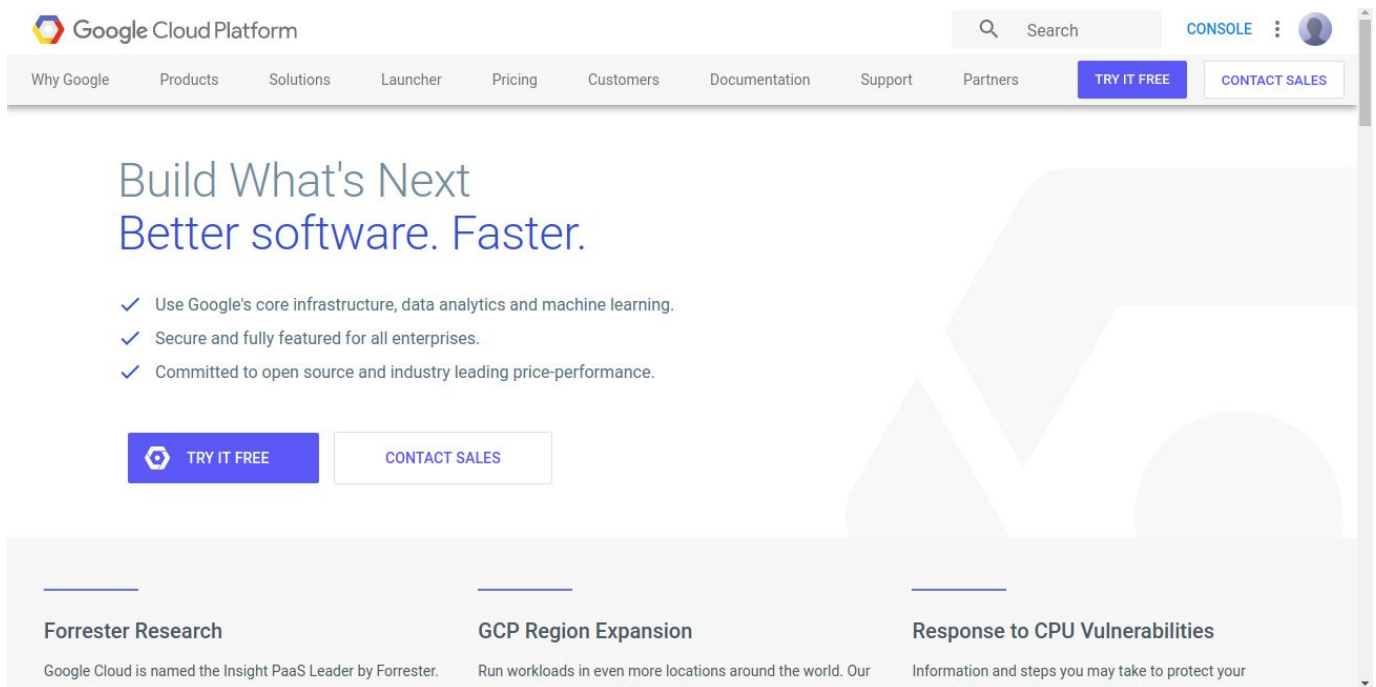


# Google Cloud Tutorial

## Abstract

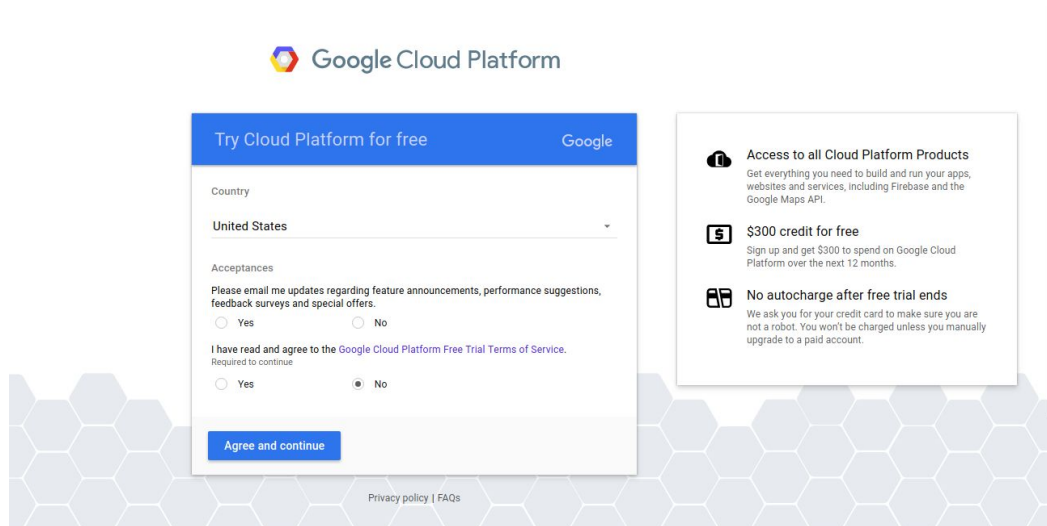
In machine learning, GPUs and/or CPUs are needed for model training, thus computation cost can be very expensive (literally). However, Google has provided cloud computing which allows users to utilize the processors remotely. What does it mean? It means that users will not be using their processors locally to do machine learning computation, instead, the data processing will be done at Google Cloud (or Google server) and it will not exhaust your computer at all. You can even use the \$100 laptop that has an Internet connection to process thousands of dataset remotely.



**Figure 1**

## Google Cloud Registration

First, go to <https://cloud.google.com/> (Figure 1) and login with your google account



The image shows the Google Cloud Platform registration page. At the top, the Google Cloud Platform logo is displayed. Below it, there is a blue header bar with the text "Try Cloud Platform for free" and the Google logo. The main form area is white and contains the following sections:

- Country:** A dropdown menu with "United States" selected.
- Acceptances:** A section with the text "Please email me updates regarding feature announcements, performance suggestions, feedback surveys and special offers." and two radio buttons: "Yes" and "No".
- I have read and agree to the Google Cloud Platform Free Trial Terms of Service:** A section with the text "Required to continue" and two radio buttons: "Yes" and "No".
- Agree and continue:** A blue button at the bottom of the form.

On the right side of the form, there are three informational boxes:

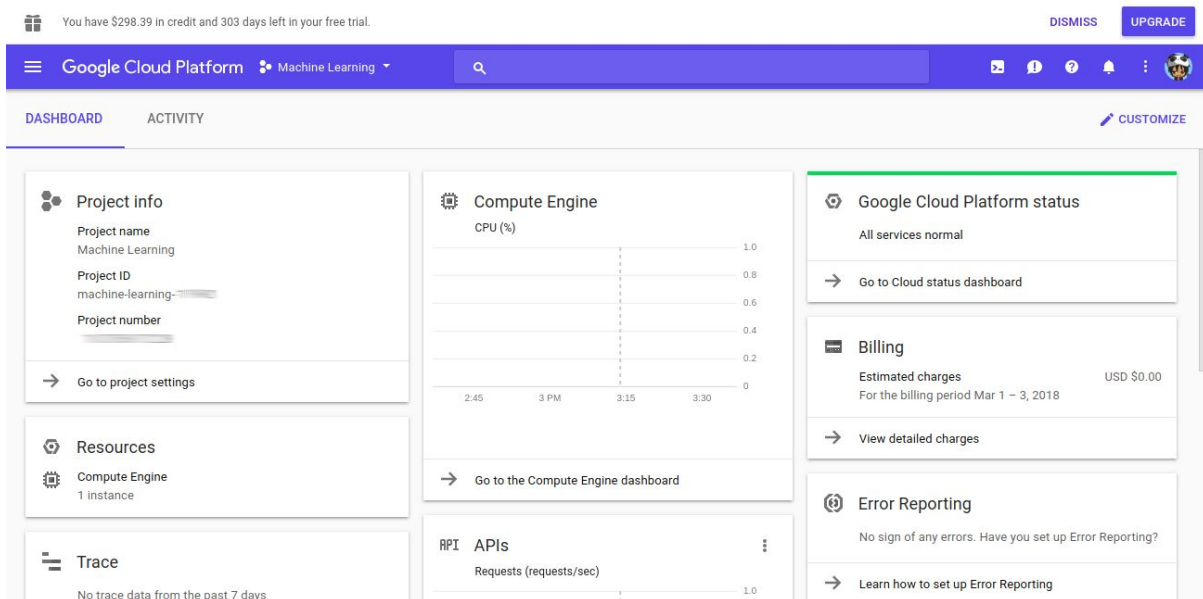
- Access to all Cloud Platform Products:** Get everything you need to build and run your apps, websites and services, including Firebase and the Google Maps API.
- \$300 credit for free:** Sign up and get \$300 to spend on Google Cloud Platform over the next 12 months.
- No autocharge after free trial ends:** We ask you for your credit card to make sure you are not a robot. You won't be charged unless you manually upgrade to a paid account.

At the bottom of the form, there is a link for "Privacy policy | FAQs".

**Figure 2**

Click "Try For Free" (Figure 1), then you will be prompted to agree with the policies. After you have done so, click "Agree and continue" (Figure 2). They may ask you for the credit card information but do not worry, this is for authentication purpose and no fee will be charged yet. In addition, Google has provided free credit for \$300 initially! So you can rest assured.

Then click "Start My Free Trial". You will be transfer to your homepage (Figure 3)



**Figure 3**

Go to "Compute Engine" and select "VM Instances". On the main section, click "Create Instance".

- Change the name as your preference
- Change the zone to be "us-west1-b"
- Change the machine cpu to "8 vCPUs"
- Change the boot disk to be "Ubuntu 16.04 LTS"
- Allow HTTP (and HTTPS) traffic
- 

Your configuration should match Figure 4

Google Cloud Platform Machine Learning

Compute Engine

VM instances

Instance groups

Instance templates

Disks

Snapshots

Images

Cloud TPUs

Committed use discounts

Metadata

Health checks

Zones

Operations

Quotas

Settings

Cloud Launcher

Create an instance

Name <sup>?</sup>

instance-1

Zone <sup>?</sup>

us-west1-b

Machine type <sup>?</sup>

Customize to select cores, memory and GPUs.

8 vCPUs 52 GB memory [Customize](#)

[Upgrade your account](#) to create instances with up to 96 cores

Container <sup>?</sup>

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

Boot disk <sup>?</sup>

New 10 GB standard persistent disk

Image

Ubuntu 16.04 LTS [Change](#)

Identity and API access <sup>?</sup>

Service account <sup>?</sup>

Compute Engine default service account

Access scopes <sup>?</sup>

☒ Allow default access

☐ Allow full access to all Cloud APIs

☐ Set access for each API

Firewall <sup>?</sup>

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

☒ Allow HTTP traffic

☒ Allow HTTPS traffic

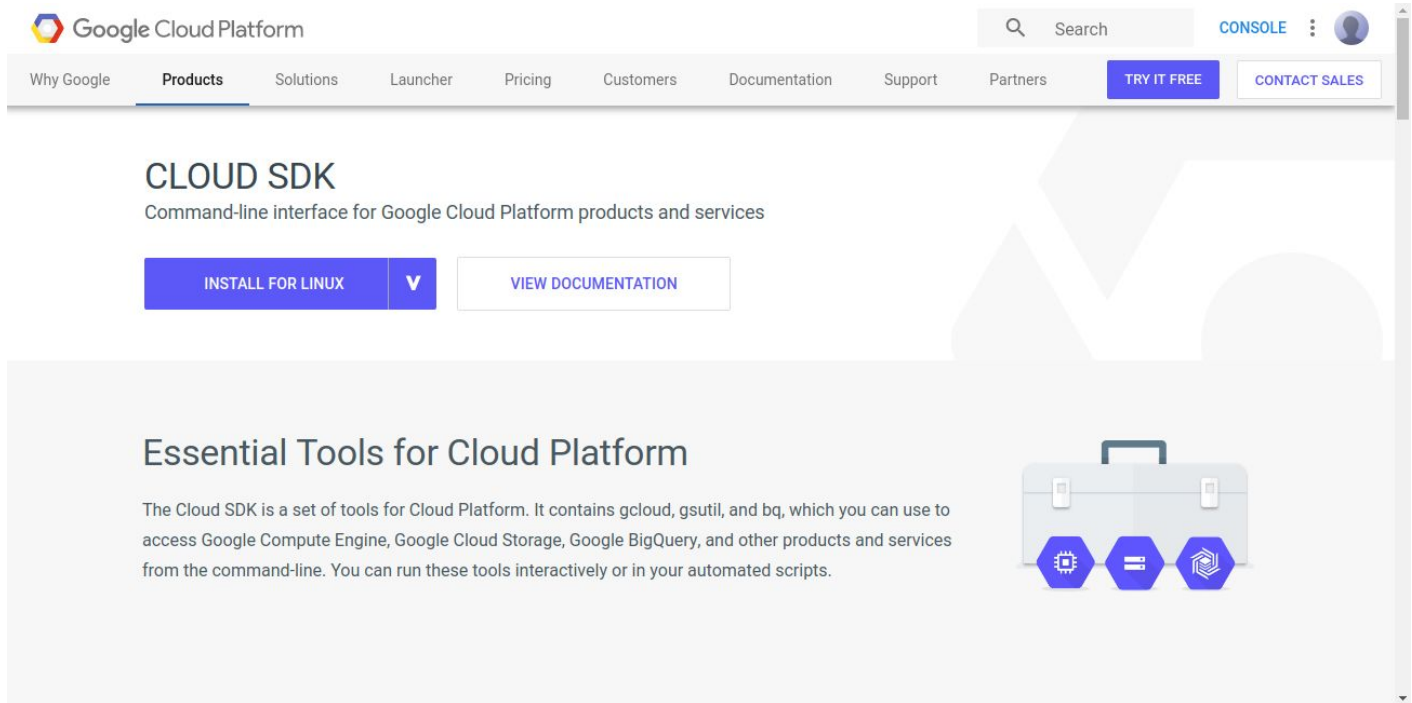
\$242.41 per month estimated

Effective hourly rate \$0.332 (730 hours per month)

[Details](#)

Figure 4

Install the Google Cloud SDK from <https://cloud.google.com/sdk> (Figure 5). Depending on operating you are using, make sure to select the correct one.



**Figure 5**

Turn on your instance by clicking "Start" on the top-center (Figure 6) and run following command in your command prompt.

```
./<DIRECTORY-WHERE-GOOGLE-CLOUD-IS-INSTALLED>/bin/gcloud compute ssh --zone=us-west1-b
```

You may be prompted to setup your location, follow the instruction carefully. Once it is completed, you will be see the username has been changed (In my case, from "Ze" to "cs1", this indicate that I am now connected to Google Cloud) (Figure 7). And congratulation! You are now running the Google Cloud instance!

## IMPORTANT

Be sure to "STOP" the instance when you are not using it. It is **extremely** important, because by leaving your instance on without using it, the server will be running in the background and Google will count and deduct your Google Cloud's credit continuously (In this case: your \$300 free credit).

your free trial.

Machine Learning ▾

instances [+ CREATE INSTANCE](#) [IMPORT VM](#) [REFRESH](#) [▶ START](#) [■ STOP](#)

Filter VM instances [Columns ▾](#)

Name ^	Zone	Recommendation	Internal IP	External IP	Connect
cs1	us-west1-b			None	SSH ▾

cs1

LABELS

Labels

Save

Figure 6

```
pat@Ze:~/Documents/google-cloud-sdk$ ./bin/gcloud compute ssh --zone=us-west1-b cs1
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.13.0-1008-gcp x86_64)

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

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

43 packages can be updated.
0 updates are security updates.

*** System restart required ***
Last login: Sun Feb 11 03:28:13 2018 from [REDACTED]
pat@cs1:~$
```

Figure 7

## Google Cloud Configuration

After you have completed the instance setup. Download the following files and install the essential packages (Figure 8):

- `wget https://csulb-ml.github.io/packages/setup_googlecloud.sh`
- `wget https://csulb-ml.github.io/packages/requirements.txt`
- `chmod +x setup_googlecloud.sh`
- `virtualenv -p python3 .env`
- `./setup_googlecloud.sh`

```
at@csulb-ml:~$ wget https://csulb-ml.github.io/packages/setup_googlecloud.sh
--2018-03-05 08:23:00-- https://csulb-ml.github.io/packages/setup_googlecloud.sh
Resolving csulb-ml.github.io (csulb-ml.github.io)... 
Connecting to csulb-ml.github.io (csulb-ml.github.io)|... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1076 (1.1K) [application/x-sh]
Saving to: 'setup_googlecloud.sh'

setup_googlecloud.sh          100%[=====] 1.05K  --.-KB/s
2018-03-05 08:23:00 (200 MB/s) - 'setup_googlecloud.sh' saved [1076/1076]

at@csulb-ml:~$ wget https://csulb-ml.github.io/packages/requirements.txt
--2018-03-05 08:23:09-- https://csulb-ml.github.io/packages/requirements.txt
Resolving csulb-ml.github.io (csulb-ml.github.io)... 151.101.53.147, 2a04:4e42:d::403
Connecting to csulb-ml.github.io (csulb-ml.github.io)|151.101.53.147|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 793 [text/plain]
Saving to: 'requirements.txt'

requirements.txt             100%[=====] 793  --.-KB/s
2018-03-05 08:23:09 (160 MB/s) - 'requirements.txt' saved [793/793]

at@csulb-ml:~$ chmod +x setup_googlecloud.sh
at@csulb-ml:~$ ./setup_googlecloud.sh
```

**Figure 8**

The installation process will take sometimes and you will be prompted to enter "y" to accept the package allocation in your instance. At the end of the installation, you should see no errors (Figure 9)



```

Installing collected packages: Cython, MarkupSafe, Jinja2, Pillow, Pygments, appn
, cyclor, decorator, future, gnureadline, ipython-genutils, traitlets, jupyter-co
, ptyprocess, pexpect, ipython, ipykernel, functools32, jsonschema, nbformat, sing
ipywidgets, qtconsole, jupyter-console, jupyter, subprocess32, pyparsing, pytz,
Successfully installed Cython-0.23.4 Jinja2-2.8 MarkupSafe-0.23 Pillow-3.0.0 Pygm
-hostname-3.5.0.1 certifi-2015.11.20.1 cyclor-0.10.0 decorator-4.0.6 functools32-
ython-4.0.1 ipython-genutils-0.1.0 ipywidgets-4.1.1 jsonschema-2.5.1 jupyter-1.0.
6 matplotlib-2.0.0 mistune-0.7.1 nbconvert-4.1.0 nbformat-4.0.1 notebook-4.0.6 nu
ess-0.5 pyparsing-2.0.7 python-dateutil-2.4.2 pytz-2015.7 pyzmq-15.1.0 qtconsole-
six-1.10.0 subprocess32-3.2.7 terminado-0.5 tornado-4.3 traitlets-4.0.0
*****
***** End of Google Cloud Set-up Script *****
*****

If you had no errors, You can proceed to work with your virtualenv as normal.
(run 'source .env/bin/activate' in your assignment directory to load the venv,
and run 'deactivate' to exit the venv. See assignment handout for details.)
pat@csulb-ml:~$

```

Figure 9

## Setup a static IP address

Go to "Networking" > "VPC networks" > "External IP addresses" (Figure 10)

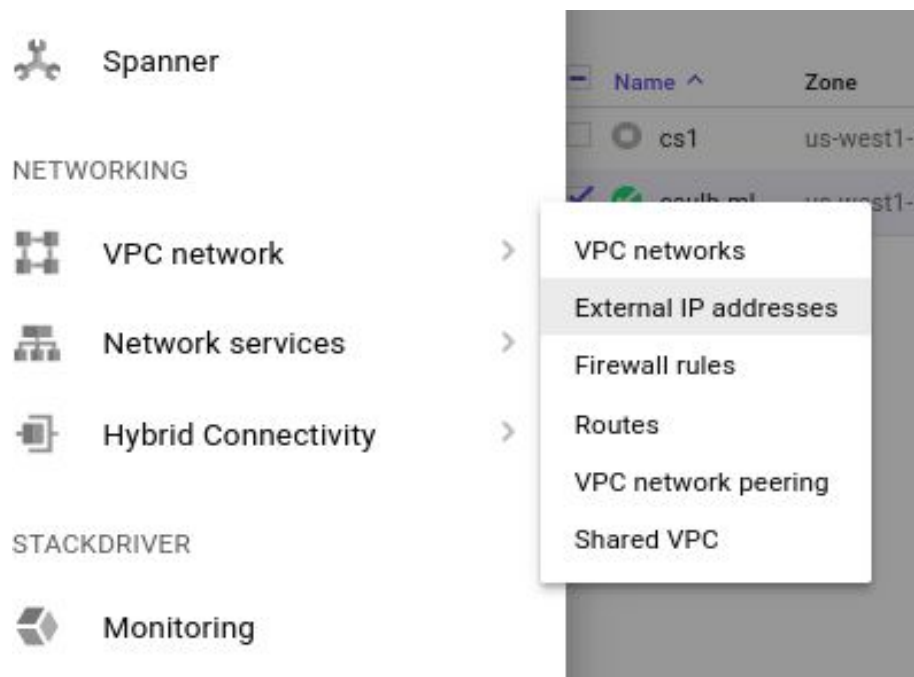


Figure 10

Change the Type from "Ephemeral" to "Static", you should have something like Figure 11 page.

Note: Once you have done with your instance usage, be sure to change Static back to Ephemeral because Google may charge you for unused static IP address

External IP addresses							<a href="#">+ RESERVE STATIC ADDRESS</a>	<a href="#">REFRESH</a>	<a href="#">RELEASE STATIC ADDRESS</a>
<input type="checkbox"/>	Name	External Address	Region	Type ▾	Version	In use by	Labels		
<input type="checkbox"/>	csulbstatic	35.230.21.100	us-west1	Static ▾	IPv4	VM instance csulb-ml (Zone b)	Change		

Figure 11

## Create a firewall rule

Go to "Networking" > "VPC Network" > "Firewall rules" (Figure 12)

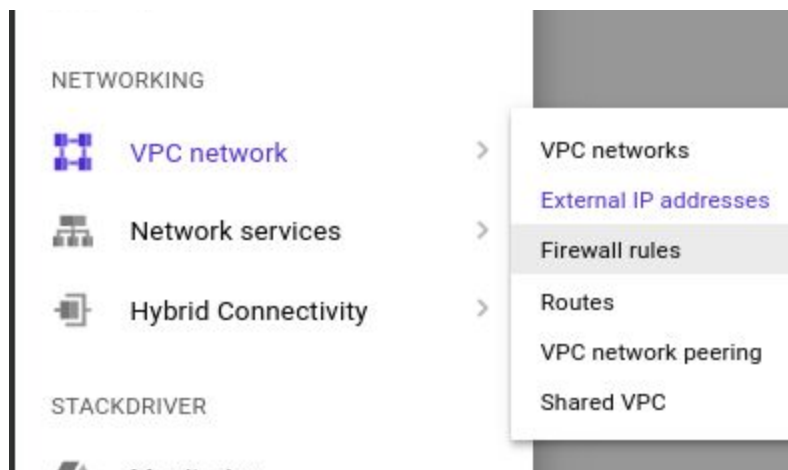


Figure 12

Click "Create firewall rule".

- Name your firewall
- Change Targets to "All instances in the network"
- Source IP range to "0.0.0.0/0"
- Add protocols and ports "tcp:7000"

The setting should match Figure 13



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

1000

**Direction of traffic** ?

☒ Ingress  
☐ Egress

**Action on match** ?

☒ Allow  
☐ Deny

**Targets** ?

All instances in the network ▼

**Source filter** ?

IP ranges ▼

**Source IP ranges** ?

0.0.0.0/0 ✕

**Second source filter** ?

None ▼

**Protocols and ports** ?

☐ Allow all  
☒ Specified protocols and ports

tcp:7000

[Create](#) [Cancel](#)

Equivalent [REST](#) or [command line](#)

**Figure 13**

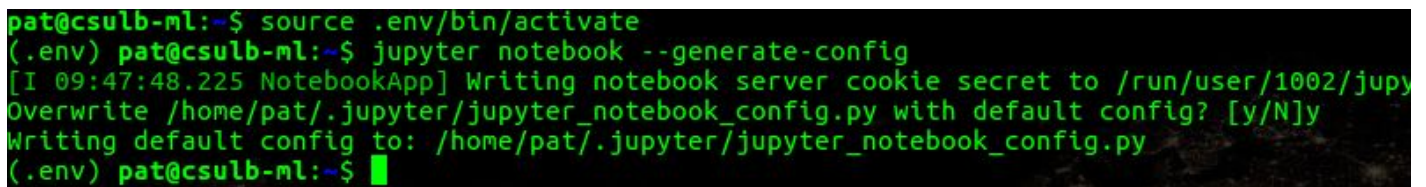
## Configuring Jupyter Notebook

Install Anaconda and Tensorflow with following commands:

- mkdir downloads
- cd downloads
- 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
- source ~/.bashrc
- conda install -c jjhelmus tensorflow=0.8.0rc0

Create a Jupyter Configuration file with following command (be sure to activate the environment first) (Figure 14):

- source .env/bin/activate
- jupyter notebook --generate-config



```
pat@csulb-ml:~$ source .env/bin/activate
(.env) pat@csulb-ml:~$ jupyter notebook --generate-config
[I 09:47:48.225 NotebookApp] Writing notebook server cookie secret to /run/user/1002/jupyter
Overwrite /home/pat/.jupyter/jupyter_notebook_config.py with default config? [y/N]y
Writing default config to: /home/pat/.jupyter/jupyter_notebook_config.py
(.env) pat@csulb-ml:~$
```

**Figure 14**

Configure the Jupyter Notebook with:

- sudo nano ~/.jupyter/jupyter\_notebook\_config.py

Add following text at the end of the "jupyter\_notebook\_config.py" file (Figure 15):

```
c = get_config()
```

```
c.NotebookApp.ip = '*'
```

```
c.NotebookApp.open_browser = False
```

```
c.NotebookApp.port = 7000
```

```
GNU nano 2.5.3 File: /home/pat/.jupyter/jupyter_notebook_conf

# The secret key with which notebooks are signed.
# c.NotebookNotary.secret = ''

# The sqlite file in which to store notebook signatures. By default, this will
# be in your Jupyter runtime directory. You can set it to ':memory:' to disable
# sqlite writing to the filesystem.
# c.NotebookNotary.db_file = u''

# The hashing algorithm used to sign notebooks.
# c.NotebookNotary.algorithm = 'sha256'

# The file where the secret key is stored.
# c.NotebookNotary.secret_file = u''

#-----
# KernelSpecManager configuration
#-----

# Whitelist of allowed kernel names.
#
# By default, all installed kernels are allowed.
# c.KernelSpecManager.whitelist = traitlets.Undefined

c = get_config()

c.NotebookApp.ip = '*'

c.NotebookApp.open_browser = False

c.NotebookApp.port = 7000

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos
^X Exit          ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Linter    ^_ Go To Li
```

Figure 15

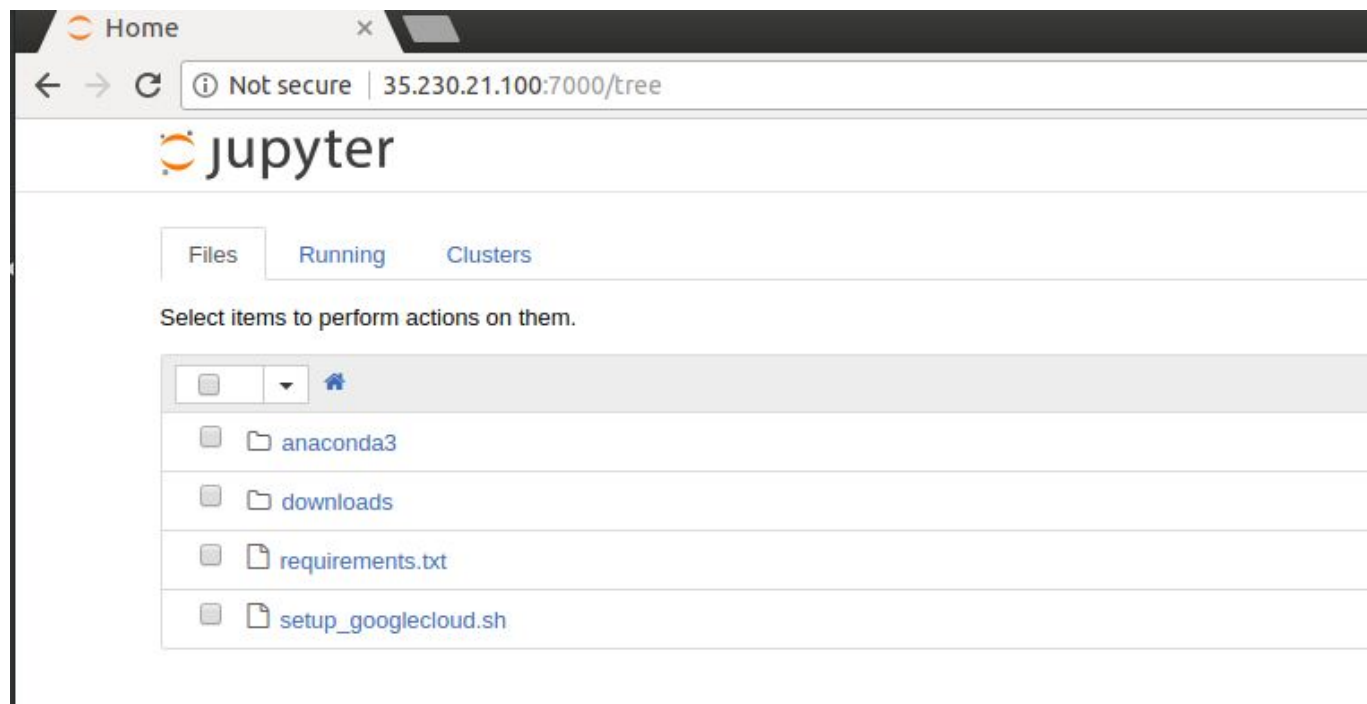
To run Jupyter, execute following command and you should see something like Figure 16:

- `jupyter-notebook --no-browser --port=7000`

```
(.env) pat@csulb-ml:~$ jupyter-notebook --no-browser --port=7000
[W 10:01:36.208 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not
[W 10:01:36.208 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using authentication. This is
a and not recommended.
[I 10:01:36.211 NotebookApp] Serving notebooks from local directory: /home/pat
[I 10:01:36.211 NotebookApp] 0 active kernels
[I 10:01:36.211 NotebookApp] The IPython Notebook is running at: http://[all ip addresses on your system]:7000/
[I 10:01:36.211 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[I 10:02:04.204 NotebookApp] 302 GET / 0.66ms
[I 10:02:41.259 NotebookApp] 302 GET / 0.38ms
```

Figure 16

Then go to your browser and type your external address (from previous section), follow by ":" and then port 7000 (Figure 17). If you see the page like in the Figure 17, that means your Jupyter Notebook is working!



**Figure 17**