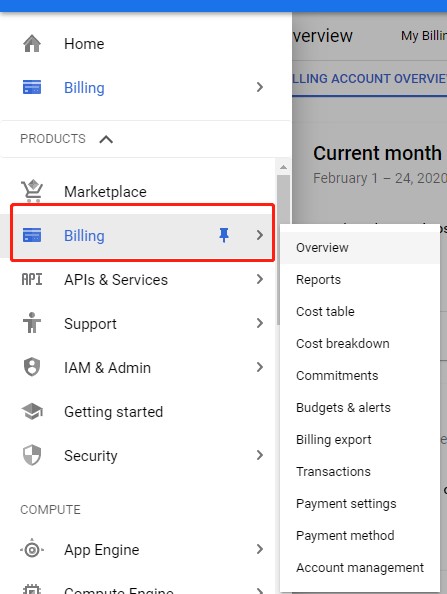
**Lab 6 - Jupyter Notebook and Google Cloud Storage**

# Step 1: Create a free account in Google Cloud with 300$ credit

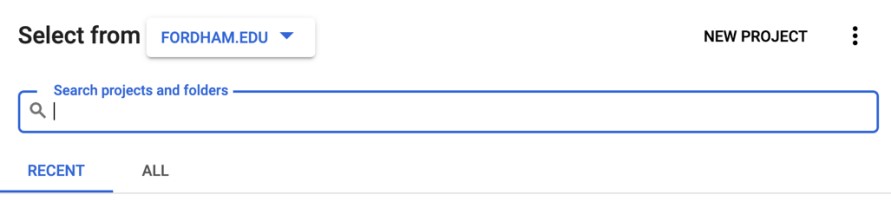
For this step, you will have to put your payment information and verify your account. If you followed and finished Lab 5, you are fine to skip this and jump to step 2.

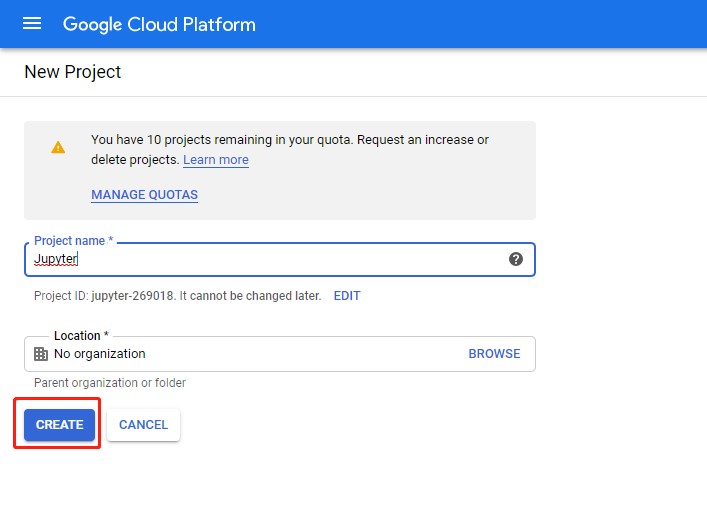
If you are first time using Google Cloud, you have to link a credit card to your Google Cloud account to receive the free $300 credit.



# Step 2: Create a new project

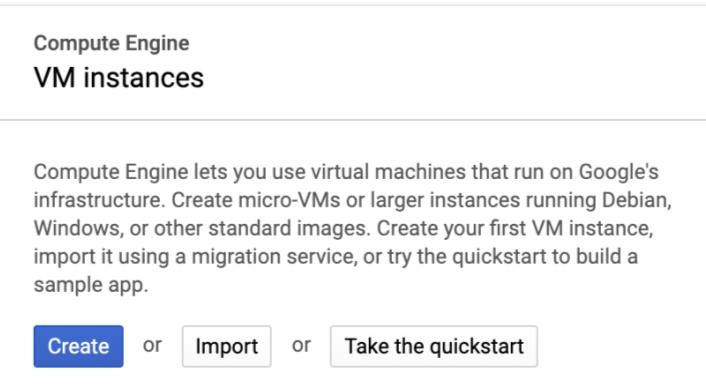
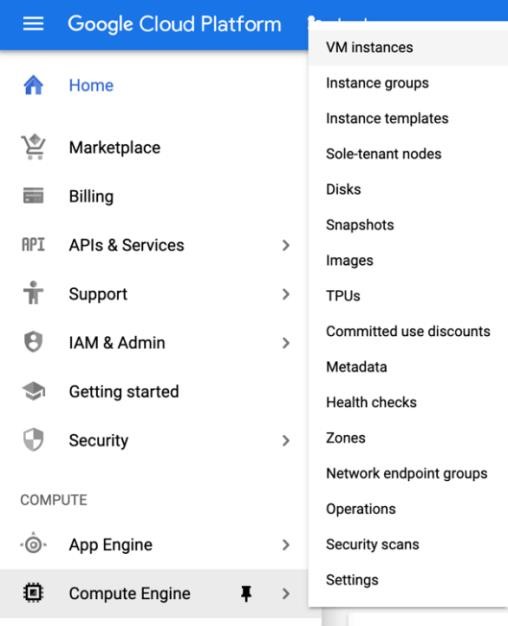
Click on the tab shown in the image below and then click on the + sign to create a new project.





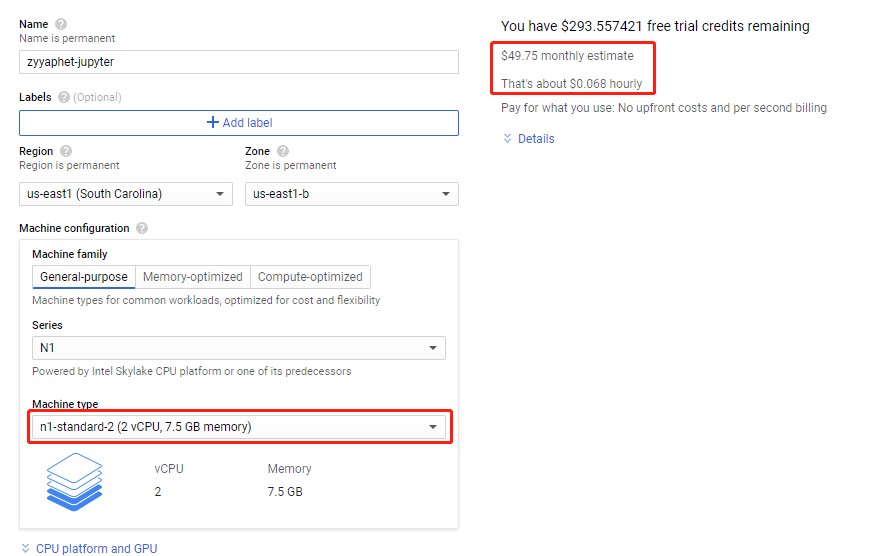
# Step 3: Create a VM instance

Click on the three lines on the upper left corner, click on "Compute Engine’ option.



Now click on ‘Create new instance’. Name your instance, select zone as ‘ us-east4-c’. Choose your ‘machine type’. (You can always pay more to use a machine with more memory and CPUs)

RECOMMENDED Configuration:



Extravagant Configuration:

Select your boot disk as ‘Ubuntu 16.04 LTS’. Under the firewall options tick both ‘http’

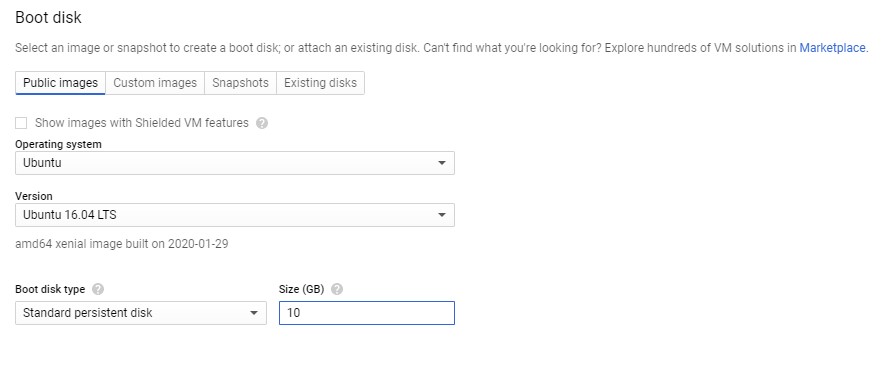
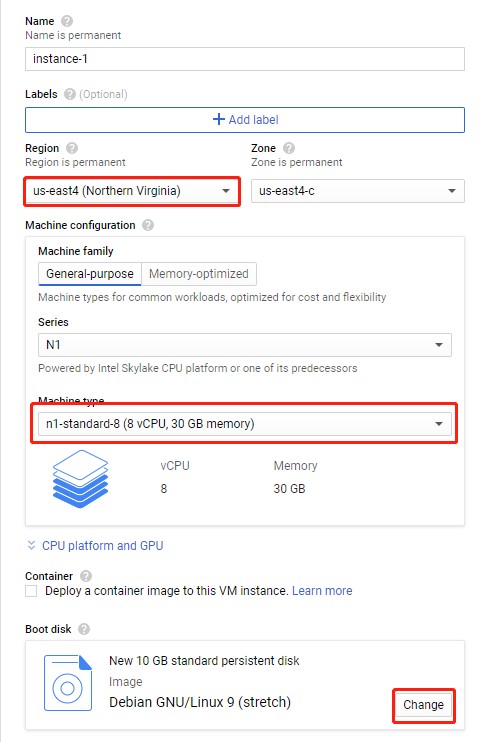
and ‘https’ (very

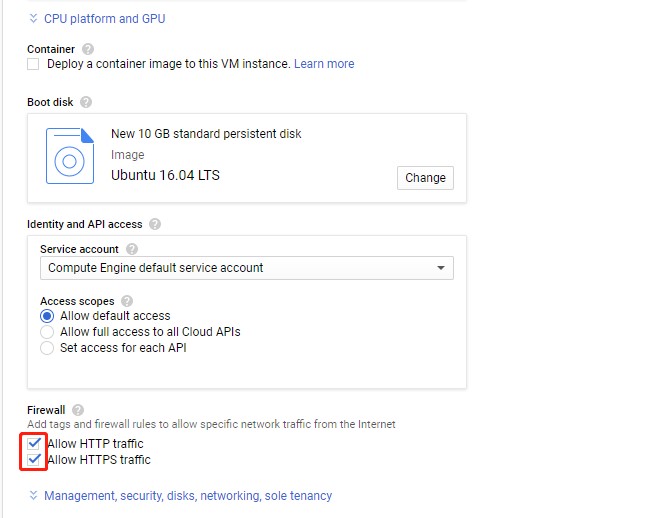
important). Then, choose the disk tab and untick

‘Delete

boot disk

when instance is deleted’.



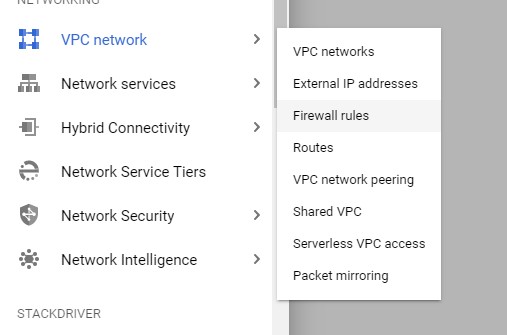


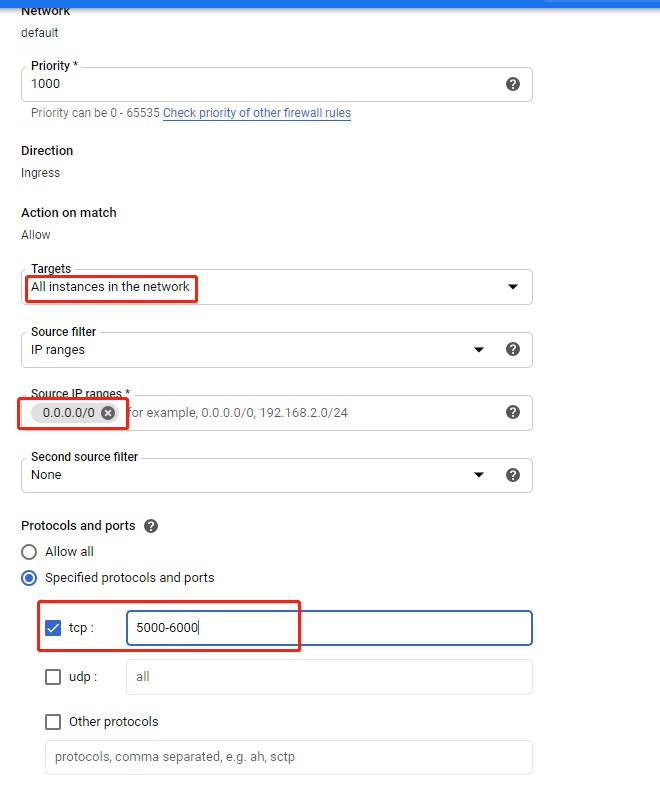
Your new VM instance should look something like this. Note down the External IP.



# Step 4: Change the Firewall setting

Now, click on the ‘Firewall rules’ setting under Networking. Then, create a new firewall rule.





# Step 5: Start your VM instance

Now start your VM instance. When you see the green tick click on SSH. This will open a command window and now you are connect to your VM instance.

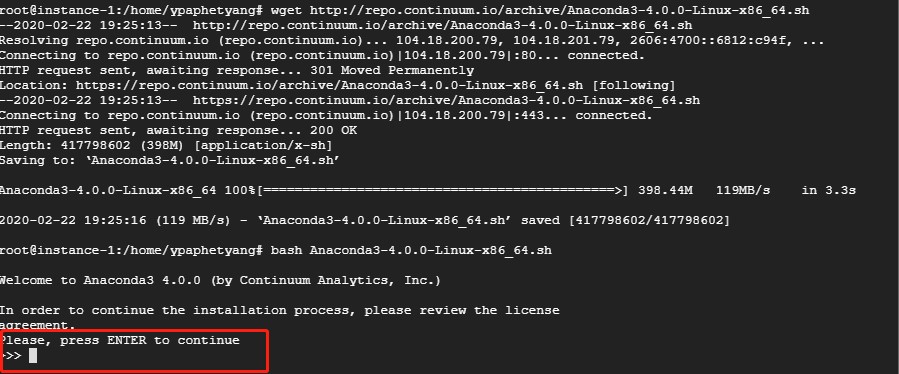
# Step 6: Install Jupyter notebook and other packages

In your SSH terminal, enter:

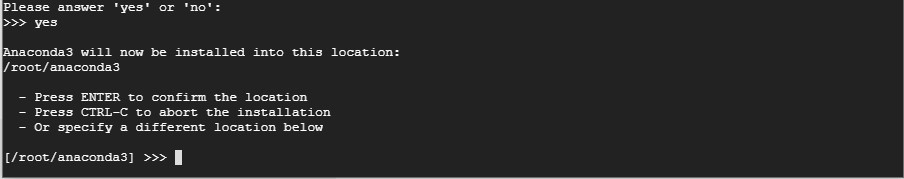
Install Anaconda.

**wget** [**http://repo.continuum.io/archive/Anaconda3-4.0.0-Linux-x86\_64.sh**](http://repo.continuum.io/archive/Anaconda3-4.0.0-Linux-x86_64.sh) **bash Anaconda3-4.0.0-Linux-x86\_64.sh**

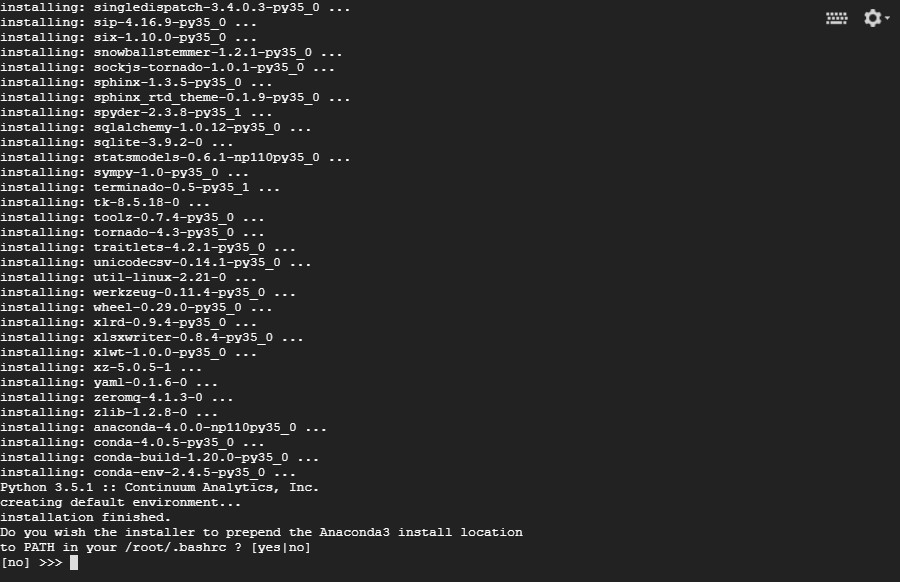
Follow the on-screen instructions. You then will be asked to review the license agreement, hit ‘Enter’ to start reviewing.



Answer yes to the last question about license, hit ‘Enter’ to start the installation process:



Answer the question regarding to prepend the install location to PATH with ‘yes’:



To make use of Anaconda right away, source your bashrc:

**source ~/.bashrc**

Now, you can try installing some packages with the command line:

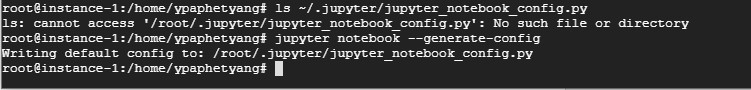
# Step 7: Set up the VM server (Not sure if this is necessary)

Open up an SSH session to your VM. Check if you have a Jupyter configuration file:

**ls ~/.jupyter/jupyter\_notebook\_config.py**

If it doesn’t exist, create one:

## jupyter notebook --generate-config



We’re going to add a few lines to your Jupyter configuration file; the file is plain text so, you can do this via your favorite editor (e.g., vim, emacs). Make sure you replace the port number with the one you allowed firewall access to in step 5.

The command to edit the file is:

## vi ~/.jupyter/jupyter\_notebook\_config.py

Change to Insert mode by pressing ‘i’, then type in code below into the py file:

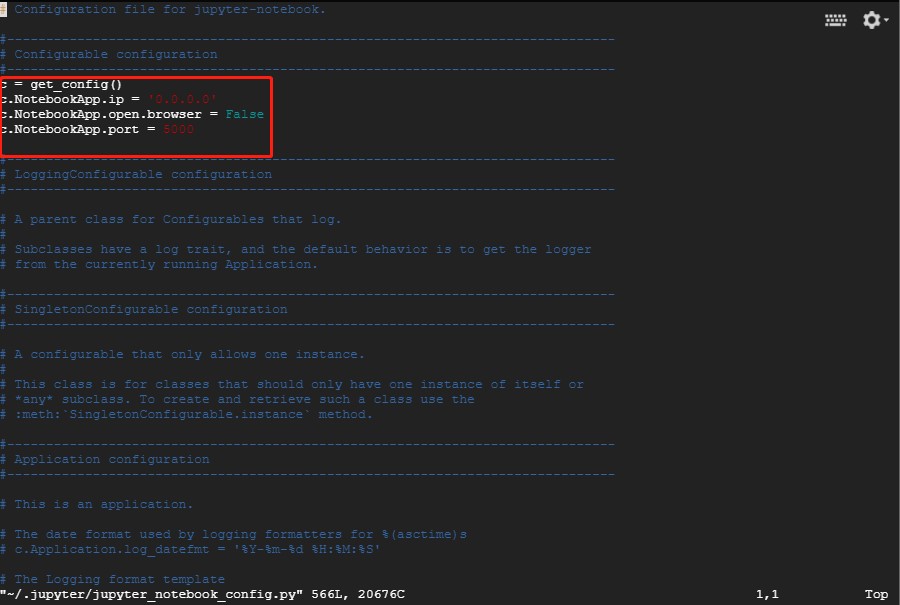
**c = get\_config()**

**c.NotebookApp.ip = '0.0.0.0'**

**c.NotebookApp.open\_browser = False**

**c.NotebookApp.port = <Port Number>**

It should look something like this:

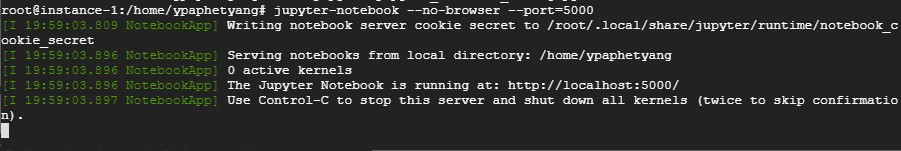


Press esc to quit insert mode and then use shift+: (windows) to change to command mode, type ‘wq!’ to save and quit the file.

# Step 8: Launching Jupyter Notebook

To run the jupyter notebook, just type the following command in the ssh window you are in:

**jupyter-notebook --no-browser --port=<PORT-NUMBER>**



Now to launch your Jupyter notebook, just type the following in your browser:

*http://<External Static IP Address>:<Port Number>*

**Step 9: Install Gsutil**

Go ahead and open another window browser for your VM instance. Run the following commands.

1. Add the Cloud SDK distribution URI as a package source:

**echo "deb [signed-by=/usr/share/keyrings/cloud.google.gpg] https://packages.cloud.google.com/apt cloud-sdk main" | sudo tee -a /etc/apt/sources.list.d/google-cloud-sdk.list**

**Make sure you have** [**apt-transport-https**](https://packages.debian.org/jessie/apt-transport-https) **installed:**

## sudo apt-get install apt-transport-https ca-certificates gnupg

1. Import the Google Cloud public key:

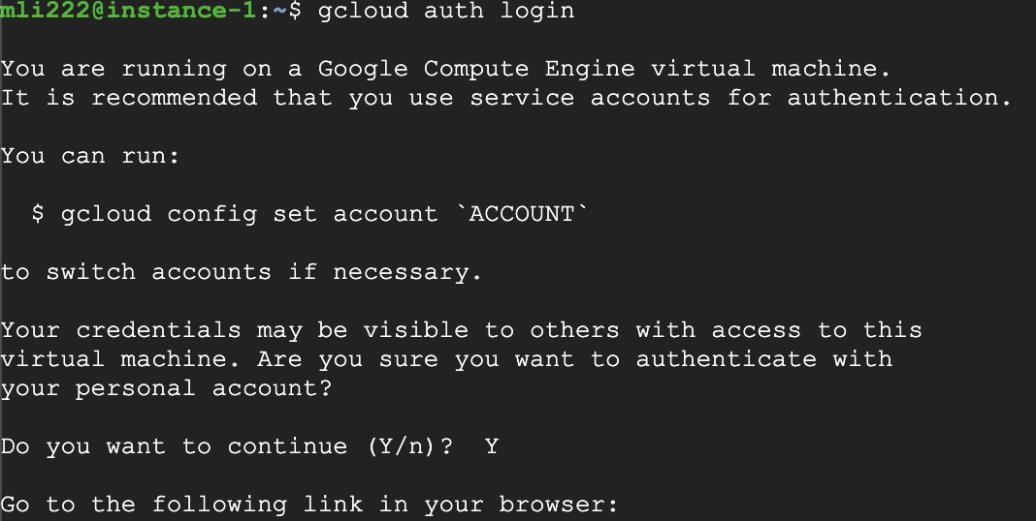
**curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key --keyring /usr/share/keyrings/cloud.google.gpg add –**

1. Update and install the Cloud SDK:

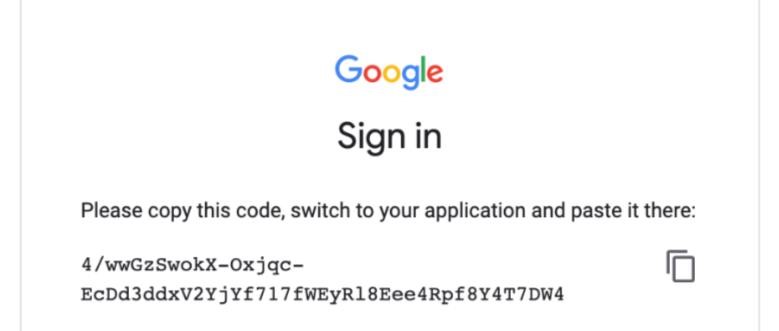
**sudo apt-get update && sudo apt-get install google-cloud-sdk**

Now, you have installed gsutil, run **gsutil -v** to check the version.

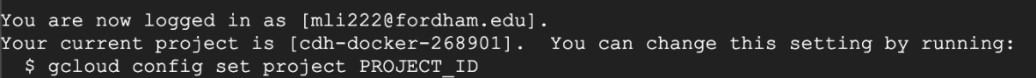
**gcloud auth login**



A link will be provided to you, so to the link and copy the verification code back to the terminal.



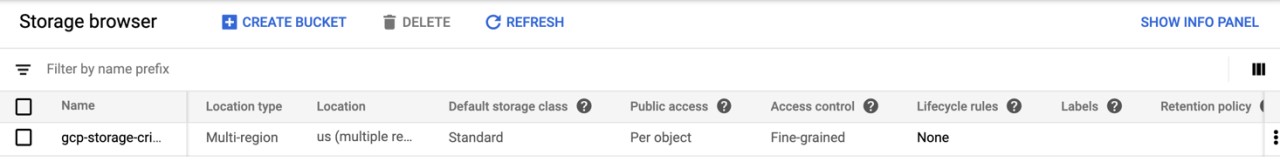
Then, you should see something like this:



**Step 10: Create a bucket on Google Cloud Storage**

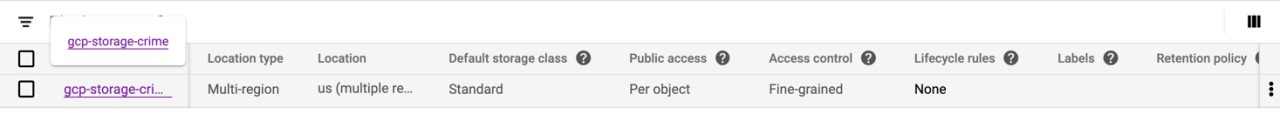
Create a bucket under Storage tab, the name has to be unique.

Note: If you have trouble creating a new bucket, it is possibly that your bucket name has been used already, give it another name and try again.

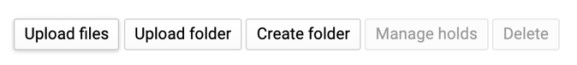


**Step 11: Upload files to the bucket**

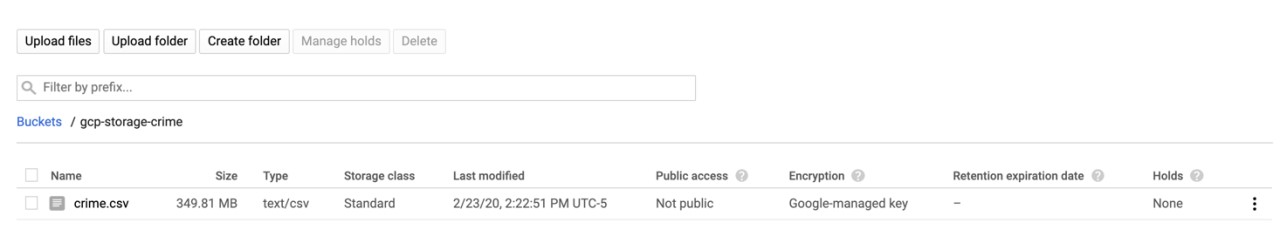
Go ahead click on the name to go inside the bucket and upload the **crime.csv** file.



Click on Upload files and browse to find the file to upload from your local machine.



Then, you should see the file uploaded.



**Step 12: Use Jupyter Notebook to check the file.**

Go to jupyter notebook and click on New



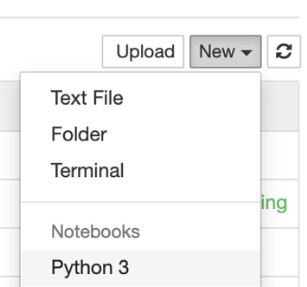
Terminal.



Run the following commands: **pip install google-cloud**

## pip install --ignore-installed google-cloud-storage

Then, create a new Python3 notebook.



Type in the following code.

from google.cloud import storage import pandas as pd

bucket\_name = "your-bucket-name" storage\_client = storage.Client()

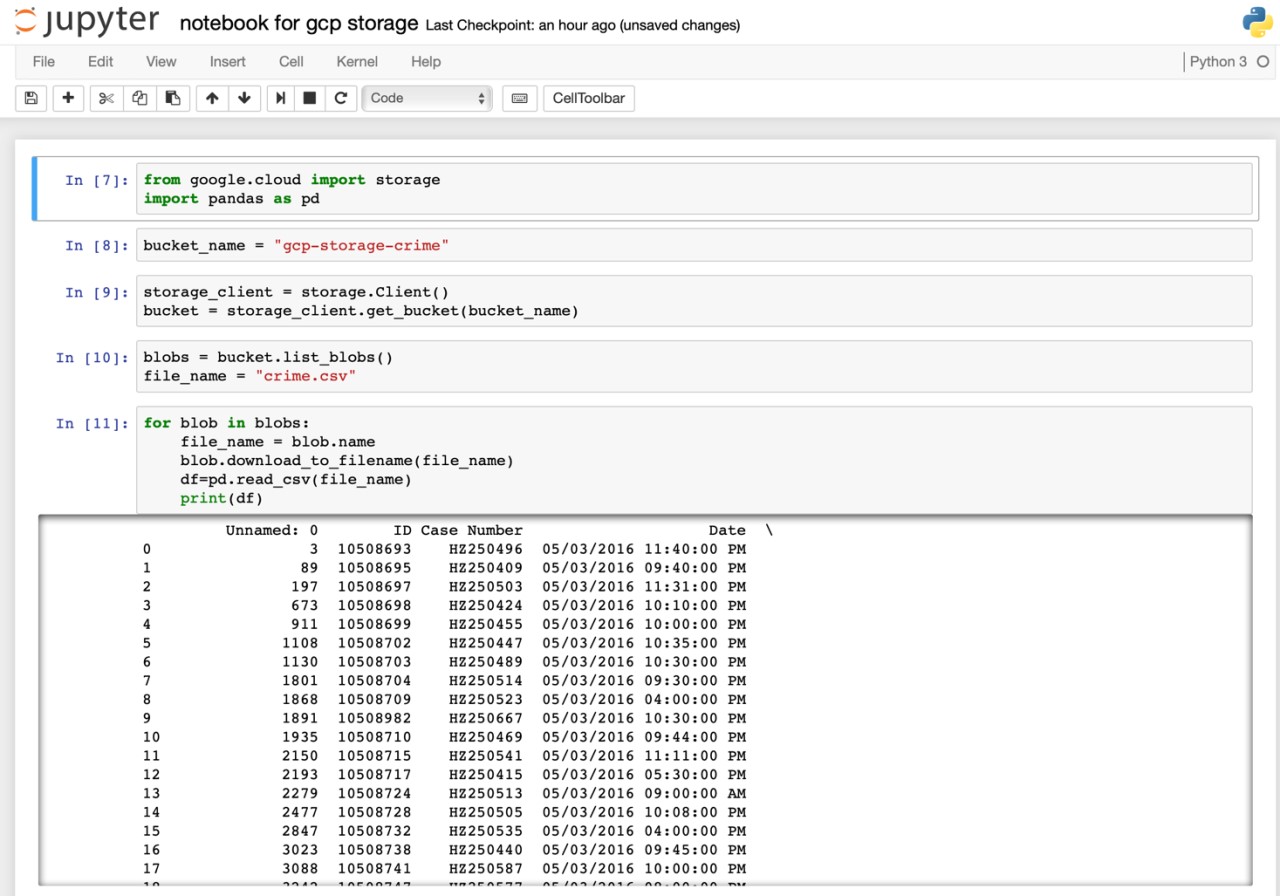
bucket = storage\_client.get\_bucket(bucket\_name) blobs = bucket.list\_blobs()

file\_name = "crime.csv"

for blob in blobs:

file\_name = blob.name blob.download\_to\_filename(file\_name) df=pd.read\_csv(file\_name) print(df)

It should be something like this:



NOTICE:

Every time when you finishing using your VM, you should go to VM instances to stop your VM, or you will be charged continuously!

