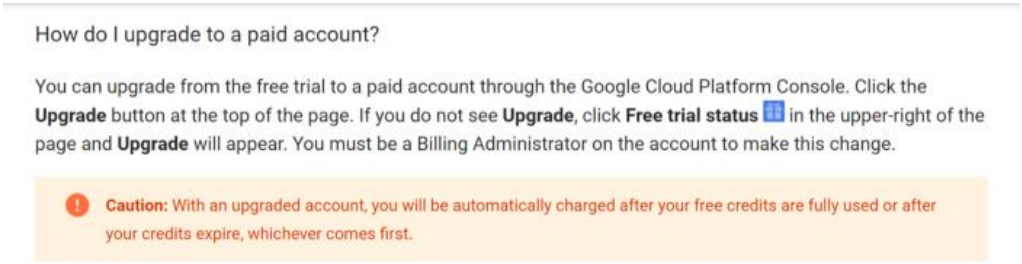
Setup guide for Google Cloud Platform

This guide explains how to set up Google Cloud Platform (GCP) to use PyTorch 1.0.0 and fastai 1.0.2. At the end of this tutorial you will be able to use both in a GPU-enabled Jupyter Notebook environment.

1. Sign up on Google Cloud Platform Free Tier.

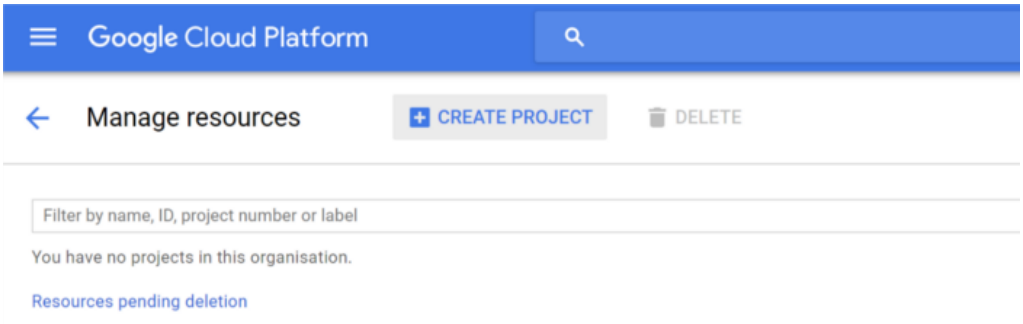
https://cloud.google.com/free/

2. Upgrade to a paid account (credit card required).



If you are unable to find an Upgrade button at this point, just continue with the next steps. The Upgrade button should appear at Step 5 below when you request for an increase in quota.

3. Create a new project for the deep learning course course



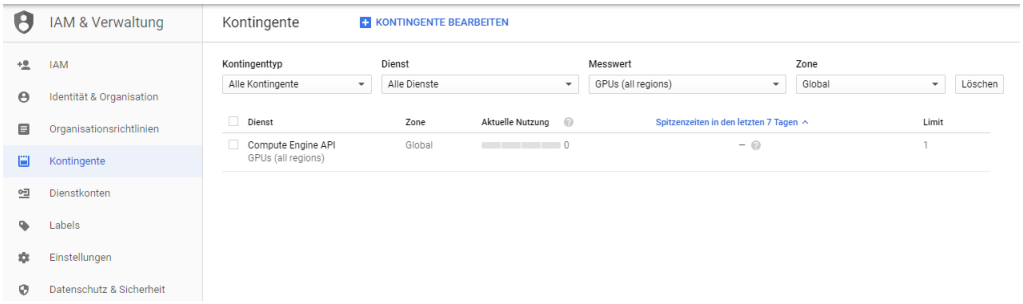
Click on “Create Project”.

4. Request for increase in quota for GPU

IAM & Admin → Quotas

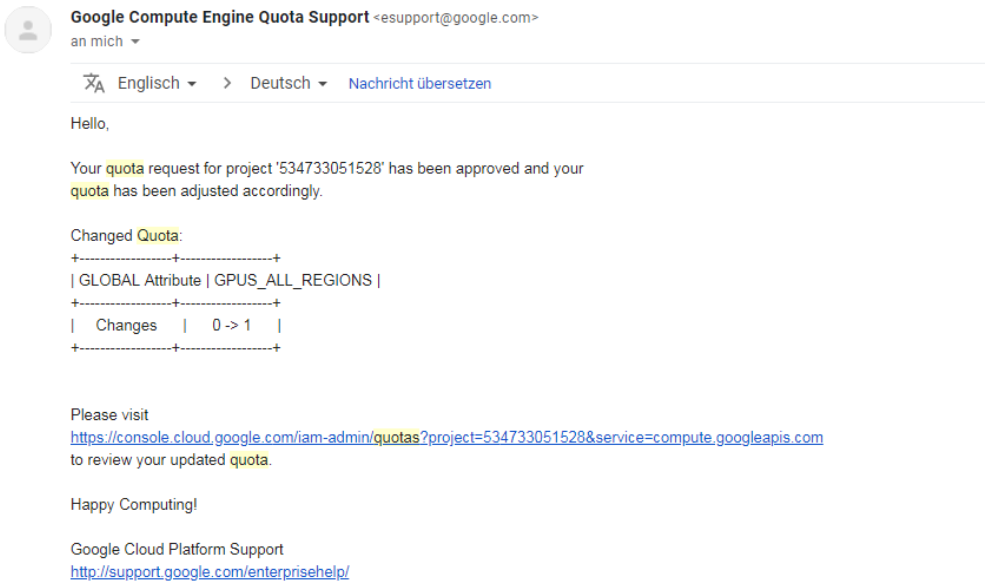
Kommentar [DHN1]: If you can't increase your quota, skip this part first and continue with step 7: create new instance. You won't be able to add a GPU, but now you can request for an increase in quota for GPU .

5. In filter type, select metric to be GPUs (all regions) and Location as Global.



Click edit quotas and select the quota to edit (GPUs All Regions). Set the new quota limit to 1 or more. Your request may require confirmation, which Google claims typically takes two business days to get.

6. Receive email approval of quota increase



Google Compute Engine Quota Support <esupport@google.com>
an mich

Englisch › Deutsch Nachricht übersetzen

Hello,

Your quota request for project '534733051528' has been approved and your quota has been adjusted accordingly.

Changed Quota:

```
+-----------------+-----------------+
| GLOBAL Attribute | GPUS_ALL_REGIONS |
+-----------------+-----------------+
|    Changes      |     0 -> 1      |
+-----------------+-----------------+
```

Please visit
https://console.cloud.google.com/iam-admin/quotas?project=534733051528&service=compute.googleapis.com
to review your updated quota.

Happy Computing!

Google Cloud Platform Support
http://support.google.com/enterprisehelp/

7. Navigate to the Compute Engine -> VM-instances

Create new instance

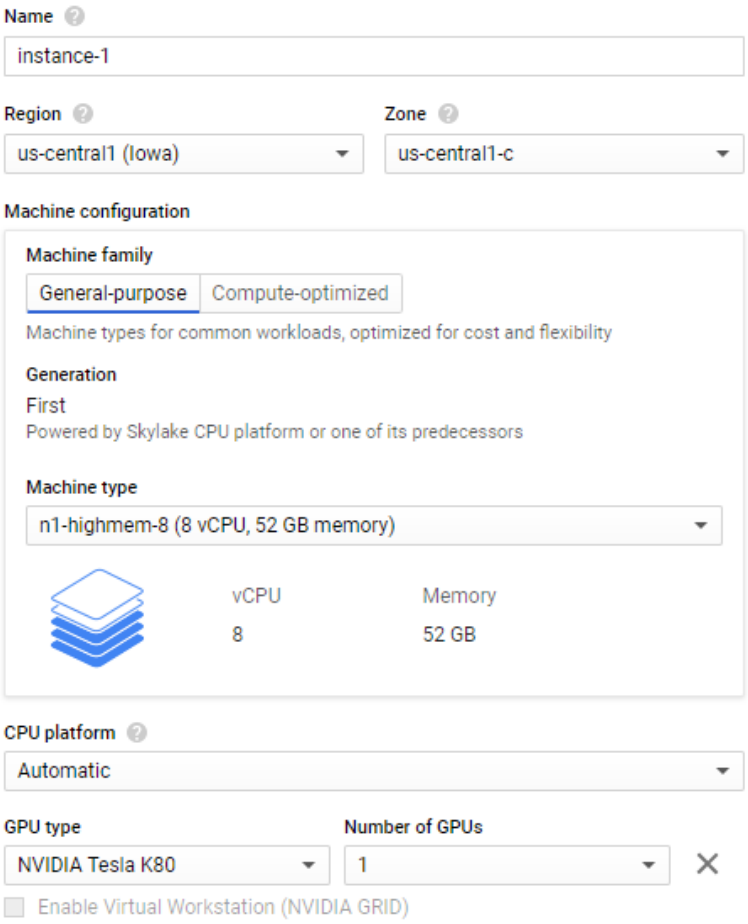
Give a name for your instance

Region: us-central1, Zone: us-central1-c

Machine type: n1-highmem-8

GPU type: NVIDIA Tesla P4

Kommentar [DHN2]: Image should show the Test P4



Boot disk: Deep Learning Image: PyTorch 1.1.0 and fastai m30

Boot disk type: Standard persisten disk with 200GB

Boot disk

Select an image or snapshot to create a boot disk; or attach an existing disk

OS images | Application images | Custom images | Snapshots | Existing disks

built on 2019-06-17

- SUSE Linux Enterprise Server 15 SP1For SAP x86_64
  built on 2019-06-25
- Windows Server version 1803 Datacenter Core for Containers
  Server Core, x64 built on 20190620
- Windows Server version 1803 Datacenter Core
  Server Core, x64 built on 20190620
- Windows Server version 1809 Datacenter Core for Containers
  Server Core, x64 built on 20190620
- Windows Server version 1809 Datacenter Core
  Server Core, x64 built on 20190620
- Windows Server 2008 R2 Datacenter
  Server with Desktop Experience, x64 built on 20190620
- Windows Server 2012 R2 Datacenter Core
  Server Core, x64 built on 20190620
- Windows Server 2012 R2 Datacenter
  Server with Desktop Experience, x64 built on 20190620
- Windows Server 2016 Datacenter Core
  Server Core, x64 built on 20190620
- Windows Server 2016 Datacenter
  Server with Desktop Experience, x64 built on 20190620
- Windows Server 2019 Datacenter Core for Containers
  Server Core, x64 built on 20190620
- Windows Server 2019 Datacenter Core
  Server Core, x64 built on 20190620
- Windows Server 2019 Datacenter for Containers
  x64 built on 20190620
- Windows Server 2019 Datacenter
  Server with Desktop Experience, x64 built on 20190620
- Deep Learning Image: Base m30 (with CUDA 10.0)
  A Debian based image with CUDA 10.0.
- (•) Deep Learning Image: PyTorch 1.1.0 and fastai m30
  PyTorch 1.1.0 (and fastai) with CUDA 10.0 and Intel® MKL-DNN, Intel® MKL.
- Deep Learning Image: TensorFlow 1.14.0 m30
  TensorFlow 1.14.0 with CUDA 10.0 and Intel® MKL-DNN, Intel® MKL.
- Debian GNU/Linux 9 Stretch + TF 1-11
  A Debian linux image with Tensorflow Version 1-11 pre-installed and optimized for Cloud TPUs.
- Debian GNU/Linux 9 Stretch + TF 1-12
  A Debian linux image with Tensorflow Version 1-12 pre-installed and optimized for Cloud TPUs.
- Debian GNU/Linux 9 Stretch + TF 1-13
  A Debian linux image with Tensorflow Version 1-13 pre-installed and optimized for Cloud TPUs.
- Debian GNU/Linux 9 Stretch + TF 1-14-1-dev20190508
  A Debian linux image with Tensorflow Version 1-14-1-dev20190508 pre-installed and optimized for Cloud TPUs.
- Debian GNU/Linux 9 Stretch + TF 1-14-1-dev20190518
  A Debian linux image with Tensorflow Version 1-14-1-dev20190518 pre-installed and optimized for Cloud TPUs.
- Debian GNU/Linux 9 Stretch + TF 1-14-1-dev20190619
  A Debian linux image with Tensorflow Version 1-14-1-dev20190619 pre-installed and optimized for Cloud TPUs.
- Debian GNU/Linux 9 Stretch + TF 1-14
  A Debian linux image with Tensorflow Version 1-14 pre-installed and optimized for Cloud TPUs.
- Debian GNU/Linux 9 Stretch + TF nightly
  A Debian linux image with Tensorflow Version nightly pre-installed and optimized for Cloud TPUs.
- Debian GNU/Linux 9 Stretch + PyTorch/XLA
  A Debian linux image with PyTorch/XLA pre-installed and optimized for Cloud TPUs.

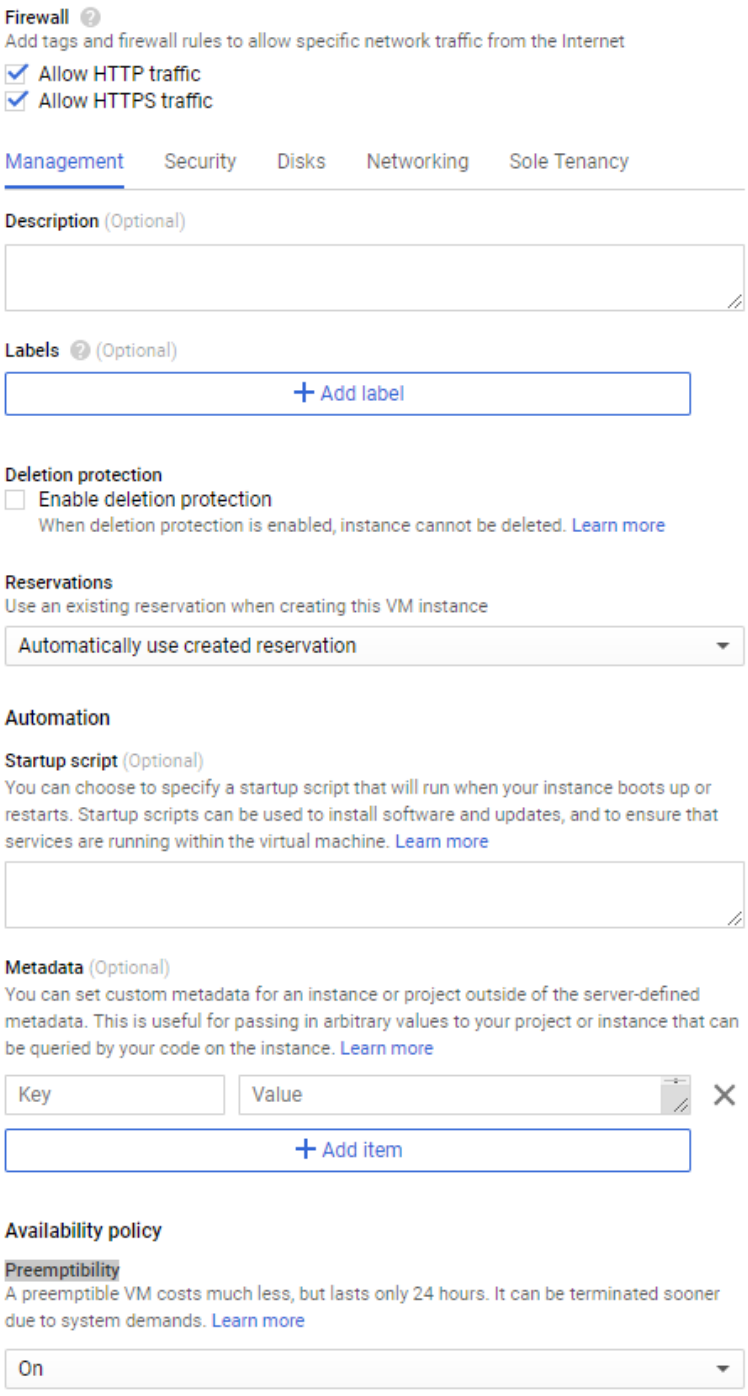
Can't find what you're looking for? Explore hundreds of VM solutions in Marketplace

Boot disk type (?) | Size (GB) (?)

Standard persistent disk | 200

Firewall: Allow HTTP and HTTPS traffic

Preemptibility: On

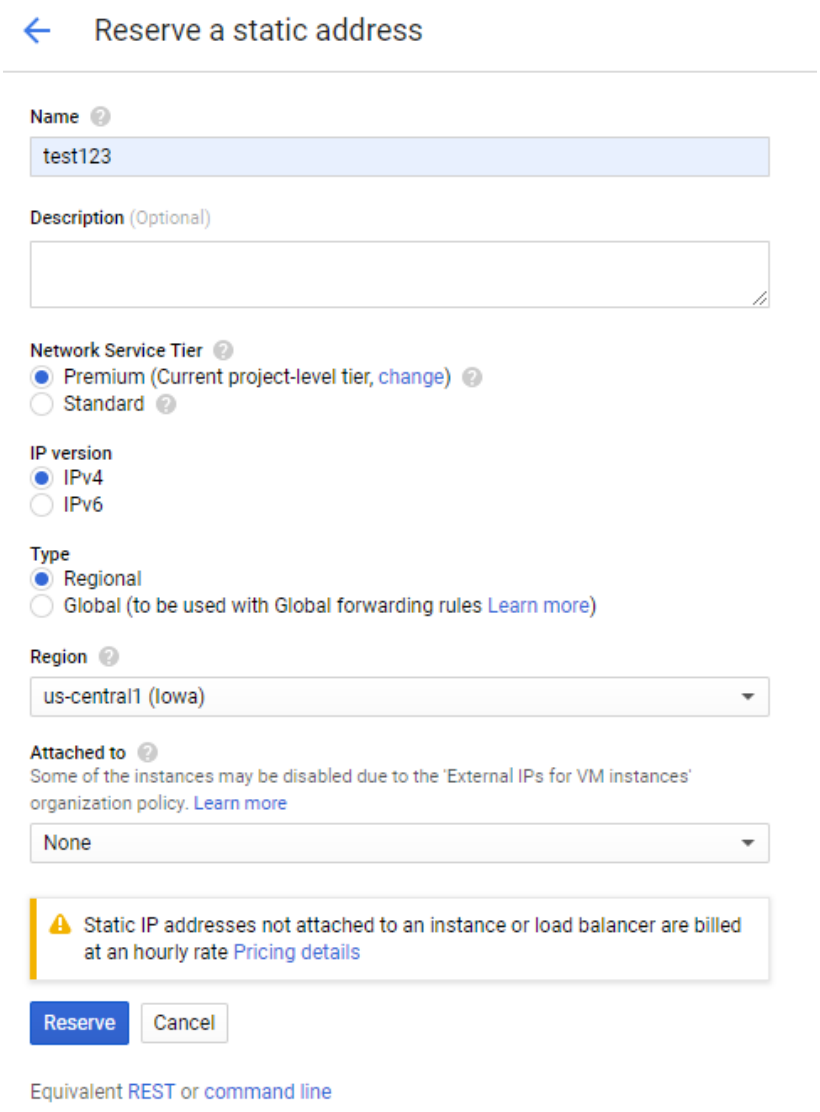


Price of your VM instace should be about $0.327 hourly

8. Create a static IP address

Click on the three dots besides your created VM instance and select ‘View network details

Navigate to external IP addresses and select reserve a static address



Reserve and click on change to attach IP address to your instance

9. Connect to the CM instance through SSH from the browser:

10. Install Nvidia driver

11. Configure Jupyter Notebook Server

Type in terminal:

jupyter notebook --generate-config

cd .jupyter

open your jupyter notebook config with editor of your chose, e.g. vim jupyter_notebook_config.py

```
x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

This VM requires Nvidia drivers to function correctly.   Installation takes ~1 minute.
Would you like to install the Nvidia driver? [y/n] y
Installing Nvidia driver.
Downloading driver from GCS location gs://nvidia-drivers-us-public/tesla/410.104/NVIDIA-Linux-x86_64-410.104.run
Verifying archive integrity... OK
Uncompressing NVIDIA Accelerated Graphics Driver for Linux-x86_64 410.104.....................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.............................

WARNING: The nvidia-drm module will not be installed. As a result, DRM-KMS will not function with this
         installation of the NVIDIA driver.


WARNING: nvidia-installer was forced to guess the X library path '/usr/lib' and X module path
         '/usr/lib/xorg/modules'; these paths were not queryable from the system.  If X fails to find the NVIDIA X
         driver module, please install the `pkg-config` utility and the X.Org SDK/development package for your
         distribution and reinstall the driver.


WARNING: Unable to find a suitable destination to install 32-bit compatibility libraries. Your system may not be
         set up for 32-bit compatibility. 32-bit compatibility files will not be installed; if you wish to install
         them, re-run the installation and set a valid directory with the --compat32-libdir option.

Nvidia driver installed.
nam@instance-test12:~$ jupyter notebook --generate-config
Writing default config to: /home/nam/.jupyter/jupyter_notebook_config.py
nam@instance-test12:~$ cd ./jupyter
-bash: cd: ./jupyter: No such file or directory
nam@instance-test12:~$ cd .jupyter
nam@instance-test12:~/.jupyter$ vim jupyter_notebook_config.py
nam@instance-test12:~/.jupyter$
```

add somewhere following lines:

c.NotebookApp.ip = '0.0.0.0'

c.NotebookApp.port = 8888

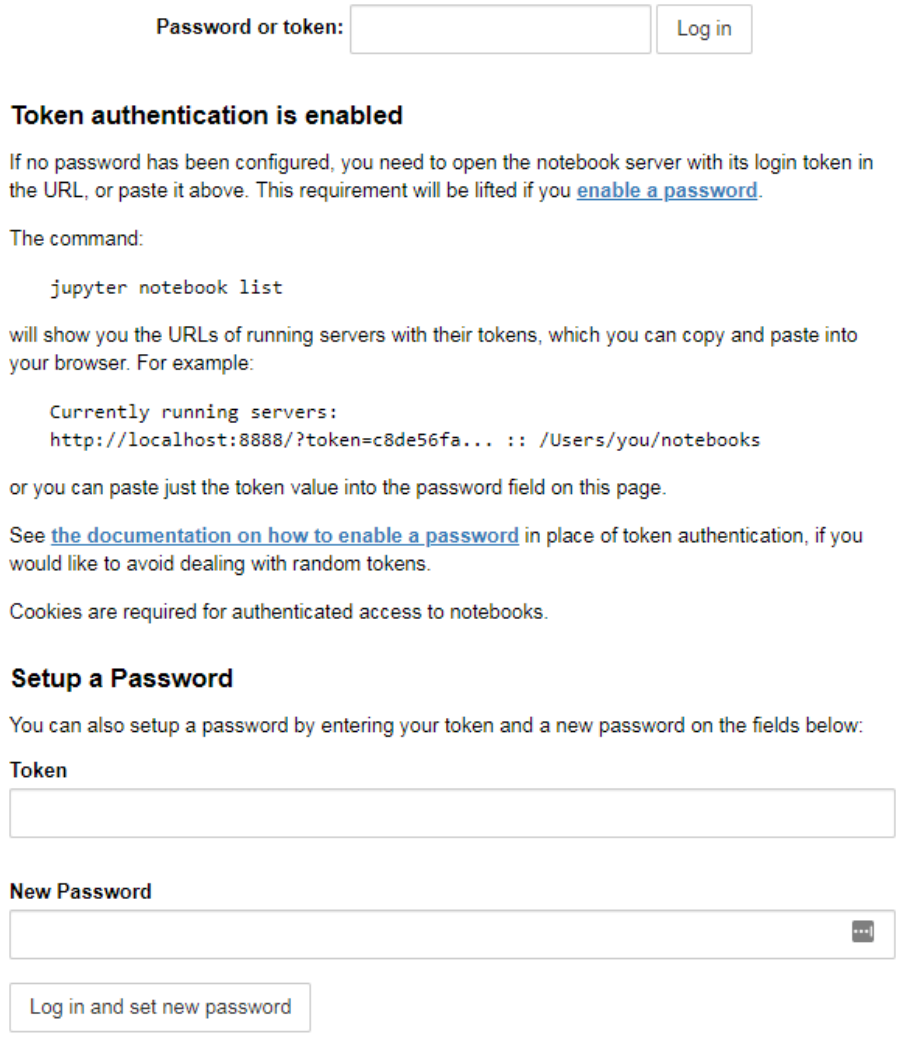
```
# Configuration file for jupyter-notebook.

#------------------------------------------------------------------------------
# Application(SingletonConfigurable) 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
#c.Application.log_format = '[%(name)s]%(highlevel)s %(message)s'

## Set the log level by value or name.
#c.Application.log_level = 30

#------------------------------------------------------------------------------
# JupyterApp(Application) configuration
#------------------------------------------------------------------------------

c.NotebookApp.ip = '0.0.0.0'
c.NotebookApp.port = 8888

## Base class for Jupyter applications

## Answer yes to any prompts.
#c.JupyterApp.answer_yes = False

## Full path of a config file.
#c.JupyterApp.config_file = ''

## Specify a config file to load.
#c.JupyterApp.config_file_name = ''

## Generate default config file.
#c.JupyterApp.generate_config = False

#------------------------------------------------------------------------------
# NotebookApp(JupyterApp) configuration
#------------------------------------------------------------------------------
"jupyter_notebook_config.py" 769L, 29412C                     24,0-1        Top
```

12. Start Jupyter Notebook

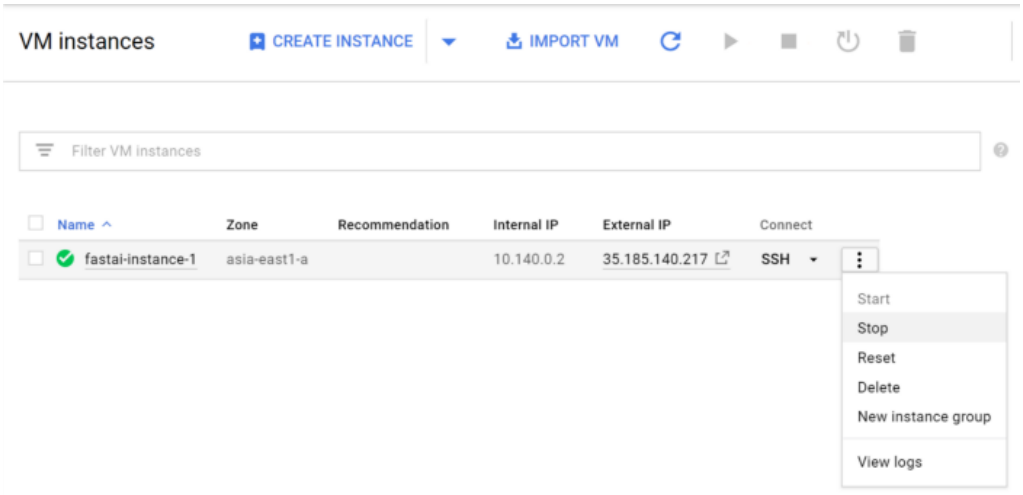
Type in terminal jupyter notebook and copy your token

Type in your browser address bar “external-ip-of-your_instance:8888”

Paste your Token and create new Password



13. Important: Remember to stop your VM instance when you are done with your notebooks or you will continue to incur charges.


VM instances
CREATE INSTANCE
IMPORT VM
Filter VM instances
Name
Zone
Recommendation
Internal IP
External IP
Connect
fastai-instance-1
asia-east1-a
10.140.0.2
35.185.140.217
SSH
Start
Stop
Reset
Delete
New instance group
View logs