NOT INSTALLED A KEY YET*)

Step 1: Download key in Downloads

Step 2: Pick the right .pem



```
MM216914-PC:~ ali$ cd Downloads/
MM216914-PC:Downloads ali$ ls *.pem
ec2-obaidi.pem                    new_key_pair_fall2019.pem
my_new_key_pair_20190825v2.pem
MM216914-PC:Downloads ali$ chmod 400 my_new_key_pair_20190825v2.pem
MM216914-PC:Downloads ali$ ls -l new_*
-r--------@ 1 ali  1529059640  1692 Aug 19 06:11 new_key_pair_fall2019.pem
```

Part 2: Connecting to the AWS Instance

Step 3: Login to AWS

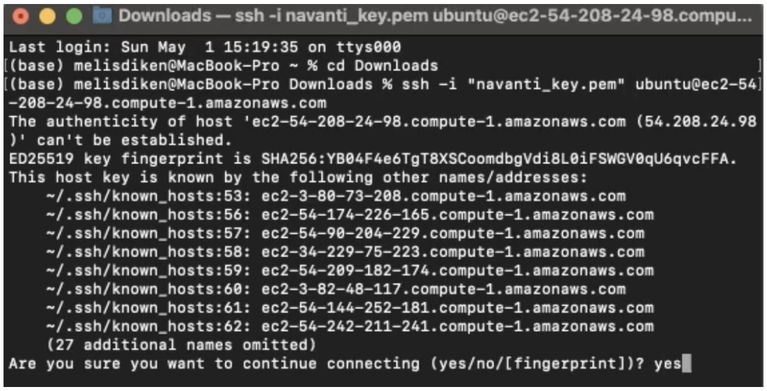
Step 4: Start Instance

Step 5: Connect and copy second link

Step 6: type “cd Downloads”

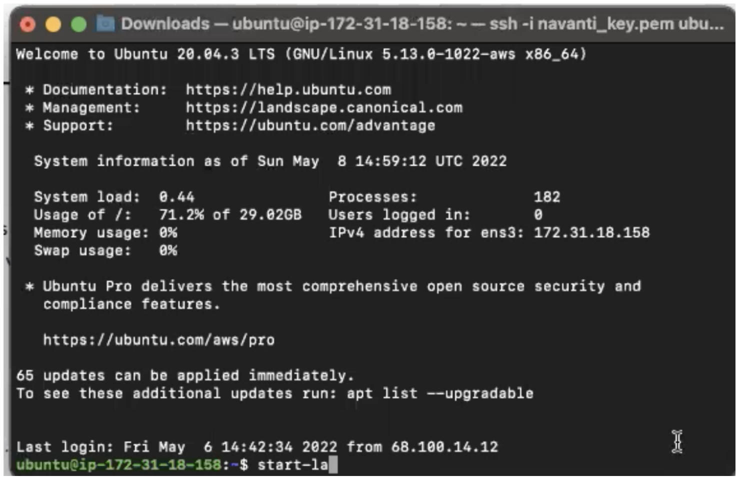
Step 7 : Paste link and change “root” to “ubuntu”

Step 8: Say “yes” and “start-lab”



Downloads — ssh -i navanti_key.pem ubuntu@ec2-54-208-24-98.compu...

```
Last login: Sun May  1 15:19:35 on ttys000
(base) melisdiken@MacBook-Pro ~ % cd Downloads
(base) melisdiken@MacBook-Pro Downloads % ssh -i "navanti_key.pem" ubuntu@ec2-54
-208-24-98.compute-1.amazonaws.com
The authenticity of host 'ec2-54-208-24-98.compute-1.amazonaws.com (54.208.24.98
)' can't be established.
ED25519 key fingerprint is SHA256:YB04F4e6TgT8XSCoomdbgVdi8L0iFSWGV0qU6qvcFFA.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:53: ec2-3-80-73-208.compute-1.amazonaws.com
    ~/.ssh/known_hosts:56: ec2-54-174-226-165.compute-1.amazonaws.com
    ~/.ssh/known_hosts:57: ec2-54-90-204-229.compute-1.amazonaws.com
    ~/.ssh/known_hosts:58: ec2-34-229-75-223.compute-1.amazonaws.com
    ~/.ssh/known_hosts:59: ec2-54-209-182-174.compute-1.amazonaws.com
    ~/.ssh/known_hosts:60: ec2-3-82-48-117.compute-1.amazonaws.com
    ~/.ssh/known_hosts:61: ec2-54-144-252-181.compute-1.amazonaws.com
    ~/.ssh/known_hosts:62: ec2-54-242-211-241.compute-1.amazonaws.com
    (27 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

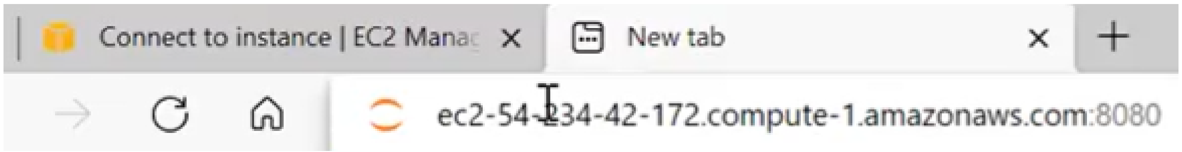


Downloads — ubuntu@ip-172-31-18-158: ~ — ssh -i navanti_key.pem ubu...

```
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.13.0-1022-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun May  8 14:59:12 UTC 2022

  System load:  0.44              Processes:             182
  Usage of /:   71.2% of 29.02GB   Users logged in:       0
  Memory usage: 0%                IPv4 address for ens3: 172.31.18.158
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

65 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


Last login: Fri May  6 14:42:34 2022 from 68.100.14.12
ubuntu@ip-172-31-18-158:~$ start-la
```

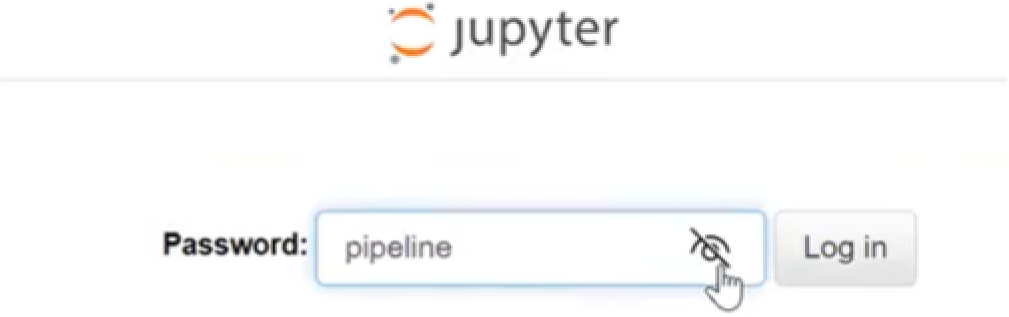
Part 3: Accessing JupyterLab Server via AWS

Step 9: Copy the public DNS from the AWS instance page and paste the link in a new internet browser ab, then add ":8080"



ec2-54-234-42-172.compute-1.amazonaws.com:8080

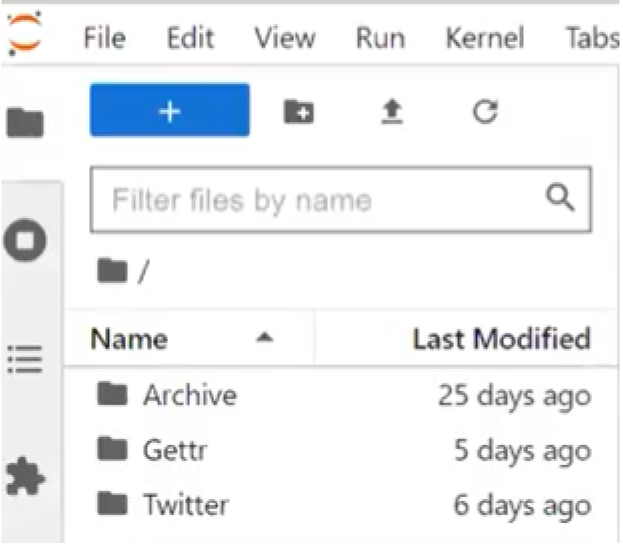
Step 10: Load the site and then type "pipeline" to log into the JupyterLab Server



Password: pipeline

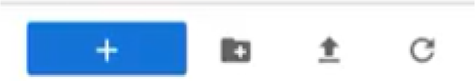
Part 4: Running the Bokeh Server to access the dashboard

Step 11: Once in the JupyterLab select the folder for the dashboard you intend on running (Select the Twitter folder for the Twitter dashboard or the Gettr folder for the Gettr dashboard)

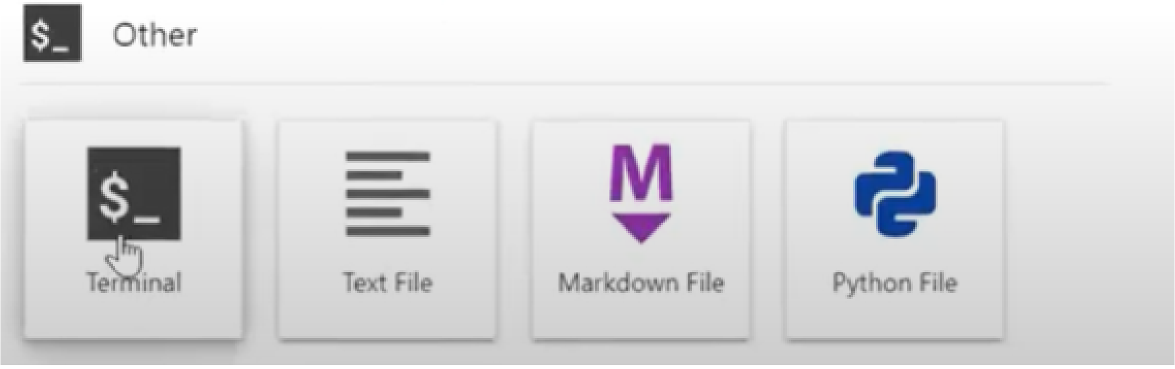


Name	Last Modified
Archive	25 days ago
Gettr	5 days ago
Twitter	6 days ago

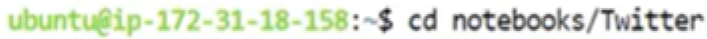
Step 12: After selecting a folder, click on the blue box with a plus sign in the upper left corner of JupyterLab to open a new tab



Step 13: Scroll down to “Other” and select “Terminal” to open a terminal window

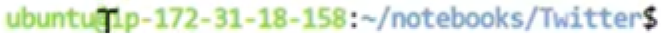


Step 14: Create a path in the terminal by typing “cd notebooks/Twitter”, and then hit enter (If you plan on using the Gettr dashboard type “cd notebooks/Gettr” within the Gettr folder)



```
ubuntu@ip-172-31-18-158:~$ cd notebooks/Twitter
```

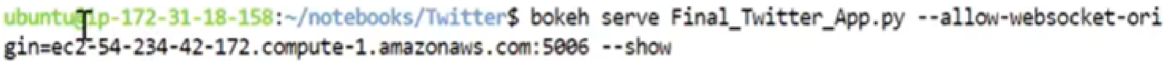
Step 15: Make sure the path previously typed is now visible in blue font in the preceding line



```
ubuntu@ip-172-31-18-158:~/notebooks/Twitter$
```

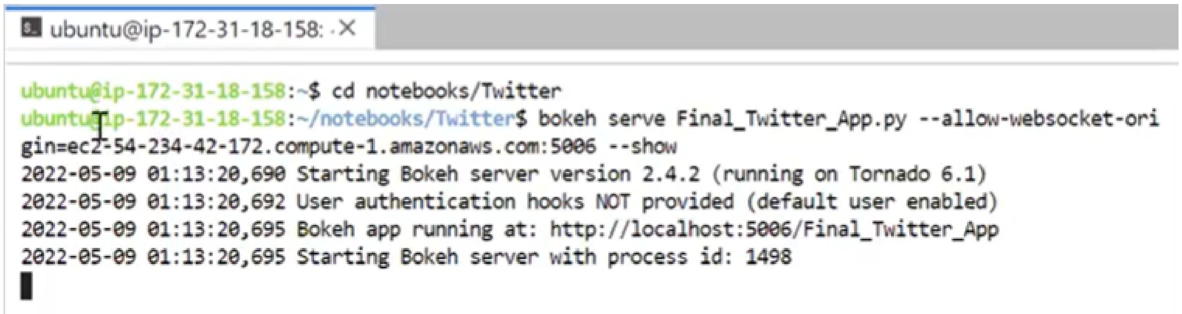
Step 16: Type “bokeh serve Final_Twitter_App.py --allow-websocket-origin=*COPY & PASTE EC2 INSTANCE NUMBER FROM PUBLIC DNS.*compute-1.amazonaws.com:5006 --show”

- ec2 instance number will change every time you load a new instance, but should look similar to this “ec2-54-234-42-172” as an example



```
ubuntu@ip-172-31-18-158:~/notebooks/Twitter$ bokeh serve Final_Twitter_App.py --allow-websocket-ori
gin=ec2-54-234-42-172.compute-1.amazonaws.com:5006 --show
```

Step 17: Hit enter after typing the previous line, and you should see the following output:

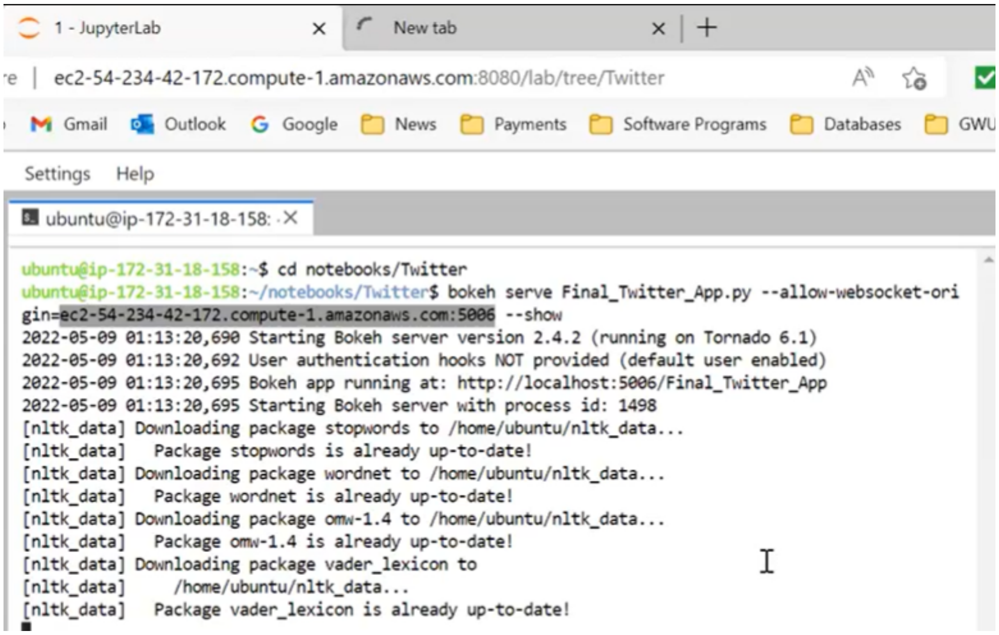


```
ubuntu@ip-172-31-18-158:~$ cd notebooks/Twitter
ubuntu@ip-172-31-18-158:~/notebooks/Twitter$ bokeh serve Final_Twitter_App.py --allow-websocket-ori
gin=ec2-54-234-42-172.compute-1.amazonaws.com:5006 --show
2022-05-09 01:13:20,690 Starting Bokeh server version 2.4.2 (running on Tornado 6.1)
2022-05-09 01:13:20,692 User authentication hooks NOT provided (default user enabled)
2022-05-09 01:13:20,695 Bokeh app running at: http://localhost:5006/Final_Twitter_App
2022-05-09 01:13:20,695 Starting Bokeh server with process id: 1498
▌
```

Step 18: Open a new tab in your internet browser

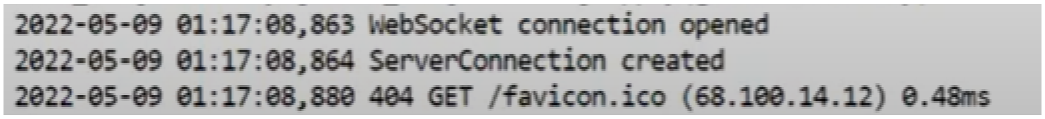
Step 19: Copy and paste the ec2 instance number from step 16 followed by typing ".compute-1.amazonaws.com:5006/Final_Twitter_App"

Step 20: Hit enter and then return to the terminal tab where you should see various lines loading in the terminal indicating that the dashboard is loading (Will take approximately 2-4 min)



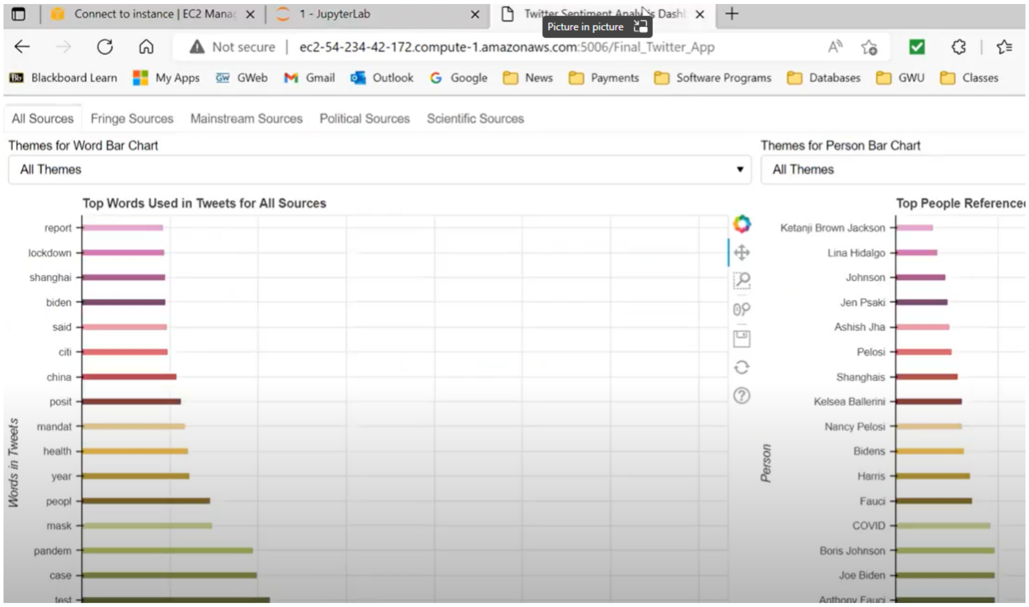
```
ubuntu@ip-172-31-18-158:~$ cd notebooks/Twitter
ubuntu@ip-172-31-18-158:~/notebooks/Twitter$ bokeh serve Final_Twitter_App.py --allow-websocket-ori
gin=ec2-54-234-42-172.compute-1.amazonaws.com:5006 --show
2022-05-09 01:13:20,690 Starting Bokeh server version 2.4.2 (running on Tornado 6.1)
2022-05-09 01:13:20,692 User authentication hooks NOT provided (default user enabled)
2022-05-09 01:13:20,695 Bokeh app running at: http://localhost:5006/Final_Twitter_App
2022-05-09 01:13:20,695 Starting Bokeh server with process id: 1498
[nltk_data] Downloading package stopwords to /home/ubuntu/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /home/ubuntu/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /home/ubuntu/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     /home/ubuntu/nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

Step 21: Once the dashboard is done loading, you should receive the following output on the terminal:



```
2022-05-09 01:17:08,863 WebSocket connection opened
2022-05-09 01:17:08,864 ServerConnection created
2022-05-09 01:17:08,880 404 GET /favicon.ico (68.100.14.12) 0.48ms
```

Step 22: Select the new tab you created in Step 18 to access the fully loaded dashboard

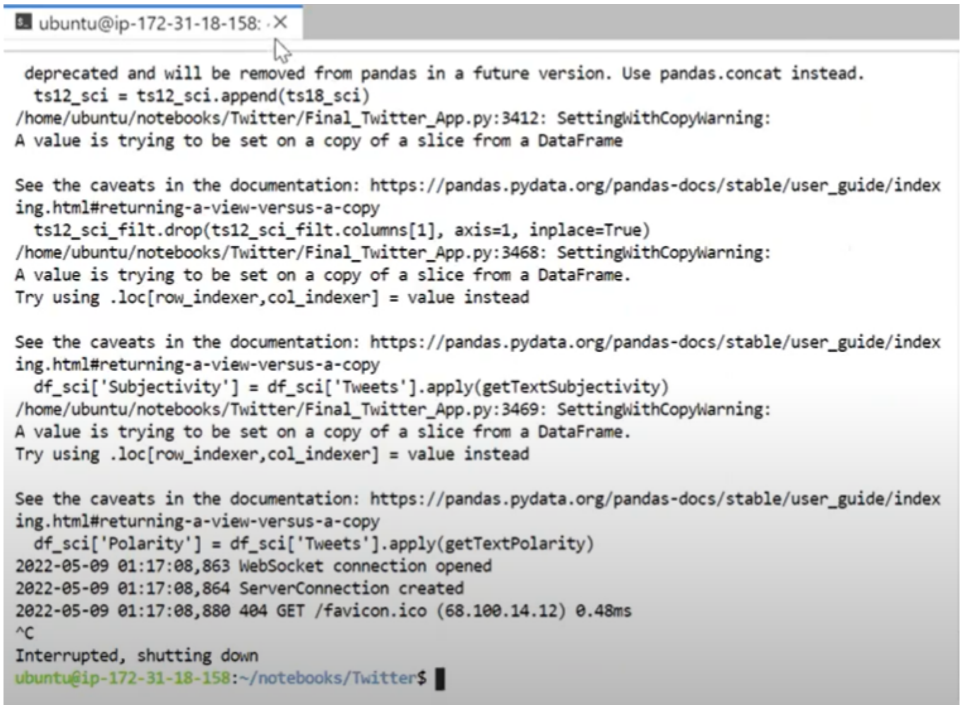


Themes for Word Bar Chart: All Themes

Themes for Person Bar Chart: All Themes

Part 5: Shutting Down the Bokeh Server

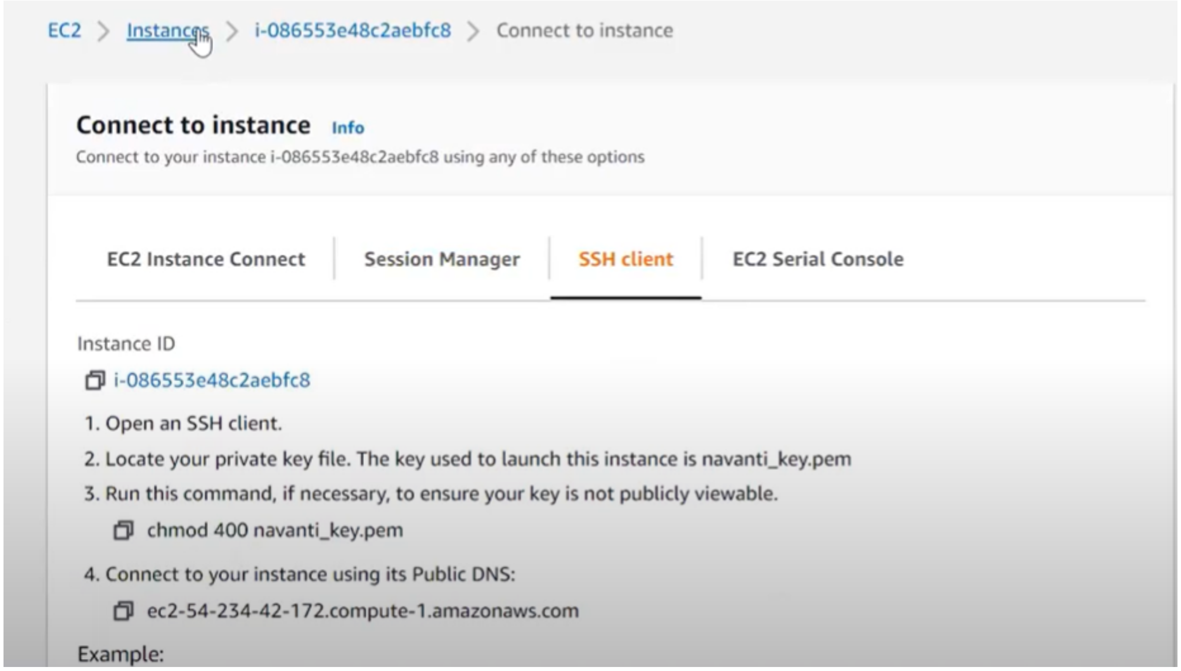
Step 23: Return to the terminal tab and scroll down to the last line, and hit CTRL+C to shut down the terminal



```
 deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  ts12_sci = ts12_sci.append(ts18_sci)
/home/ubuntu/notebooks/Twitter/Final_Twitter_App.py:3412: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/index
ing.html#returning-a-view-versus-a-copy
  ts12_sci_filt.drop(ts12_sci_filt.columns[1], axis=1, inplace=True)
/home/ubuntu/notebooks/Twitter/Final_Twitter_App.py:3468: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/index
ing.html#returning-a-view-versus-a-copy
  df_sci['Subjectivity'] = df_sci['Tweets'].apply(getTextSubjectivity)
/home/ubuntu/notebooks/Twitter/Final_Twitter_App.py:3469: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/index
ing.html#returning-a-view-versus-a-copy
  df_sci['Polarity'] = df_sci['Tweets'].apply(getTextPolarity)
2022-05-09 01:17:08,863 WebSocket connection opened
2022-05-09 01:17:08,864 ServerConnection created
2022-05-09 01:17:08,880 404 GET /favicon.ico (68.100.14.12) 0.48ms
^C
Interrupted, shutting down
ubuntu@ip-172-31-18-158:~/notebooks/Twitter$
```

Part 6: Shutting down the AWS Instance and Logging out of AWS

Step 24: Return to the AWS instance page, and select “instances”



EC2 > Instances > i-086553e48c2aebfc8 > Connect to instance

Connect to instance Info

Connect to your instance i-086553e48c2aebfc8 using any of these options

Instance ID

i-086553e48c2aebfc8

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is navanti_key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.

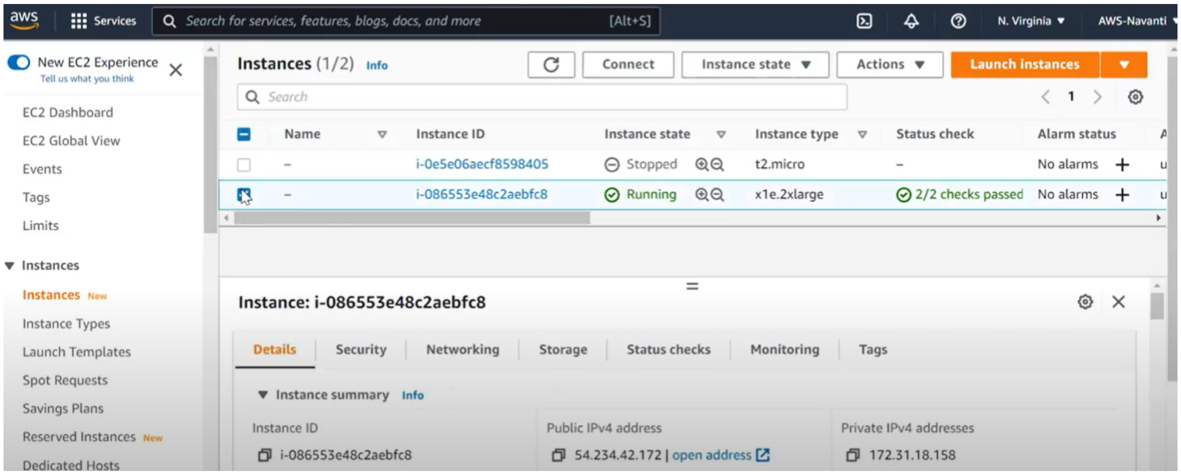
```
chmod 400 navanti_key.pem
```

4. Connect to your instance using its Public DNS:

```
ec2-54-234-42-172.compute-1.amazonaws.com
```

Example:

Step 25: Select the running instance



Instances (1/2) Info

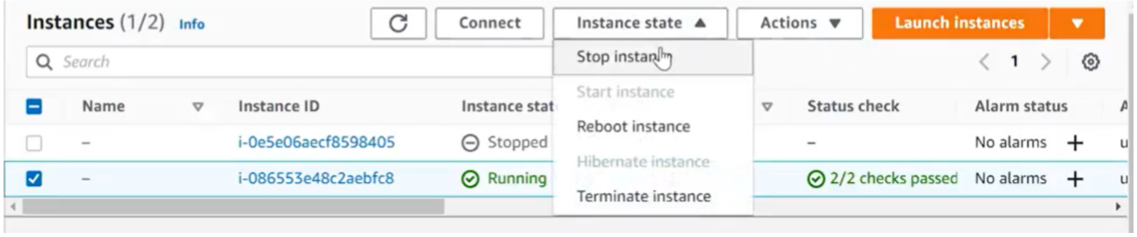
☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
☐	–	i-0e5e06aecf8598405	Stopped	t2.micro	–	No alarms
☑	–	i-086553e48c2aebfc8	Running	x1e.2xlarge	2/2 checks passed	No alarms

Instance: i-086553e48c2aebfc8

Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-086553e48c2aebfc8	54.234.42.172 | open address	172.31.18.158

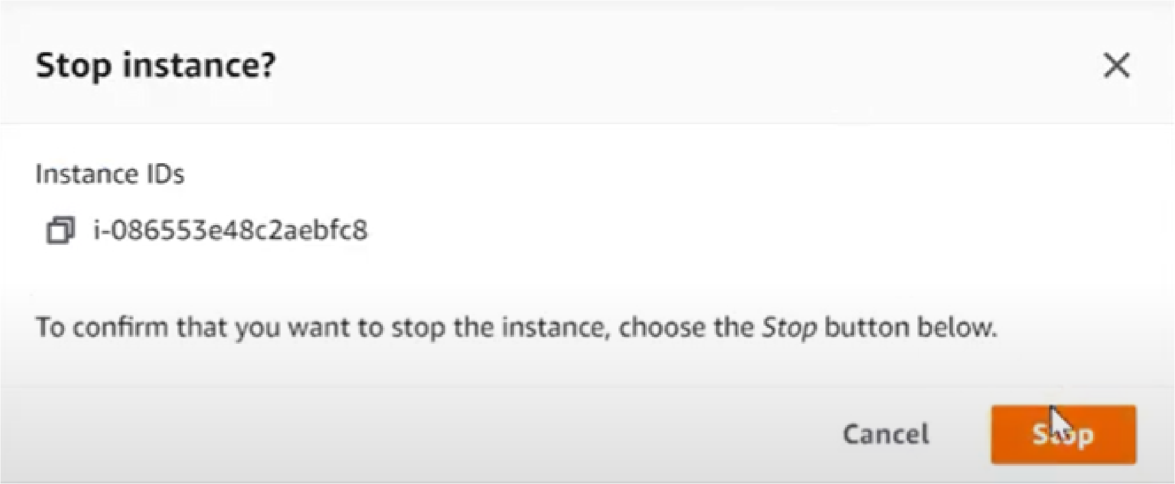
Step 26: Click on the instance state dropdown and select “stop instance” (*DO NOT SELECT “TERMINATE INSTANCE”*)



Instances (1/2) Info

☐	Name	Instance ID	Instance stat	Status check	Alarm status
☐	–	i-0e5e06aecf8598405	Stopped	–	No alarms
☑	–	i-086553e48c2aebfc8	Running	2/2 checks passed	No alarms

Step 27: Stop instance confirmation will pop out and then click on stop



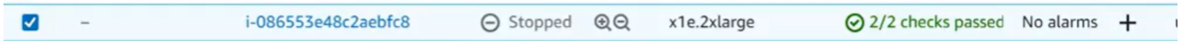
Stop instance?

Instance IDs

i-086553e48c2aebfc8

To confirm that you want to stop the instance, choose the *Stop* button below.

Step 28: Wait until instance state changes from “Stopping” to “Stopped” (May have to click on the refresh button next to the “Connect” button to update the instance state)



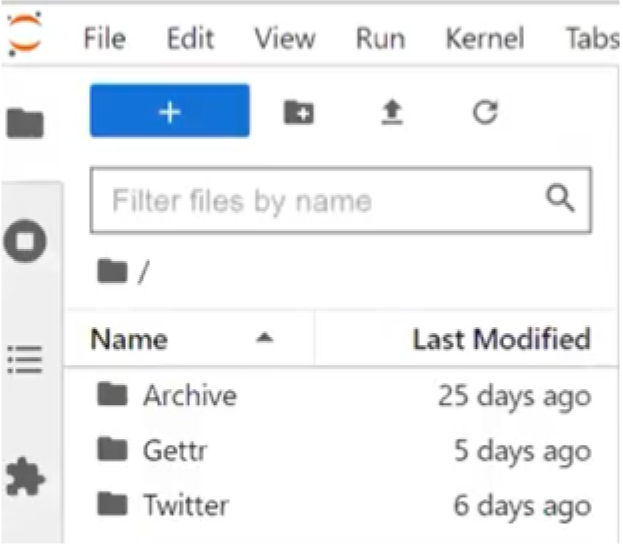
☑	–	i-086553e48c2aebfc8	⊖ Stopped	x1e.2xlarge	✓ 2/2 checks passed	No alarms +

Step 29: Log out of AWS after confirming that the instance has been stopped

Additional: Running the database

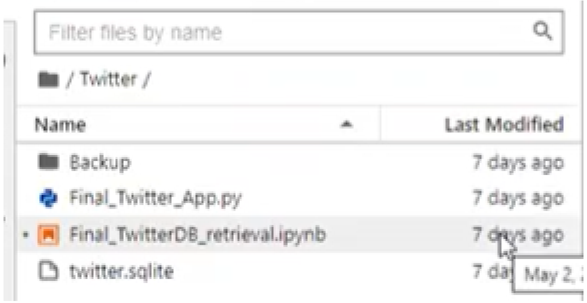
This section proceeds Step 9 in Part 3

Step 1: Once in JupyterLab select the folder for the dashboard you intend on running (Select the Twitter folder for the Twitter database or the Gettr folder for the Gettr database)



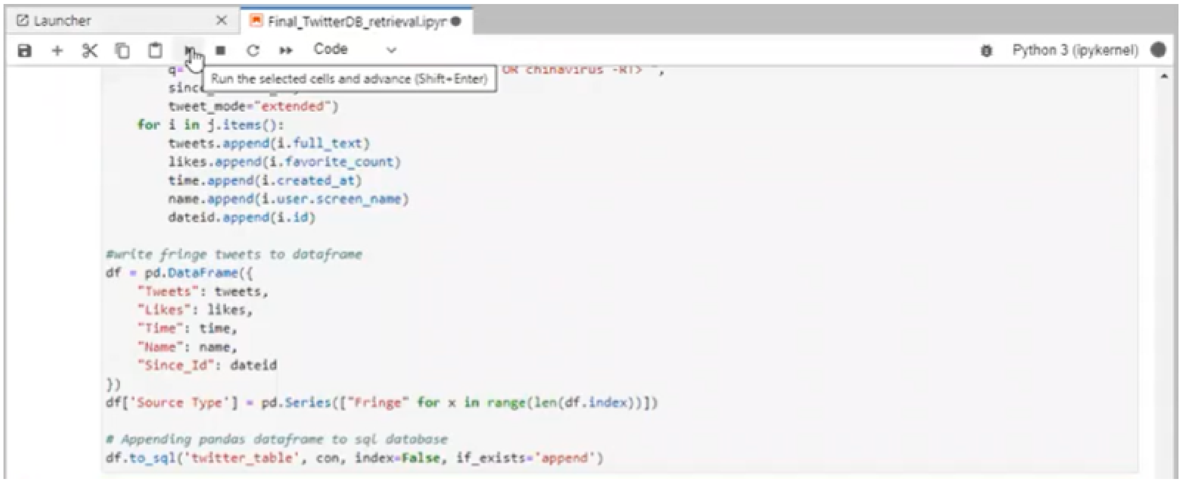
Name	Last Modified
Archive	25 days ago
Gettr	5 days ago
Twitter	6 days ago

Step 2: Select the notebook titled “Final_TwitterDB_retrieval.ipynb” and open it



Name	Last Modified
Backup	7 days ago
Final_Twitter_App.py	7 days ago
Final_TwitterDB_retrieval.ipynb	7 days ago
twitter.sqlite	7 da

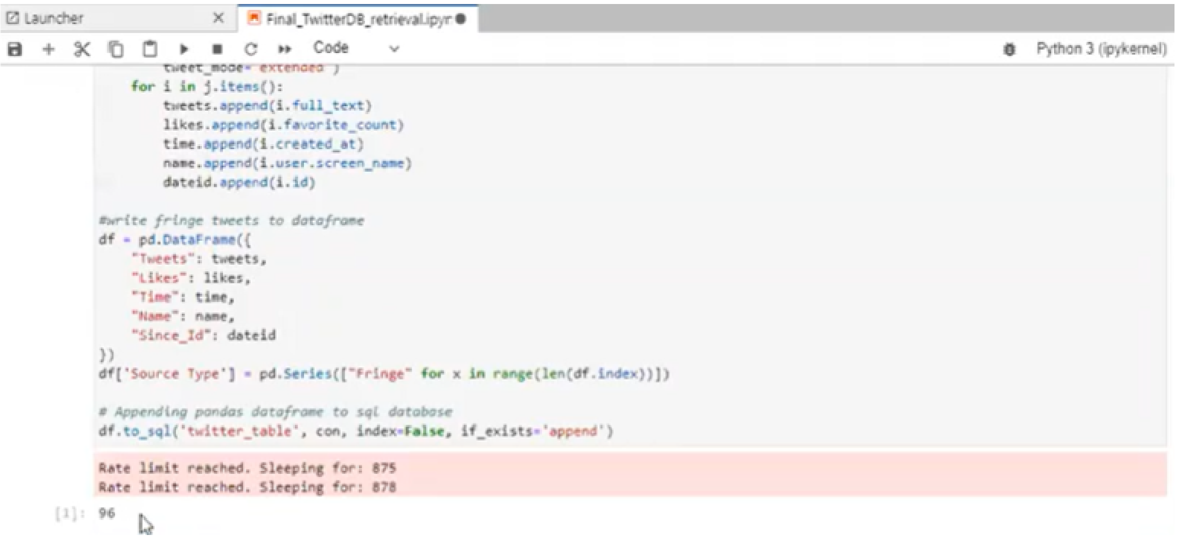
Step 3: Select the play button on the top of the “Final_TwitterDB_retrieval.ipynb” notebook tab (This will take anywhere from 5 -20 minutes depending on how frequently you run the notebook)


Launcher
Final_TwitterDB_retrieval.ipyr
Code
Python 3 (ipykernel)
Run the selected cells and advance (Shift+Enter)

```
        q=                                       OR chinavirus -RT> ",
        sinc
        tweet_mode="extended")
    for i in j.items():
        tweets.append(i.full_text)
        likes.append(i.favorite_count)
        time.append(i.created_at)
        name.append(i.user.screen_name)
        dateid.append(i.id)

#write fringe tweets to dataframe
df = pd.DataFrame({
    "Tweets": tweets,
    "Likes": likes,
    "Time": time,
    "Name": name,
    "Since_Id": dateid
})
df['Source Type'] = pd.Series(["Fringe" for x in range(len(df.index))])

# Appending pandas dataframe to sql database
df.to_sql('twitter_table', con, index=False, if_exists='append')
```

Step 4: Once the code in the notebook ran you should have a number at the bottom of the notebook (Screenshot 1) and a line number at the top of the notebook (Screenshot 2) indicating that new data has been added to the database

Screenshot 1

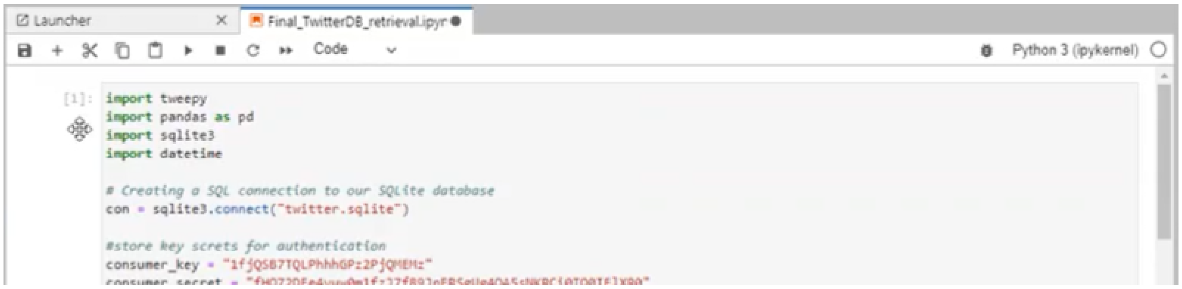


```
        tweet_mode='extended')
    for i in j.items():
        tweets.append(i.full_text)
        likes.append(i.favorite_count)
        time.append(i.created_at)
        name.append(i.user.screen_name)
        dateid.append(i.id)

#write fringe tweets to dataframe
df = pd.DataFrame({
    "Tweets": tweets,
    "Likes": likes,
    "Time": time,
    "Name": name,
    "Since_Id": dateid
})
df['Source Type'] = pd.Series(["Fringe" for x in range(len(df.index))])

# Appending pandas dataframe to sql database
df.to_sql('twitter_table', con, index=False, if_exists='append')
```

```
Rate limit reached. Sleeping for: 875
Rate limit reached. Sleeping for: 878
```

```
[1]: 96
```

Screenshot 2



```
[1]: import tweepy
     import pandas as pd
     import sqlite3
     import datetime

     # Creating a SQL connection to our SQLite database
     con = sqlite3.connect("twitter.sqlite")

     #store key screts for authentication
     consumer_key = "1fjQSB7TQLPhhhGPz2PjQMEMz"
     consumer_secret = "fHO72DFe4vuwOm1fz27f89JoFRSeUe4OAS5NKRCj0TQ0IFlXR0"
```

Step 5: Close the notebook once you are done and follow Step 24 in Part 6 to shut down the AWS instance

Notes for Databases:

- *DO NOT RUN THE* “Final_GettrDB_retrieval.ipynb” *MORE THAN ONCE WITHIN A 24 HOUR PERIOD*
- Best approach would be to run the “Final_TwitterDB_retrieval.ipynb” and “Final_GettrDB_retrieval.ipynb” notebooks once every week