MSDS610 Data Engineering

Week 5 Lab Supplemental – Part 1

This is a supplemental document that supports the MSDS610 Week 5 lab.

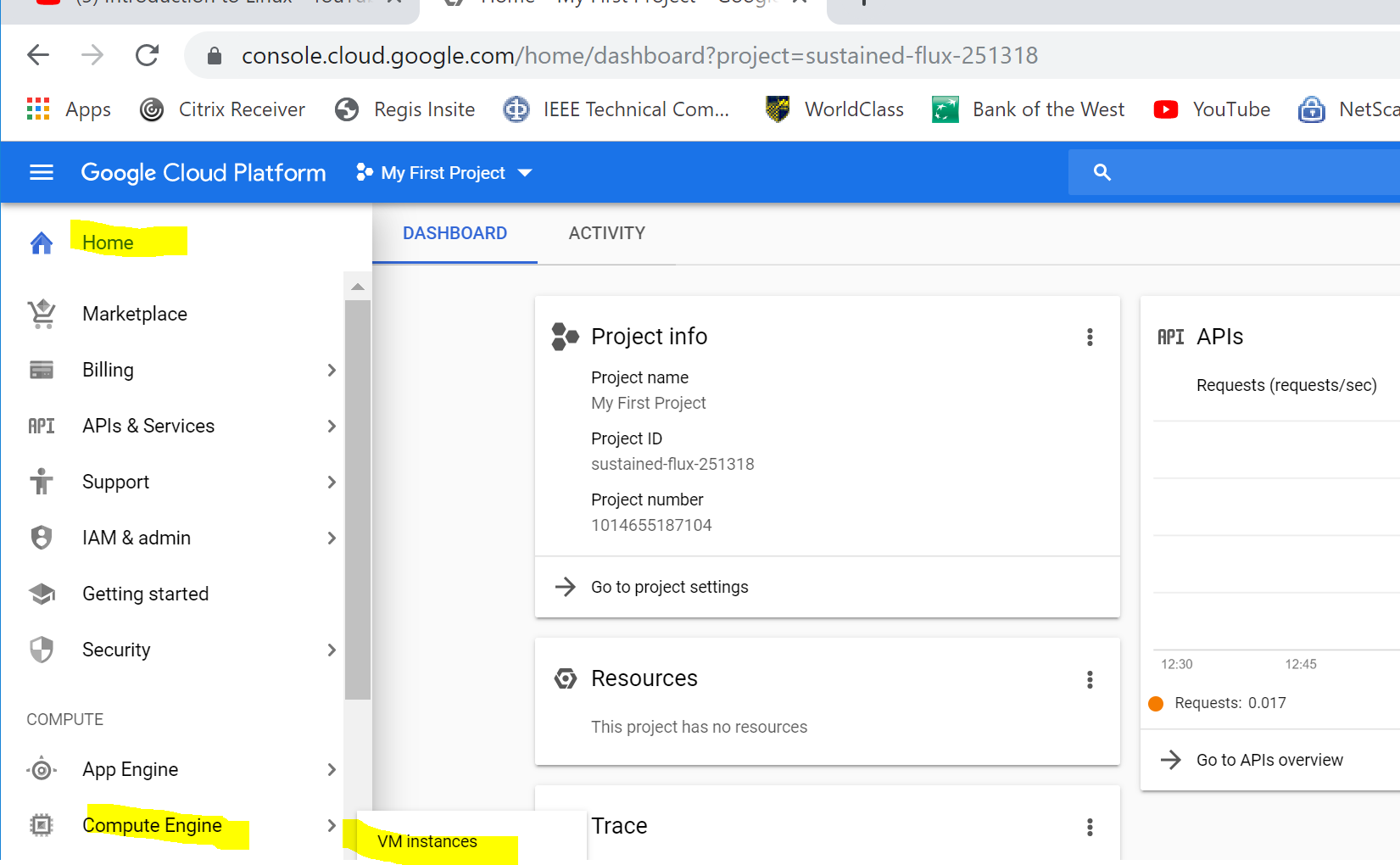
Created by Dr. Bob Mason on 9/26/2019 in support of Dr. Nate George’s Week 5 lab video.

**Based on this youtube video by Dr. George (Nate), please follow the steps shown below.**

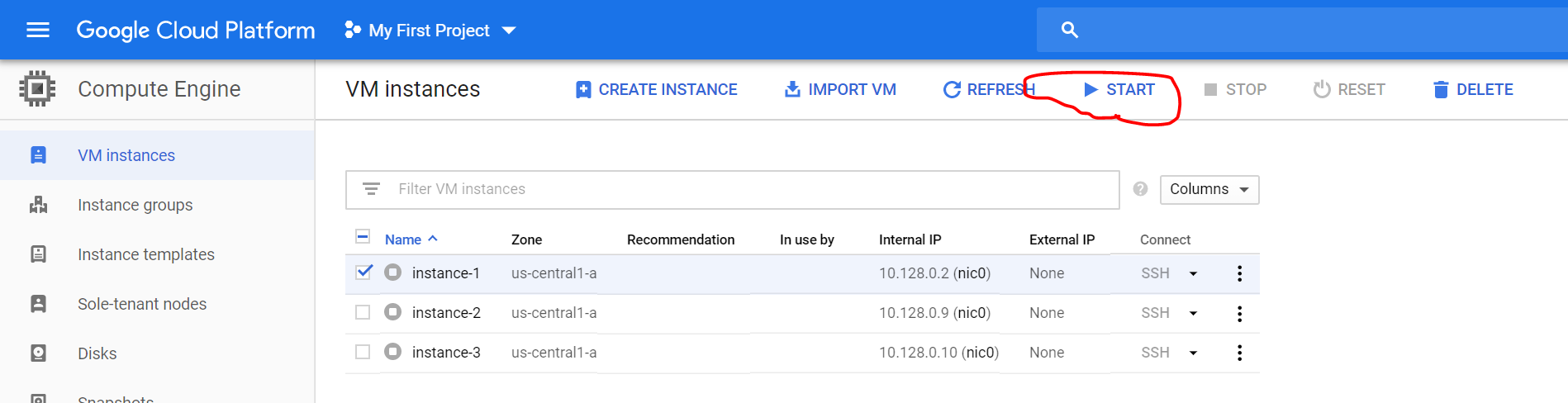
<https://www.youtube.com/watch?v=tRruqJhjgFQ&feature=youtu.be>

**Login to the Google Cloud and to the console page. You have the option of using Ubuntu in a VM or other Linux terminal that you prefer to use. This approach will use the VM we created previously in Lab 1. Whatever works for you is okay.**

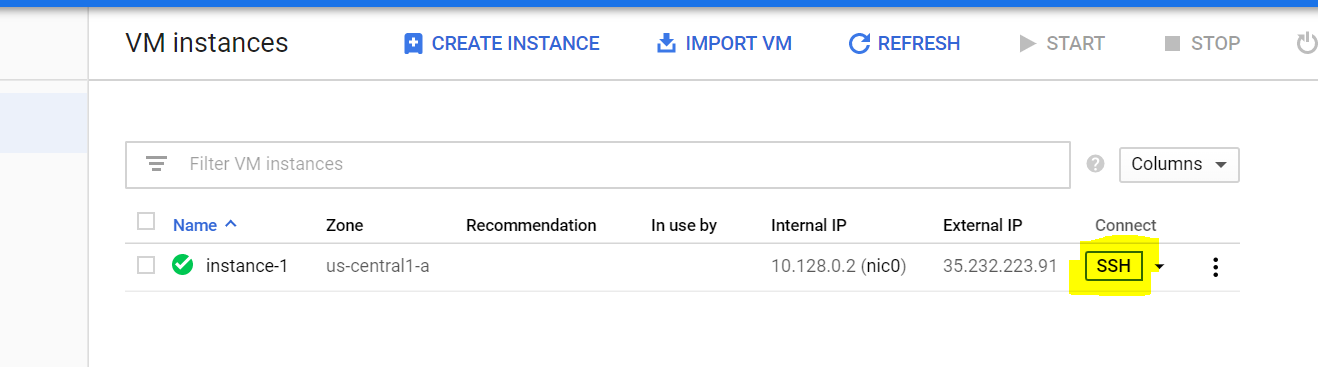
**Use the VM Instance from Lab 1, select** **Compute Engine, VM Instances**



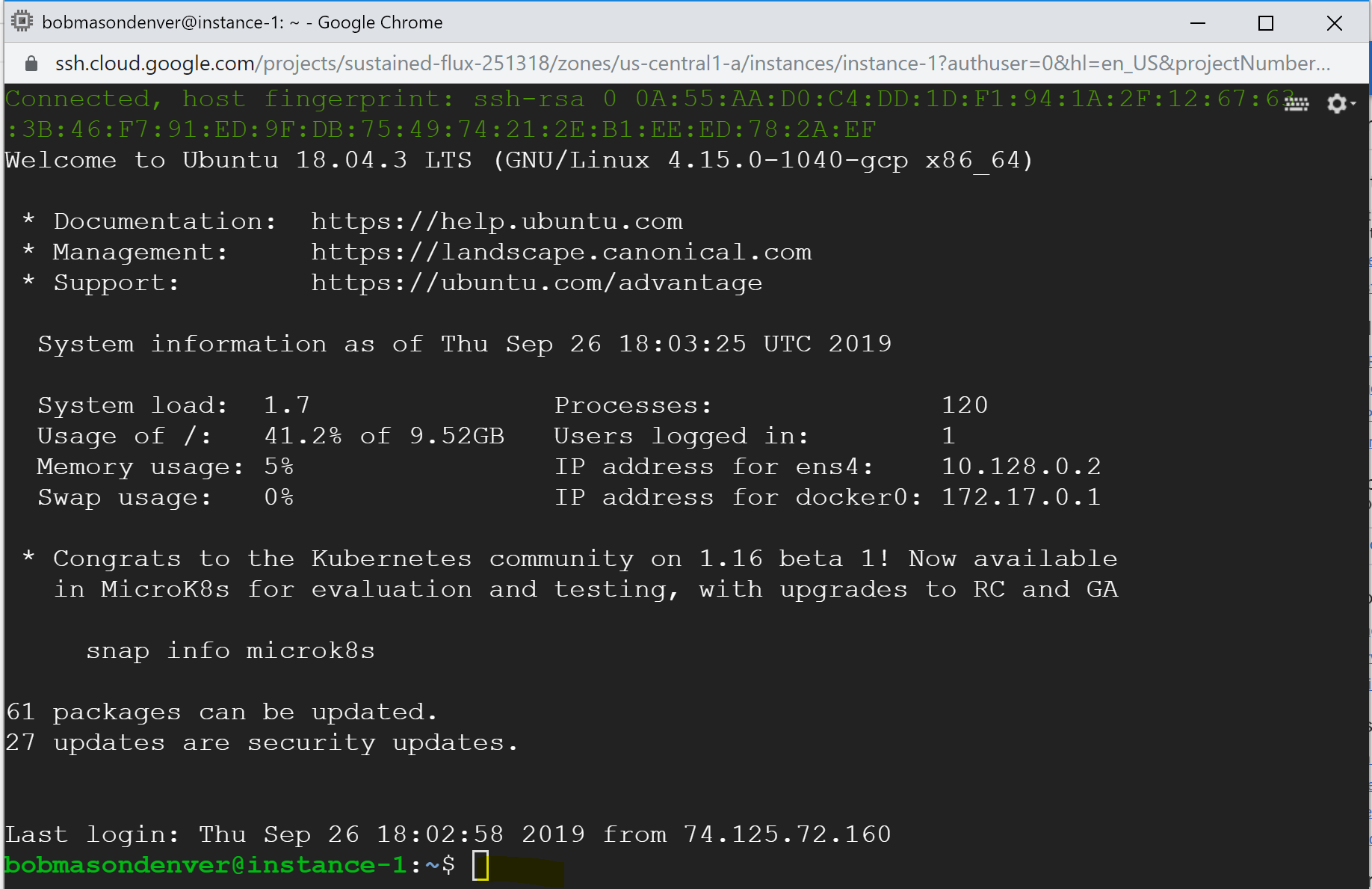
**Select your VM instance (somehow along the way I created three instance). In this case, I’m selecting the first Instance-1 and clicking on the start button to get it running.**



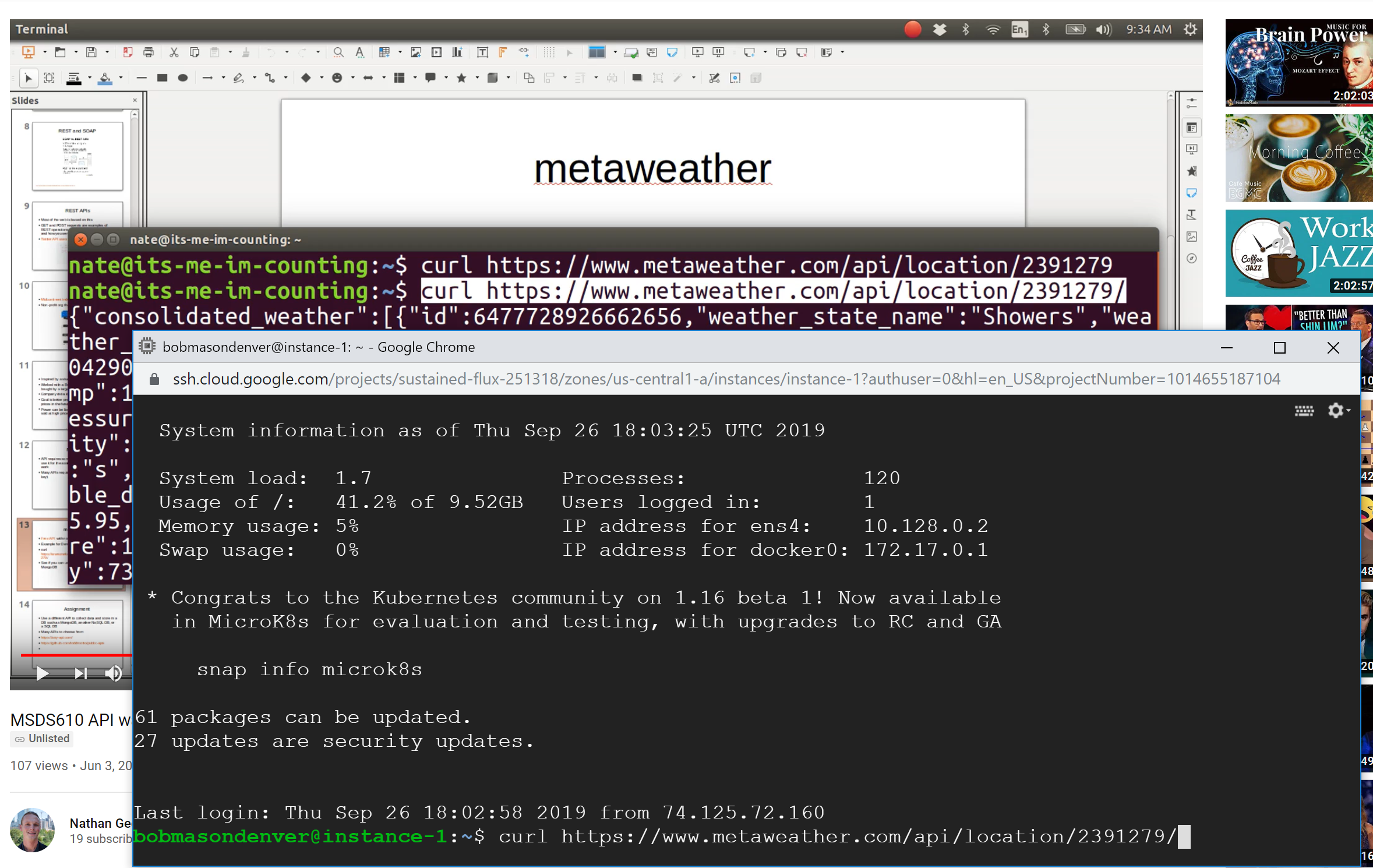
**Click on the SSH Button to open a terminal.**



Your new SSH terminal should appear as the following:

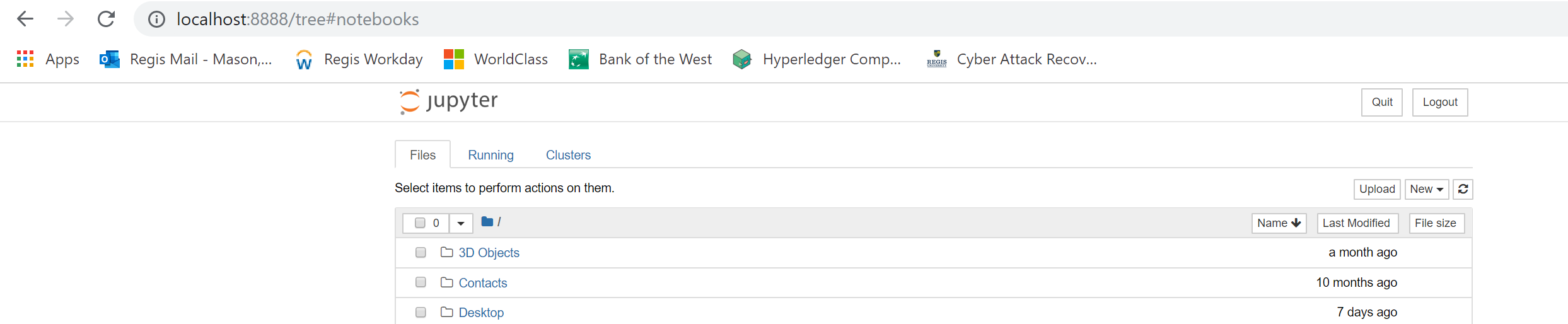


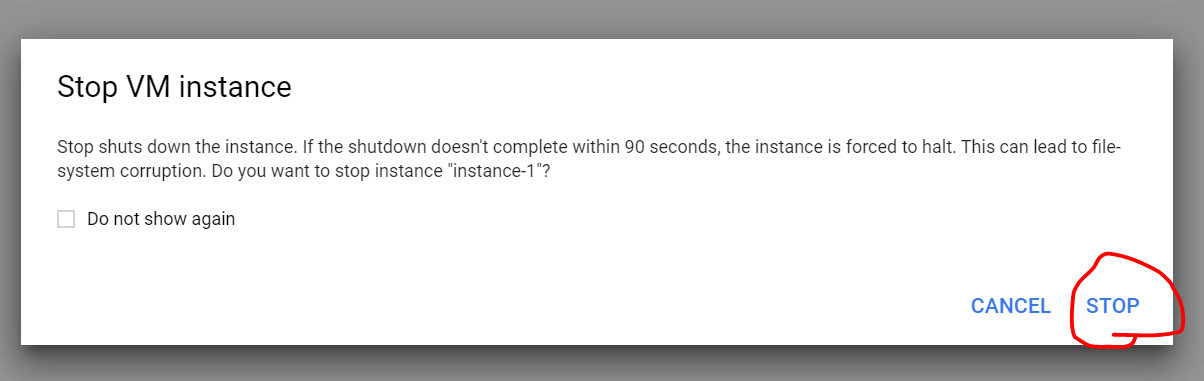
**Now we can test the API to the meta weather website using the Curl command that Nate demonstrates in the Week 5 video.**



**As mentioned on the video, the api works and returns a lot of data about Denver weather. So far, so good.**

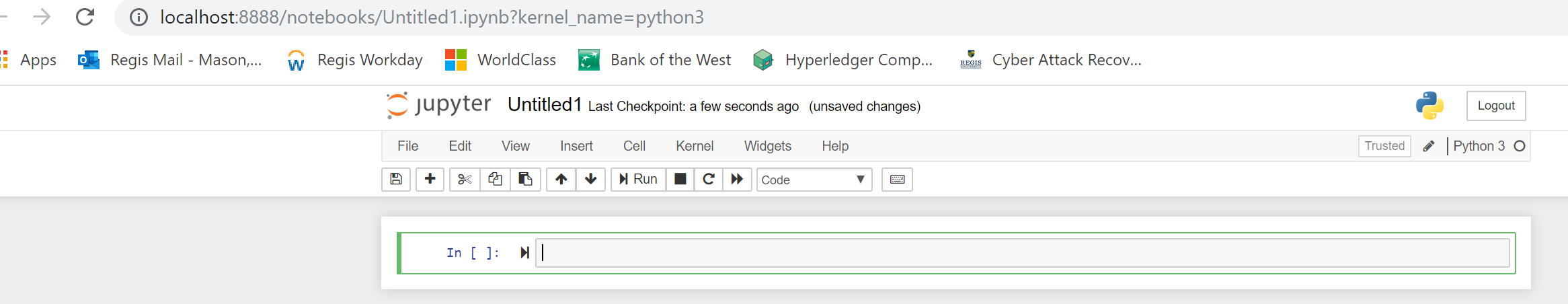


**At this point, Nate switches to using a Jupyter Notebooks Python Program. I’d recommend that you use the Jupyter Notebooks installation that you did in a prior lab. In my example, I installed under windows. You may have installed in a Ubuntu instance. Whatever you did is okay**



**I’m going to continue with my Windows installation of Anaconda3 Jupyter Notebooks and I’m going to create a new notebook for this lab. Remember to STOP your VM as shown above.**

**You can include all of your code in one large cell or spread out the code over multiple cells as Nate does in the video. Below I decided to put all of my code into one large cell.**



**I entered the following python program into one cell of the notebook and then ran it.**

*import requests as req*

*response = req.get('https://www.metaweather.com/api/location/2391279/')*

*response*

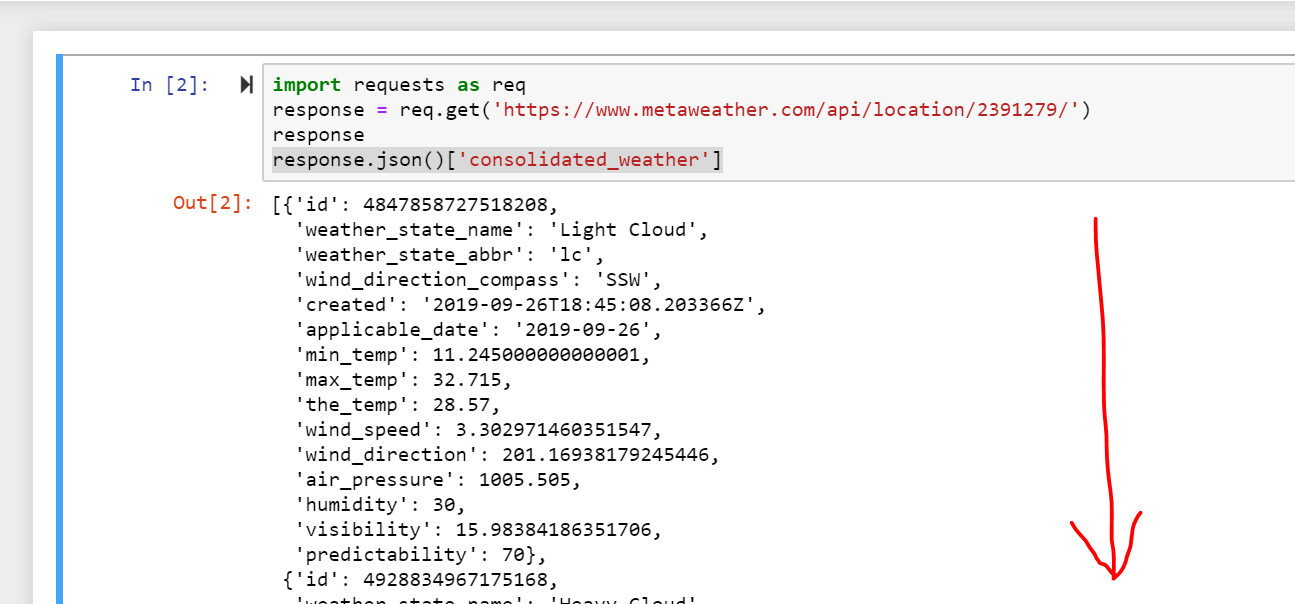
*response.json()*



**Note: the dates in Jason format data will vary depending on when you run the program.**

**Add consolidated weather to the response.json() and run the program again.**

*response.json()['consolidated\_weather']*



**Import pandas which is add on package. Add a Panda Data Frame object called df that uses the json normalize and then display the head (top).**

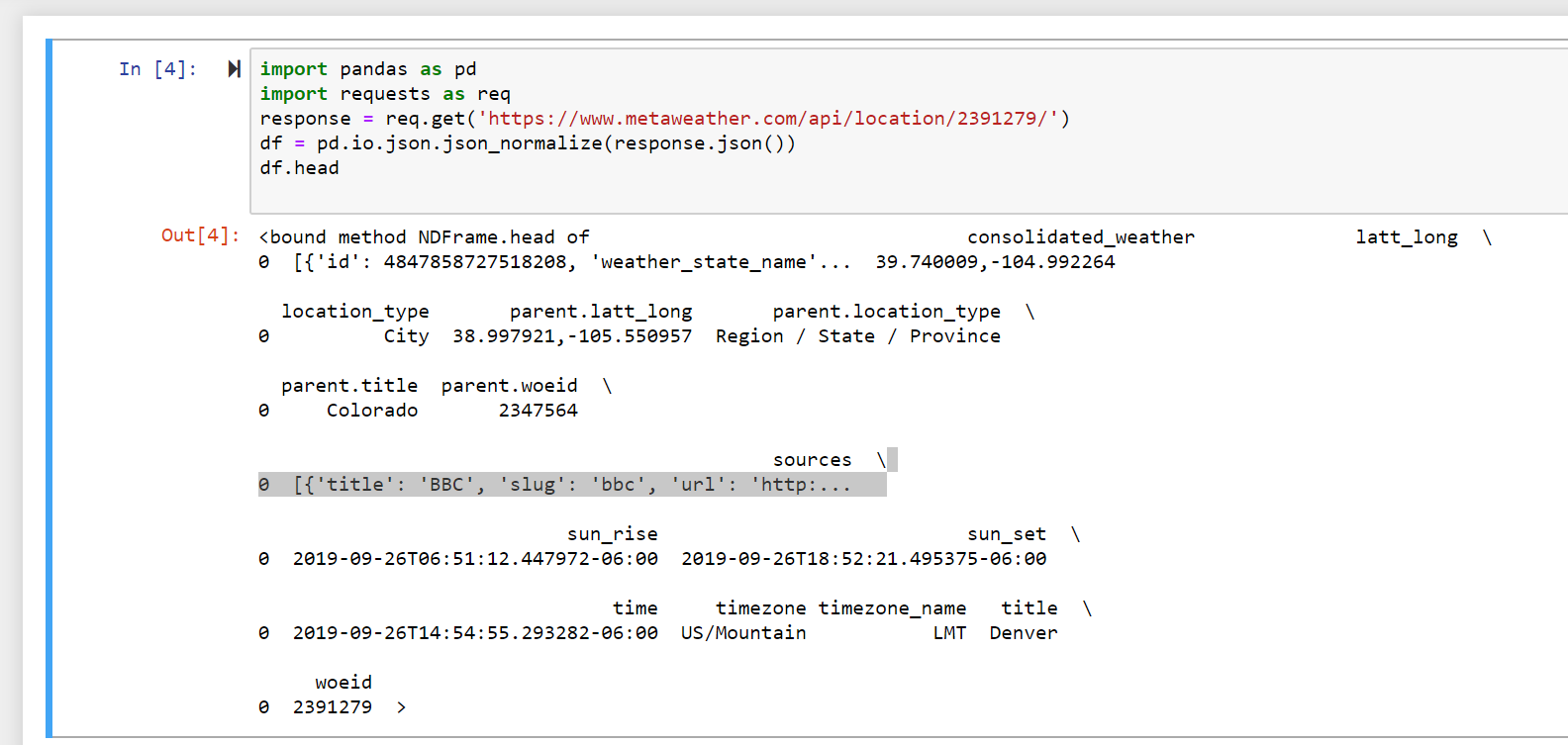
*import pandas as pd*

*import requests as req*

*response = req.get('https://www.metaweather.com/api/location/2391279/')*

*df = pd.io.json.json\_normalize(response.json())*

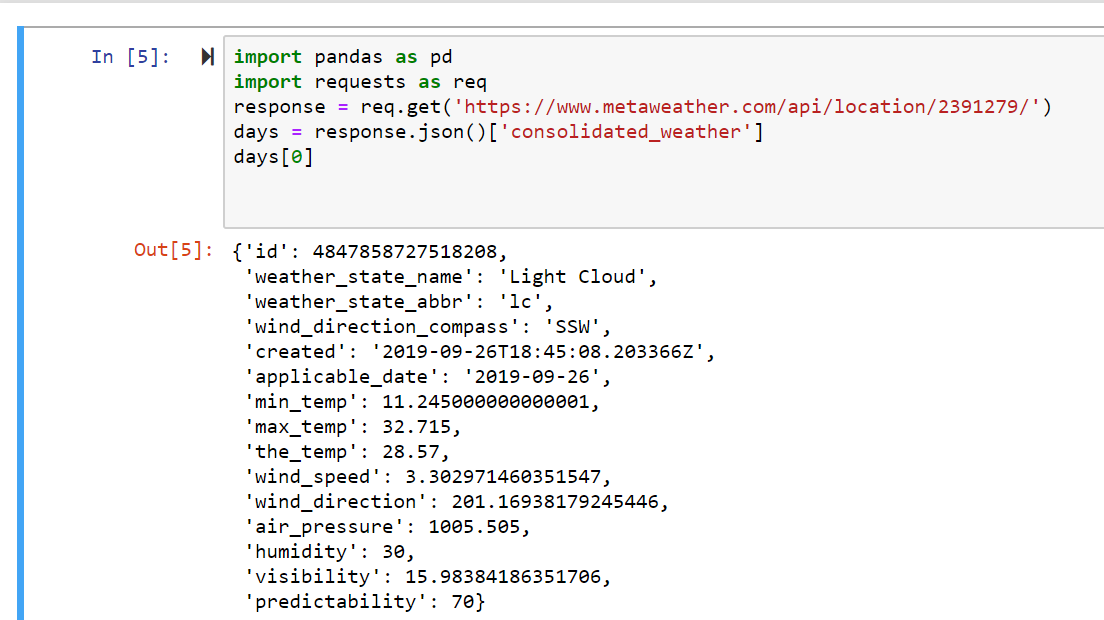
*df.head*



**Then we will change the python program to include a new array called days. The first entry in the array shows the first day, to reference the second day we would use [1] since the array is zero relative.**

*days = response.json()['consolidated\_weather']*

*days[0]*



**We will continue to embellish the program.****This is the current code of the program after more changes. When you run, you will see the results shown below.**

*import pandas as pd*

*import requests as req*

*response = req.get('https://www.metaweather.com/api/location/2391279/')*

*days = response.json()['consolidated\_weather']*

*df = pd.io.json.json\_normalize(days[0])*

*df.head*



**Now we will add a for loop to the program that loops thru the days array, appending each day to the date frame. Remember to indent for Python.**

*import pandas as pd*

*import requests as req*

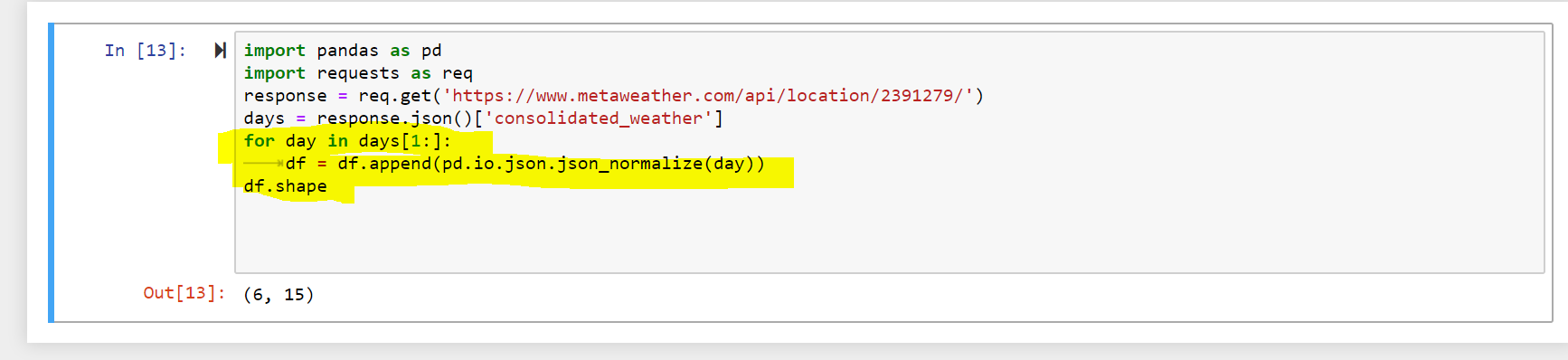
*response = req.get('https://www.metaweather.com/api/location/2391279/')*

*days = response.json()['consolidated\_weather']*

*for day in days[1:]:*

*df = df.append(pd.io.json.json\_normalize(day))*

*df.shape*



**The Panda shape command will show the dimensions of the data frame at the bottom which is 6 x 15.**

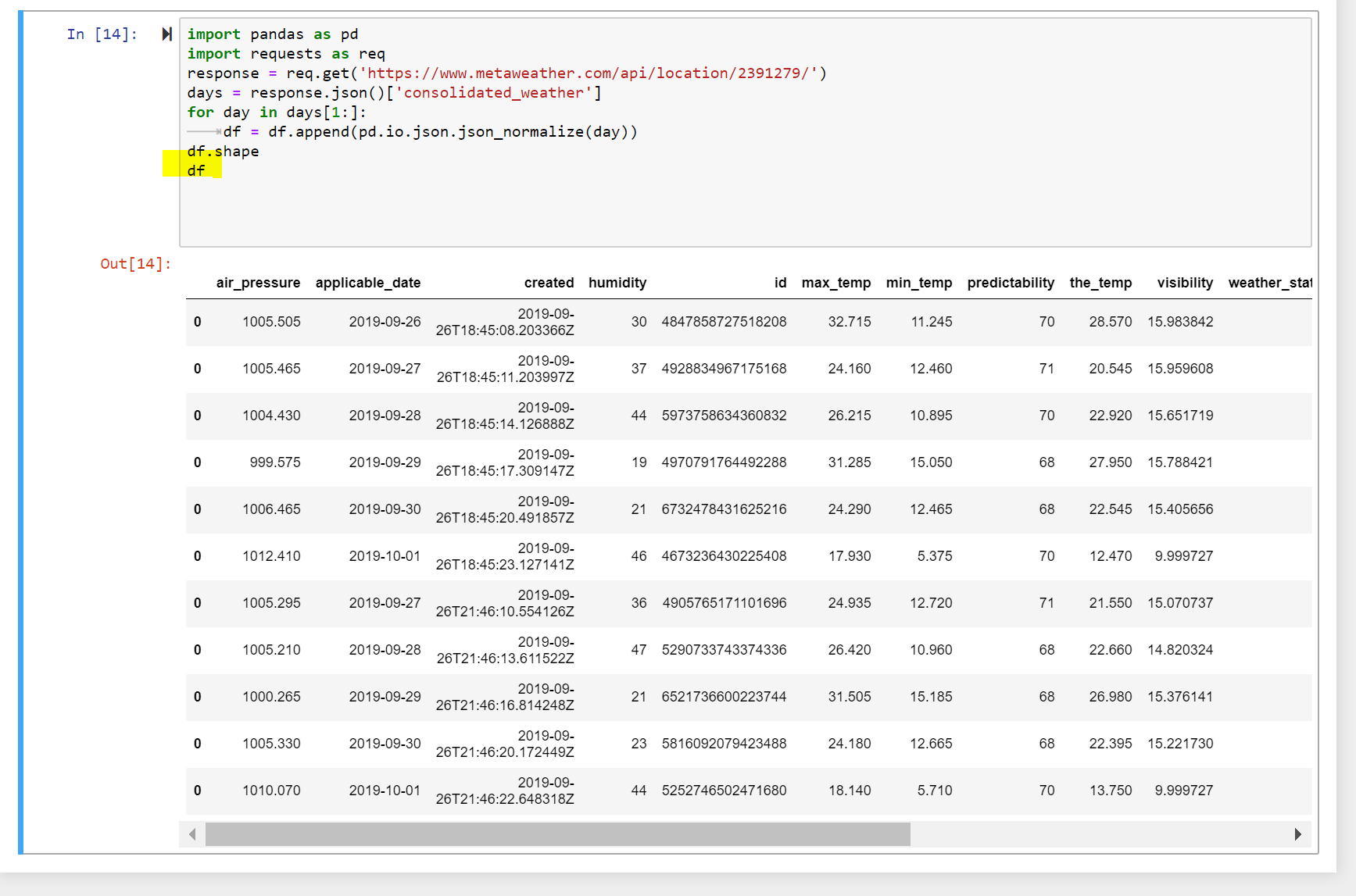
pandas.DataFrame.shape

DataFrame.**shape**

Return a tuple representing the dimensionality of the DataFrame.

---------------------------------------------------------------------------------------------------------

**If we add a df command, python will display the Data Frame in a tabular format. (Awesome)**



**We can now add a df.info()**

**Pandas dataframe.info() function is used to get a concise summary of the dataframe. It comes really handy when doing exploratory analysis of the data.**

*import pandas as pd*

*import requests as req*

*response = req.get('https://www.metaweather.com/api/location/2391279/')*

*days = response.json()['consolidated\_weather']*

*for day in days[1:]:*

*df = df.append(pd.io.json.json\_normalize(day))*

*df.info()*



**This is end of Part 1. Please continue to Part 2 of Lab 5.**