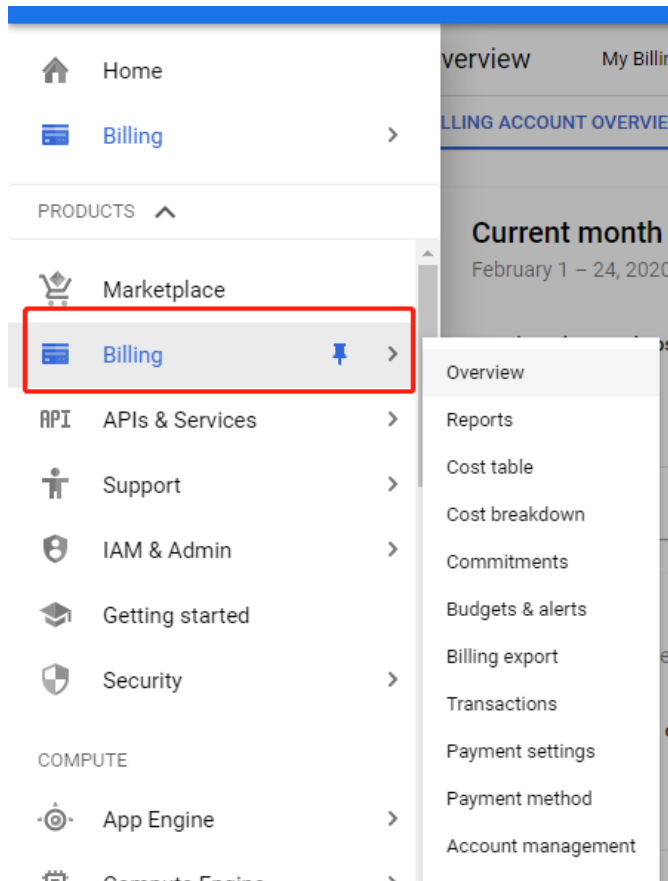


## Lab 6 - Jupyter Notebook and Google Cloud Storage

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

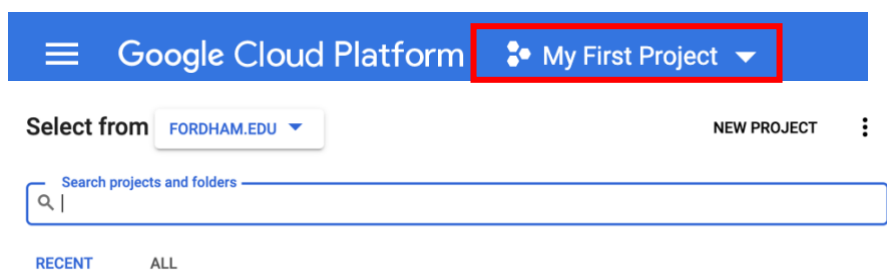
For this step, you will have to put your payment information and verify your account. **If you followed and finished Lab 5, you are fine to skip this and jump to step 2.**

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.



Google Cloud Platform

New Project

You have 10 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name \*

Jupyter

?

Project ID: jupyter-269018. It cannot be changed later. [EDIT](#)

Location \*

No organization

BROWSE

Parent organization or folder

CREATE

CANCEL

### Step 3: Create a VM instance

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

Google Cloud Platform

Home

Marketplace

Billing

APIs & Services

Support

IAM & Admin

Getting started

Security

COMPUTE

App Engine

Compute Engine

VM instances

Instance groups

Instance templates

Self-managed nodes

Disks

Snapshots

Images

TPUs

Committed use discounts

Metadata

Health checks

Zones

Network endpoint groups

Operations

Security scans

Settings

Compute Engine

VM instances

Compute Engine lets you use virtual machines that run on Google's infrastructure. Create micro-VMs or larger instances running Debian, Windows, or other standard images. Create your first VM instance, import it using a migration service, or try the quickstart to build a sample app.

Create

 or 

Import

 or 

Take the quickstart

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:

**Name** ⓘ  
Name is permanent

**Labels** ⓘ (Optional)

**Region** ⓘ  
Region is permanent


**Zone** ⓘ  
Zone is permanent

**Machine configuration** ⓘ

**Machine family**  
    
Machine types for common workloads, optimized for cost and flexibility

**Series**  
  
Powered by Intel Skylake CPU platform or one of its predecessors

**Machine type**

	vCPU	Memory
	2	7.5 GB

⌵ CPU platform and GPU

You have \$293.557421 free trial credits remaining  
\$49.75 monthly estimate  
That's about \$0.068 hourly  
Pay for what you use: No upfront costs and per second billing  
[Details](#)

## Extravagant Configuration:

**Name** ⓘ  
Name is permanent

**Labels** ⓘ (Optional)

**Region** ⓘ  
Region is permanent


**Zone** ⓘ  
Zone is permanent

**Machine configuration** ⓘ

**Machine family**  
   
Machine types for common workloads, optimized for cost and flexibility

**Series**  
  
Powered by Intel Skylake CPU platform or one of its predecessors


**Machine type**

	vCPU	Memory
	8	30 GB

⌵ CPU platform and GPU

**Container** ⓘ  
☐ Deploy a container image to this VM instance. [Learn more](#)

**Boot disk** ⓘ



New 10 GB standard persistent disk  
Image  
Debian GNU/Linux 9 (stretch)

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'.

## Boot disk

Select an image or snapshot to create a boot disk, or attach an existing disk. Can't find what you're looking for? Explore hundreds of VM solutions in [Marketplace](#).

☐ Show images with Shielded VM features ⓘ

**Operating system**

**Version**

amd64 xenial image built on 2020-01-29


**Boot disk type** ⓘ

**Size (GB)** ⓘ

⌵ CPU platform and GPU

**Container** ?  
☐ Deploy a container image to this VM instance. [Learn more](#)

**Boot disk** ?

 New 10 GB standard persistent disk  
Image  
Ubuntu 16.04 LTS Change

**Identity and API access** ?

**Service account** ?  
Compute Engine default service account



**Access scopes** ?  
☒ Allow default access  
☐ Allow full access to all Cloud APIs  
☐ Set access for each API

**Firewall** ?  
Add tags and firewall rules to allow specific network traffic from the Internet

☒ Allow HTTP traffic  
☒ Allow HTTPS traffic

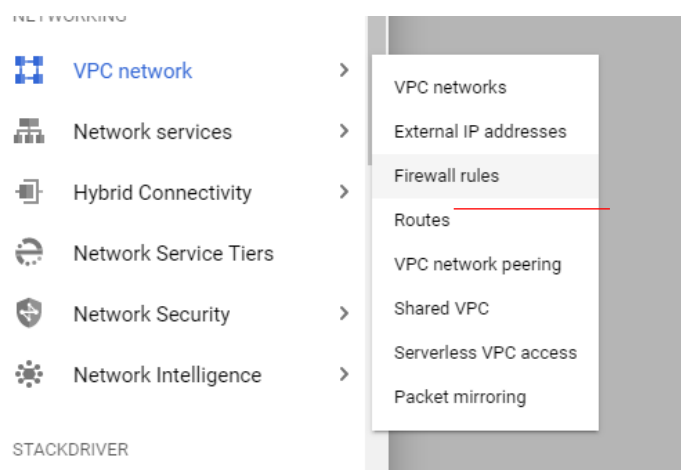
⌵ Management, security, disks, networking, sole tenancy

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

<input type="checkbox"/> Name ^	Zone	Recommendation	In use by	Internal IP	External IP	Connect
<input type="checkbox"/>  instance-1	us-east4-c			10.150.0.2 (nic0)	35.245.246.202 	SSH ▾ ⋮

#### Step 4: Change the Firewall setting

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



**Network**  
default

Priority \*  
1000  
Priority can be 0 - 65535 [Check priority of other firewall rules](#)

**Direction**  
Ingress

**Action on match**  
Allow

**Targets**  
All instances in the network

**Source filter**  
IP ranges

**Source IP ranges \***  
0.0.0.0/0 for example, 0.0.0.0/0, 192.168.2.0/24

**Second source filter**  
None

**Protocols and ports**  
☐ Allow all  
☒ Specified protocols and ports

☒ tcp : 5000-6000

☐ udp : all

☐ Other protocols  
protocols, comma separated, e.g. ah, sctp

### 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
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.

```

root@instance-1:/home/ypaphetyang# wget http://repo.continuum.io/archive/Anaconda3-4.0.0-Linux-x86_64.sh
--2020-02-22 19:25:13-- http://repo.continuum.io/archive/Anaconda3-4.0.0-Linux-x86_64.sh
Resolving repo.continuum.io (repo.continuum.io)... 104.18.200.79, 104.18.201.79, 2606:4700::6812:c94f, ...
Connecting to repo.continuum.io (repo.continuum.io)|104.18.200.79|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://repo.continuum.io/archive/Anaconda3-4.0.0-Linux-x86_64.sh [following]
--2020-02-22 19:25:13-- https://repo.continuum.io/archive/Anaconda3-4.0.0-Linux-x86_64.sh
Connecting to repo.continuum.io (repo.continuum.io)|104.18.200.79|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 417798602 (398M) [application/x-sh]
Saving to: 'Anaconda3-4.0.0-Linux-x86_64.sh'

Anaconda3-4.0.0-Linux-x86_64 100%[=====>] 398.44M 119MB/s in 3.3s

2020-02-22 19:25:16 (119 MB/s) - 'Anaconda3-4.0.0-Linux-x86_64.sh' saved [417798602/417798602]

root@instance-1:/home/ypaphetyang# bash Anaconda3-4.0.0-Linux-x86_64.sh

Welcome to Anaconda3 4.0.0 (by Continuum Analytics, Inc.)

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

```

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

```

Please answer 'yes' or 'no':
->>> yes

Anaconda3 will now be installed into this location:
/root/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/root/anaconda3] >>>

```

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

```

installing: singledispatch-3.4.0.3-py35_0 ...
installing: sip-4.16.9-py35_0 ...
installing: six-1.10.0-py35_0 ...
installing: snowballstemmer-1.2.1-py35_0 ...
installing: socksjs-tornado-1.0.1-py35_0 ...
installing: sphinx-1.3.5-py35_0 ...
installing: sphinx_rtd_theme-0.1.9-py35_0 ...
installing: spyder-2.3.8-py35_1 ...
installing: sqlalchemy-1.0.12-py35_0 ...
installing: sqlite-3.9.2-0 ...
installing: statsmodels-0.6.1-np110py35_0 ...
installing: sympy-1.0-py35_0 ...
installing: terminado-0.5-py35_1 ...
installing: tk-8.5.18-0 ...
installing: toolz-0.7.4-py35_0 ...
installing: tornado-4.3-py35_0 ...
installing: traitlets-4.2.1-py35_0 ...
installing: unicodedcsv-0.14.1-py35_0 ...
installing: util-linux-2.21-0 ...
installing: werkzeug-0.11.4-py35_0 ...
installing: wheel-0.29.0-py35_0 ...
installing: xlrd-0.9.4-py35_0 ...
installing: xlswriter-0.8.4-py35_0 ...
installing: xlwt-1.0.0-py35_0 ...
installing: xz-5.0.5-1 ...
installing: yaml-0.1.6-0 ...
installing: zeromq-4.1.3-0 ...
installing: zlib-1.2.8-0 ...
installing: anaconda-4.0.0-np110py35_0 ...
installing: conda-4.0.5-py35_0 ...
installing: conda-build-1.20.0-py35_0 ...
installing: conda-env-2.4.5-py35_0 ...
Python 3.5.1 :: Continuum Analytics, Inc.
creating default environment...
installation finished.

Do you wish the installer to prepend the Anaconda3 install location
to PATH in your /root/.bashrc ? [yes|no]
[no] >>>

```

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
```

```
root@instance-1:/home/ypaphetyang# ls ~/.jupyter/jupyter_notebook_config.py
ls: cannot access '/root/.jupyter/jupyter_notebook_config.py': No such file or directory
root@instance-1:/home/ypaphetyang# jupyter notebook --generate-config
Writing default config to: /root/.jupyter/jupyter_notebook_config.py
root@instance-1:/home/ypaphetyang#
```

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:



```

Configuration file for jupyter-notebook.

#-----#
# Configurable configuration
#-----#
c = get_config()
c.NotebookApp.ip = '0.0.0.0'
c.NotebookApp.open_browser = False
c.NotebookApp.port = 5000

#-----#
# LoggingConfigurable configuration
#-----#

# A parent class for Configurables that log.
#
# Subclasses have a log trait, and the default behavior is to get the logger
# from the currently running Application.
#-----#
# SingletonConfigurable configuration
#-----#

# A configurable that only allows one instance.
#
# This class is for classes that should only have one instance of itself or
# *any* subclass. To create and retrieve such a class use the
# :meth:`SingletonConfigurable.instance` method.
#-----#
# Application configuration
#-----#

# This is an application.
#
# The date format used by logging formatters for %(asctime)s
# c.Application.log_datefmt = '%Y-%m-%d %H:%M:%S'
#
# The Logging format template
#
~/jupyter/jupyter_notebook_config.py" 566L, 20676C
1,1 Top

```

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>
```

```

root@instance-1:/home/ypaphetyang# jupyter-notebook --no-browser --port=5000
[I 19:59:03.809 NotebookApp] Writing notebook server cookie secret to /root/.local/share/jupyter/runtime/notebook_cookie_secret
[I 19:59:03.896 NotebookApp] Serving notebooks from local directory: /home/ypaphetyang
[I 19:59:03.896 NotebookApp] 0 active kernels
[I 19:59:03.896 NotebookApp] The Jupyter Notebook is running at: http://localhost:5000/
[I 19:59:03.897 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

```

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](#) installed:

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

2. 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 -
```

3. 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

```
mli222@instance-1:~$ gcloud auth login

You are running on a Google Compute Engine virtual machine.
It is recommended that you use service accounts for authentication.

You can run:

  $ gcloud config set account `ACCOUNT`

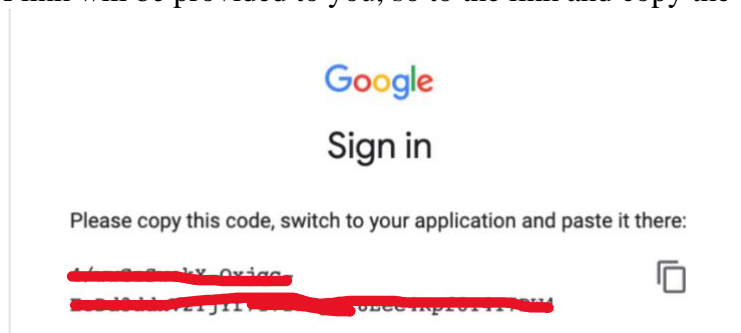
to switch accounts if necessary.

Your credentials may be visible to others with access to this
virtual machine. Are you sure you want to authenticate with
your personal account?

Do you want to continue (Y/n)? Y

Go to the following link in your browser:
```

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:

```
You are now logged in as [redacted].
Your current project is [redacted]. You can change this setting by running:
$ gcloud config set project PROJECT_ID
```

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

Run the following command to create a bucket, the name has to be unique.

```
gsutil mb gs://your-budget-name
```

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.

Then, you should be able to find your bucket under Storage tab.

Storage browser

CREATE BUCKET

DELETE

REFRESH

SHOW INFO PANEL

Filter by name prefix

<input type="checkbox"/>	Name	Location type	Location	Default storage class	Public access	Access control	Lifecycle rules	Labels	Retention policy
<input type="checkbox"/>	gcp-storage-cri...	Multi-region	us (multiple re...	Standard	Per object	Fine-grained	None		

## Step 11: Upload files to the bucket

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

gcp-storage-crime								
<input type="checkbox"/>	Name	Location type	Location	Default storage class ?	Public access ?	Access control ?	Lifecycle rules ?	Labels ?
<input type="checkbox"/>	gcp-storage-cri...	Multi-region	us (multiple re...	Standard	Per object	Fine-grained	None	Retention policy

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

Upload files Upload folder Create folder Manage holds Delete

Then, you should see the file uploaded.

Upload files Upload folder Create folder Manage holds Delete

Filter by prefix...

Buckets / gcp-storage-crime

<input type="checkbox"/>	Name	Size	Type	Storage class	Last modified	Public access ?	Encryption ?	Retention expiration date ?	Holds ?
<input type="checkbox"/>	crime.csv	349.81 MB	text/csv	Standard	2/23/20, 2:22:51 PM UTC-5	Not public	Google-managed key	–	None

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

Go to jupyter notebook and click on New → Terminal.

jupyter

Files Running Clusters

Select items to perform actions on them.

Upload New

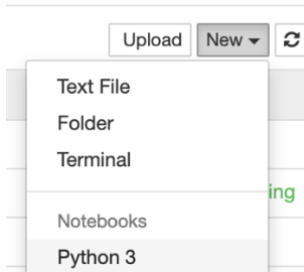
☐ anaconda3

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

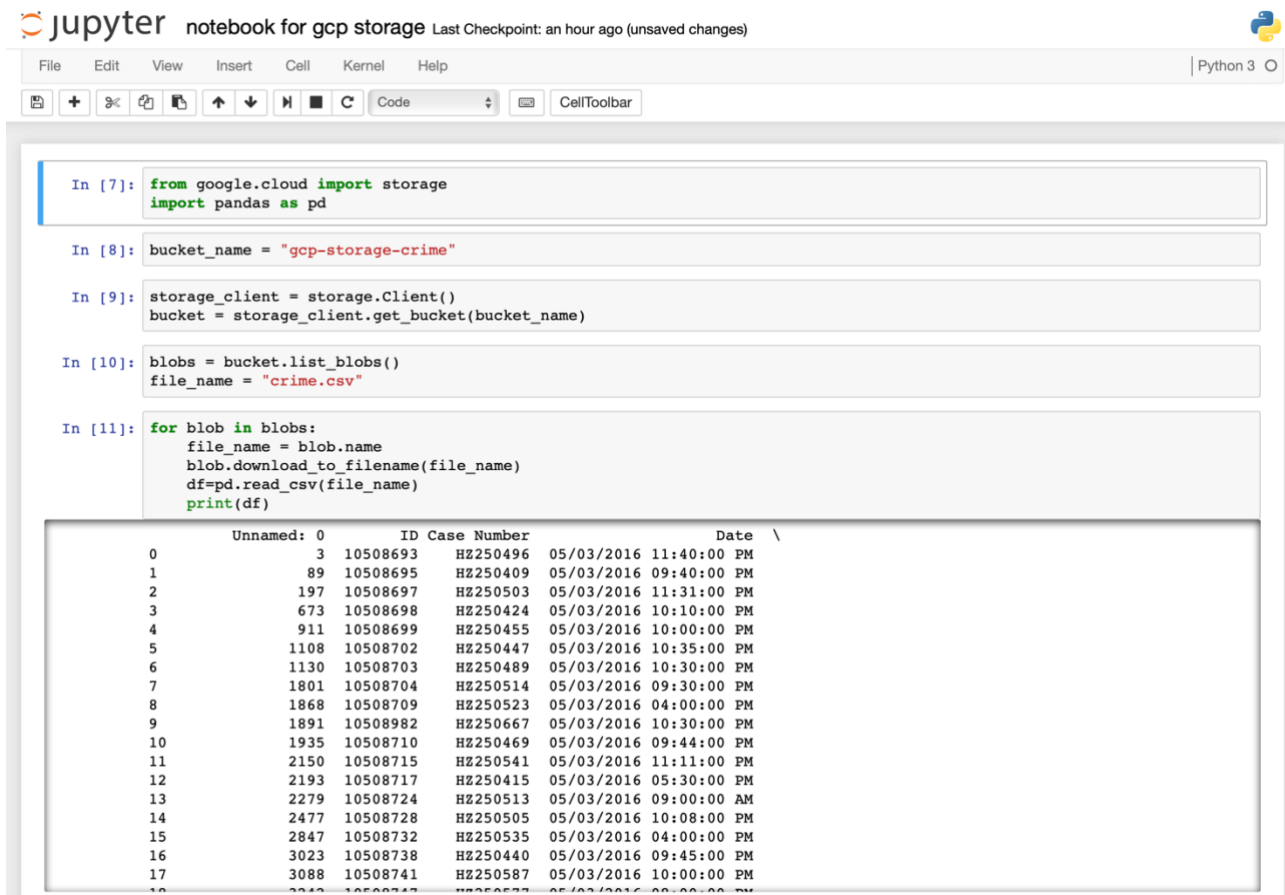
bucket_name = "your-bucket-name"
storage_client = storage.Client()

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:



The image shows a Jupyter Notebook interface titled "notebook for gcp storage" with a status bar indicating "Last Checkpoint: an hour ago (unsaved changes)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar with icons for file operations and code execution. The code is written in Python and uses the Google Cloud Storage (GCS) and Pandas libraries. It lists blobs in a bucket named "gcp-storage-crime" and downloads them to a local file named "crime.csv". The output of the code is a table of data.

```
In [7]: from google.cloud import storage
import pandas as pd

In [8]: bucket_name = "gcp-storage-crime"

In [9]: storage_client = storage.Client()
bucket = storage_client.get_bucket(bucket_name)

In [10]: blobs = bucket.list_blobs()
file_name = "crime.csv"

In [11]: for blob in blobs:
    file_name = blob.name
    blob.download_to_filename(file_name)
    df=pd.read_csv(file_name)
    print(df)
```

	Unnamed: 0	ID	Case Number	Date
0	3	10508693	HZ250496	05/03/2016 11:40:00 PM
1	89	10508695	HZ250409	05/03/2016 09:40:00 PM
2	197	10508697	HZ250503	05/03/2016 11:31:00 PM
3	673	10508698	HZ250424	05/03/2016 10:10:00 PM
4	911	10508699	HZ250455	05/03/2016 10:00:00 PM
5	1108	10508702	HZ250447	05/03/2016 10:35:00 PM
6	1130	10508703	HZ250489	05/03/2016 10:30:00 PM
7	1801	10508704	HZ250514	05/03/2016 09:30:00 PM
8	1868	10508709	HZ250523	05/03/2016 04:00:00 PM
9	1891	10508982	HZ250667	05/03/2016 10:30:00 PM
10	1935	10508710	HZ250469	05/03/2016 09:44:00 PM
11	2150	10508715	HZ250541	05/03/2016 11:11:00 PM
12	2193	10508717	HZ250415	05/03/2016 05:30:00 PM
13	2279	10508724	HZ250513	05/03/2016 09:00:00 AM
14	2477	10508728	HZ250505	05/03/2016 10:08:00 PM
15	2847	10508732	HZ250535	05/03/2016 04:00:00 PM
16	3023	10508738	HZ250440	05/03/2016 09:45:00 PM
17	3088	10508741	HZ250587	05/03/2016 10:00:00 PM

## 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!

