Setting up Deep Learning Environment on AWS & GCP

CSCI 599

2019-02-12

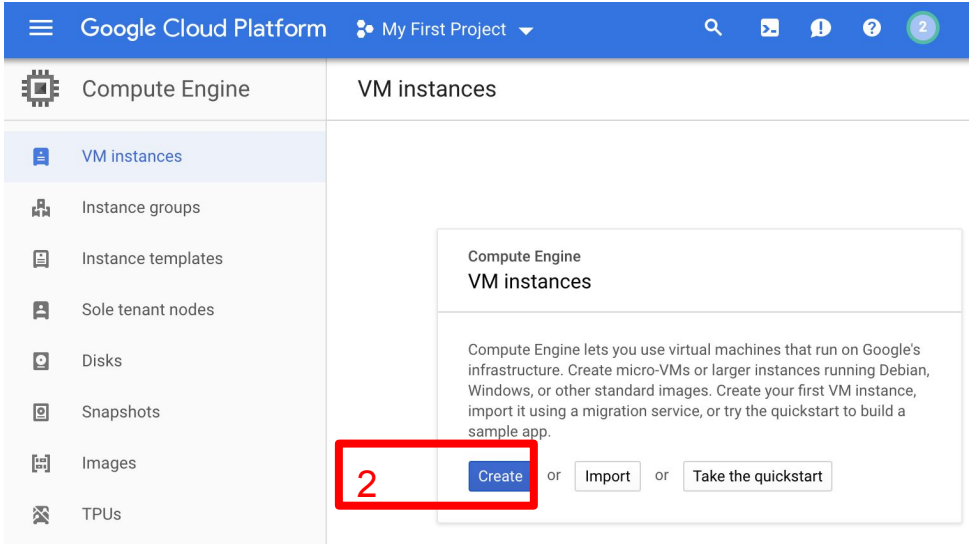
AWS & GCP are not free!
GPU Instances are expensive!
You are responsible for all the billings!
Even if you are not running processes, they will charge you if your machine is running.
Remember to shutdown/terminate your machines when not using them.

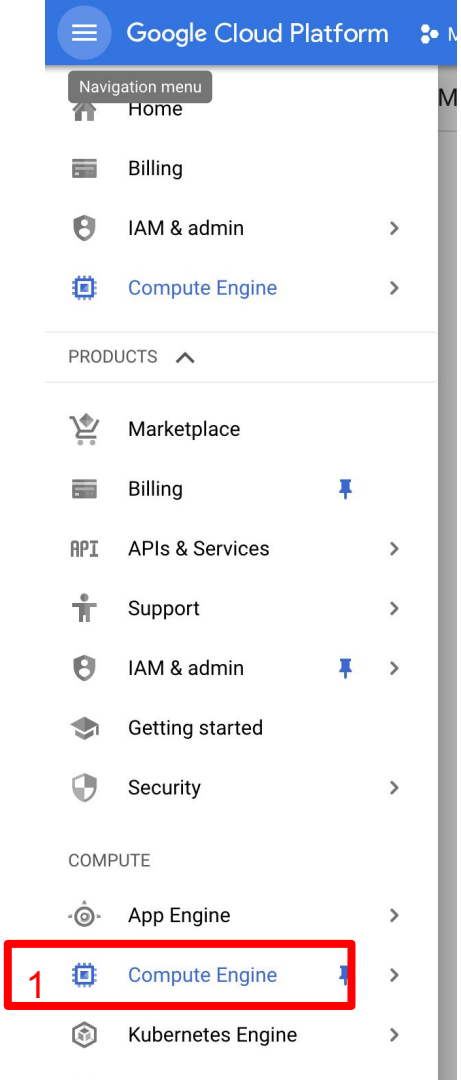
In case you cannot create a GPU instance

- By default, AWS/GCP don’t allow you to create GPU instances (GPU limits = 0)
- You need to increase the GPU limits
- The instructions are https://docs.google.com/presentation/d/1iZQ_KuwdYDdZkpBjmWRahP1NzIPFS8he-qOySqRPumQ/edit#slide=id.p

GCP

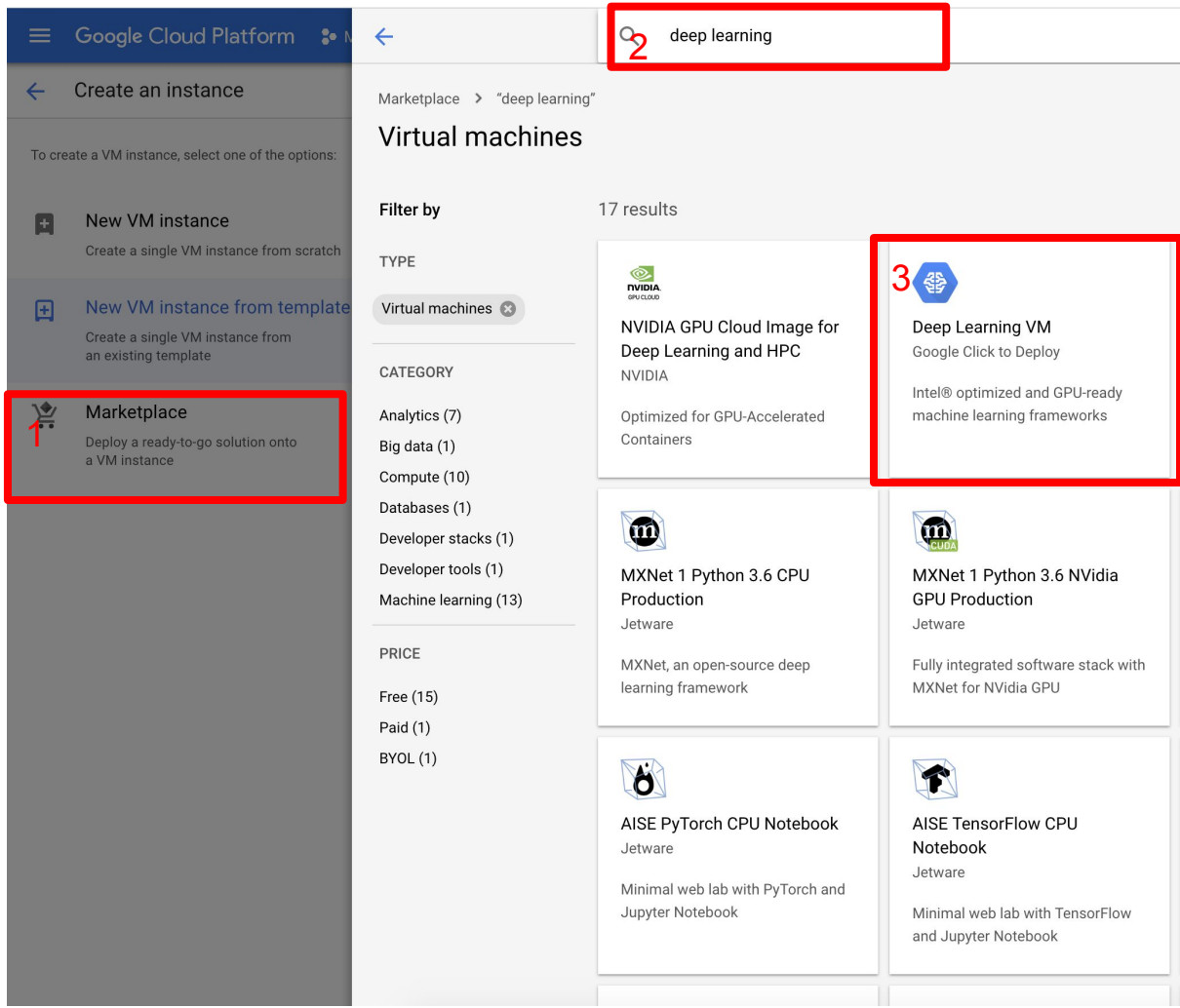
- Create a project if you haven't
- Go to "**Compute Engine**"
- Click "Create" to create a VM





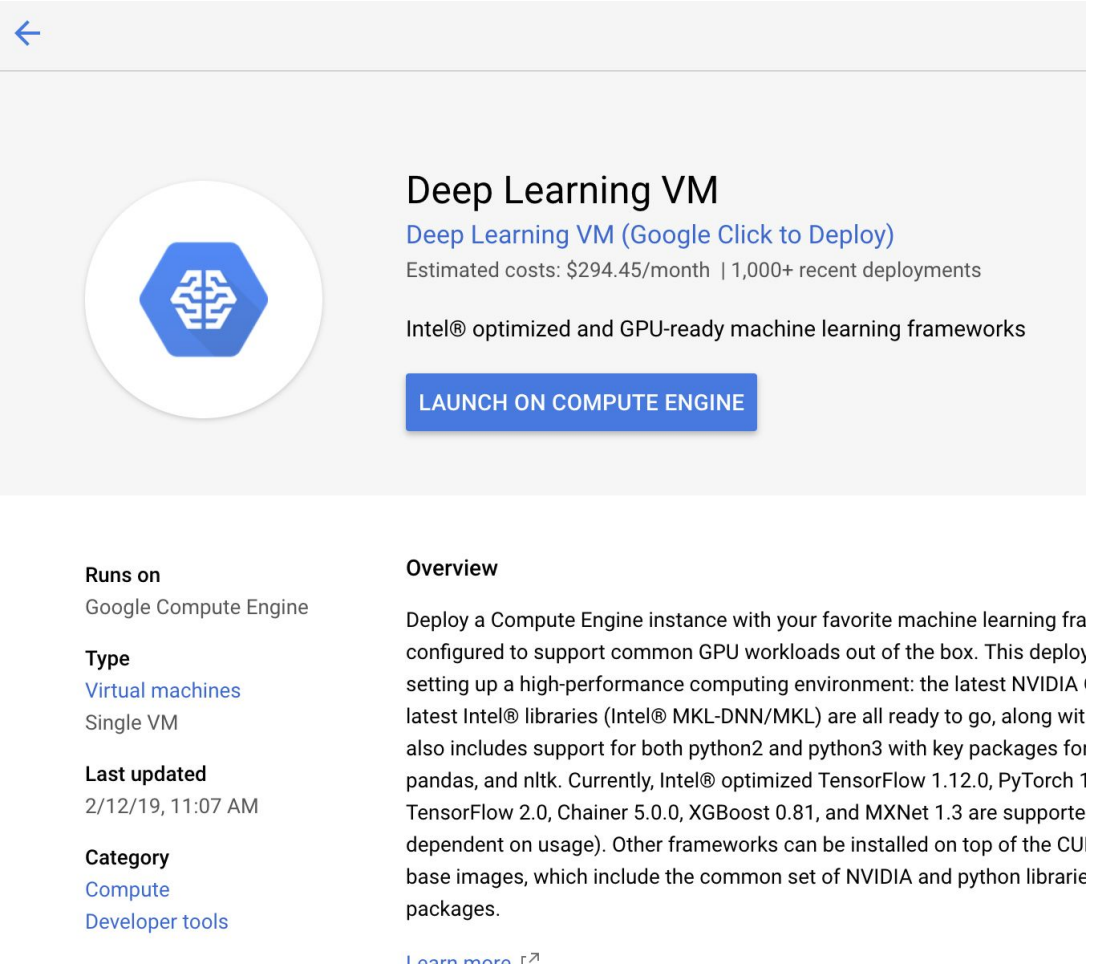
GCP

- Click “**Marketplace**”
- search “**deep learning**”
- choose “**Deep Learning VM**”



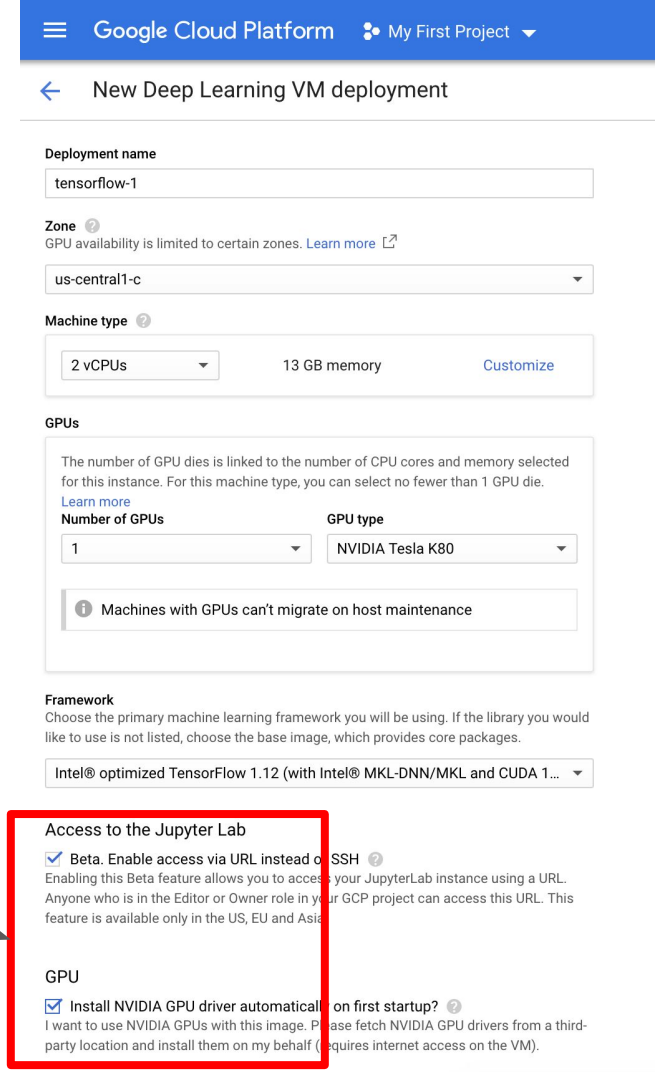
GCP

- click “**LAUNCH ON COMPUTE ENGINE**”



GCP

- if you want, you can
  - update the name
  - update the zone
  - update number of CPUs
  - change disk size
  - change network
- you can also
  - change the number of GPUs and the GPU type
  - change a different framework
- **remember to select**
  - Beta. Enable access via URL...
  - Install NVIDIA GPU Driver…
- review the terms and then click “**Deploy**” at the bottom of the page



GCP

you server is being deployed (it takes time)

Google Cloud Platform | My First Project

Deployment Manager

Deployments

Type registry

tensorflow-1 | STOP | DELETE

tensorflow-1 is being deployed | View details

Overview - tensorflow-1

tensorflow tensorflow.jinja

tensorflow-vm-tmpl vm_instance.py

tensorflow-1-vm vm instance

software-status software_status.py

tensorflow-1-config config

tensorflow-1-software config waiter

tensorflow

tensorflow has resource warnings

tensorflow-1-vm: Disk size: '100 GB' is larger than image size: '30 GB'. You might need to resize the root repartition manually if the operating system does not support automatic resizing. See https://cloud.google.com/compute/docs/disks/persistent-disks#repartitionrootpd for details.

Deep Learning VM

Solution provided by Google Click to Deploy

Instance	Pending
Instance zone	Pending
Instance machine type	Pending

More about the software

Get started with Deep Learning VM

You will be able to use Deep Learning VM after the deployment is completed.

Documentation

Official Documentation

StackOverflow: Deep Learning VM

Google Group: Deep Learning VM

Support

If you have non-framework related issues, you can bring them up at the Deep Learning VM Stack Overflow.

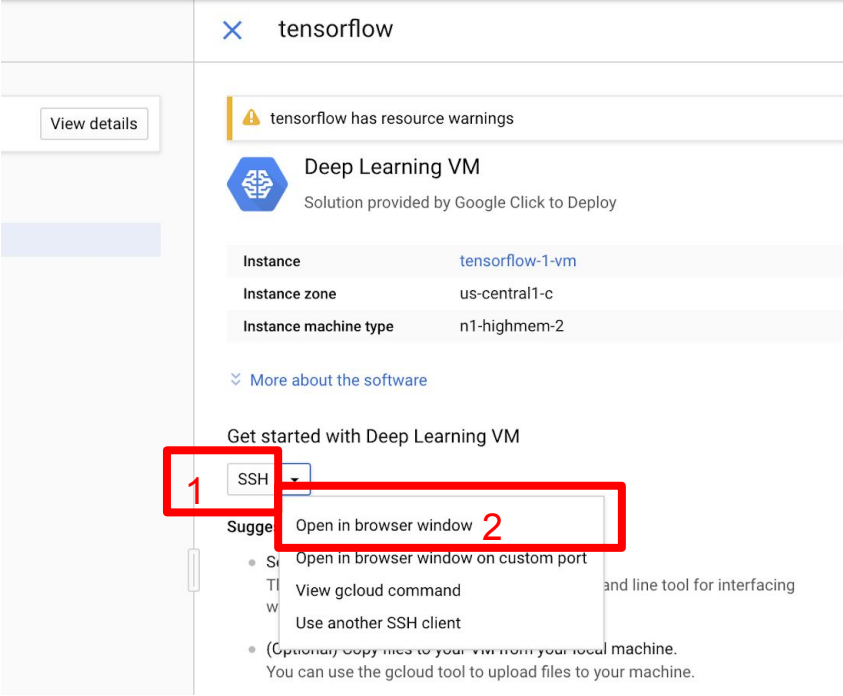
Template properties

More

GCP

Once it's done, you can connect to your server by

- click the "**SSH**" button under "Get started with Deep Learning VM"
- click "**Open in browser window**"



You can also find the SSH button in the "**VM instance detail**" page

GCP

After you connect successfully,

- type “nvidia-smi” if you are using a GPU instance and you can see
- type “python3 -V” to check the python version
- then you can upload your code and run your scripts.

```
please use the binaries that are pre-built for this image. You can find the bina
ries at
/opt/deeplearning/binaries/tensorflow/
If you need to install a different version of Tensorflow manually, use the commo
n Deep Learning image with the
right version of CUDA

Linux tensorflow-1-vm 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) 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.
hanpeng_liu_cs@tensorflow-1-vm:~$ nvidia-smi
Tue Feb 12 23:59:56 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 410.72       Driver Version: 410.72       CUDA Version: 10.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla K80           Off  | 00000000:00:04.0 Off |                    0 |
| N/A   33C    P0    62W / 149W |      0MiB / 11441MiB |    100%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
hanpeng_liu_cs@tensorflow-1-vm:~$ python3 -V
Python 3.5.3
```

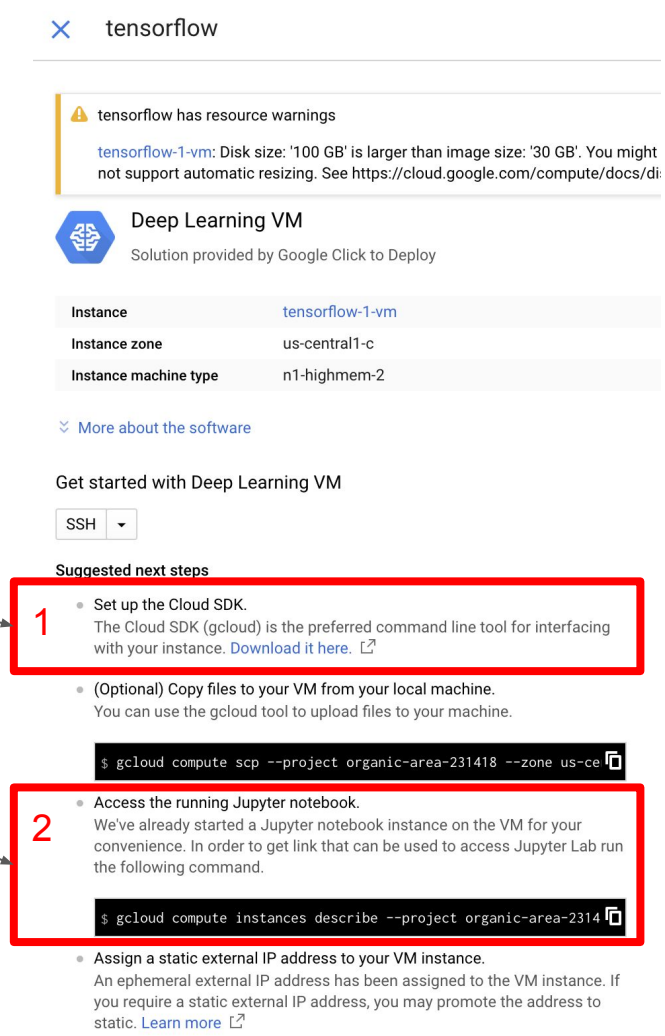
Tips

- It’s highly recommended to run your code in **tmux or screen**, so that you can detach the window and reattach to the terminal window later.
- It’s highly recommended to save checkpoints when you train your model, so that they can be resumed after unexpected program halts.

- To upload files to GCP, you can check the tutorial given by Google https://cloud.google.com/compute/docs/instances/transfer-files

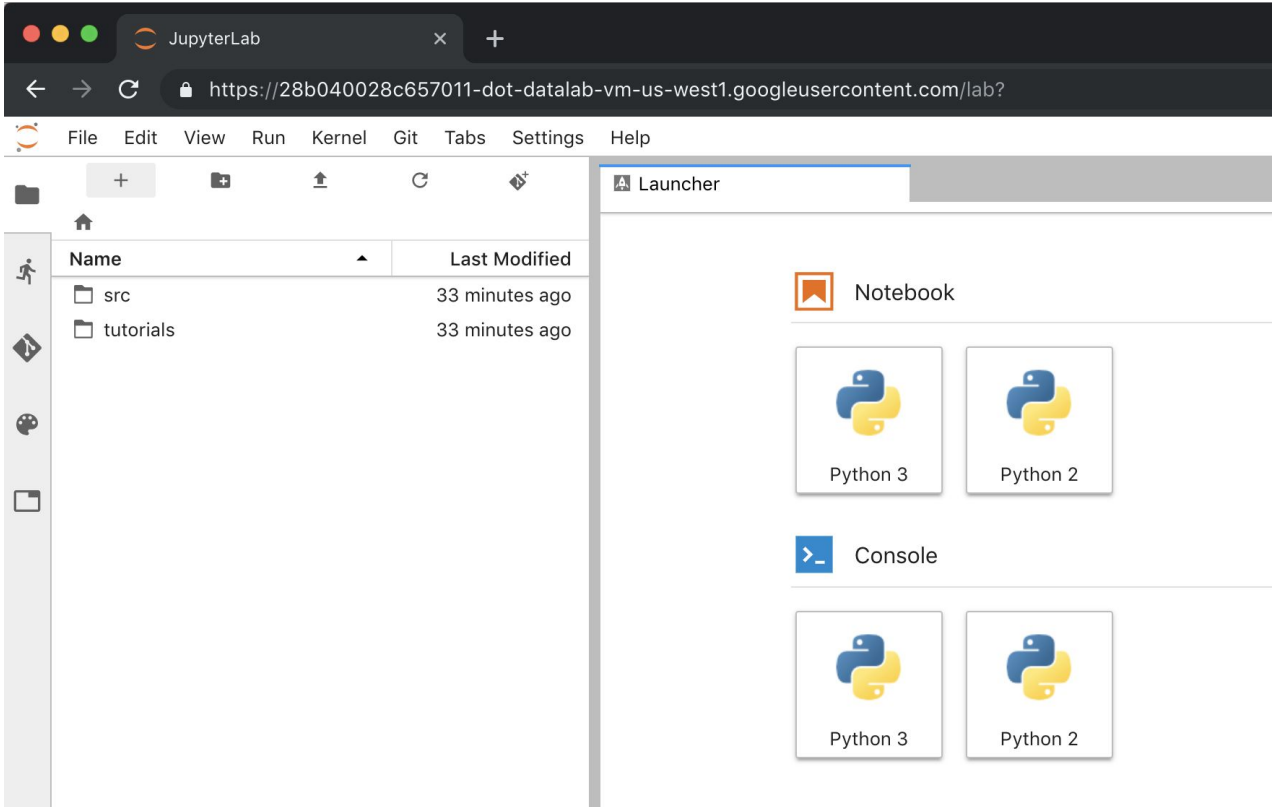
GCP Jupyter Notebook

- The VM instance has set up a jupyter notebook,
- in order to access it, you need to
  - install Google Cloud SDK
  - once installed, run “gcloud auth login”
  - run the corresponding command shown in the deployment page
  - it will give you a link
  - open the link in browser
    - if it shows login error, you can try open it in a private window



GCP Jupyter Notebook

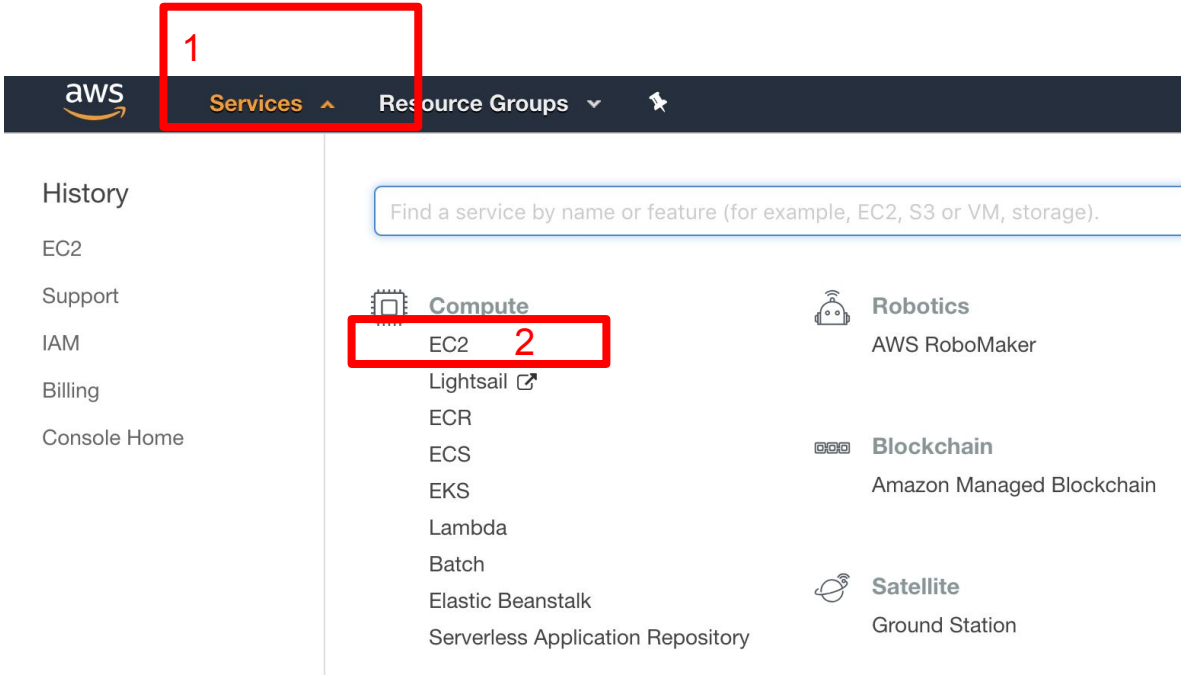
You should be able to open the JupyterLab website,



AWS & GCP are not free!
GPU Instances are expensive!
You are responsible for all the billings!
Even if you are not running processes, they will charge you if your machine is running.
Remember to shutdown/terminate your machines when not using them.

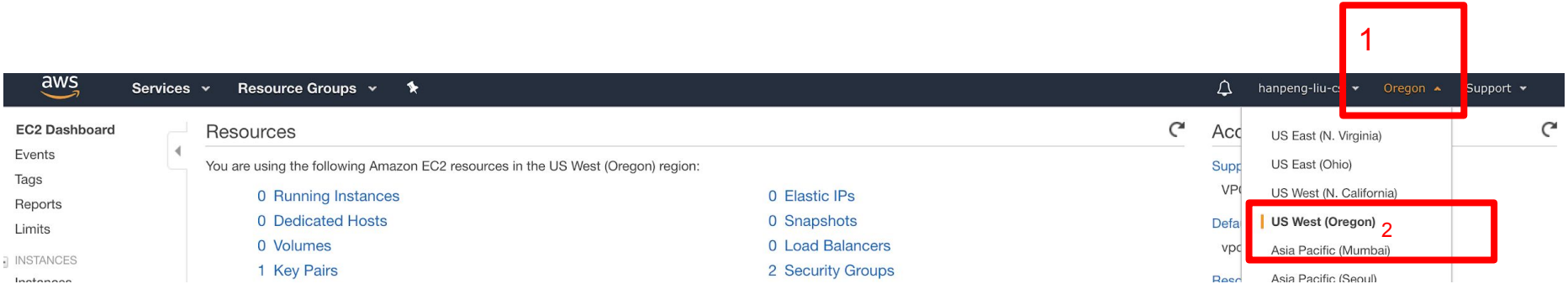
AWS

- login to AWS console
- click “**Services**”
- then click “**EC2**”



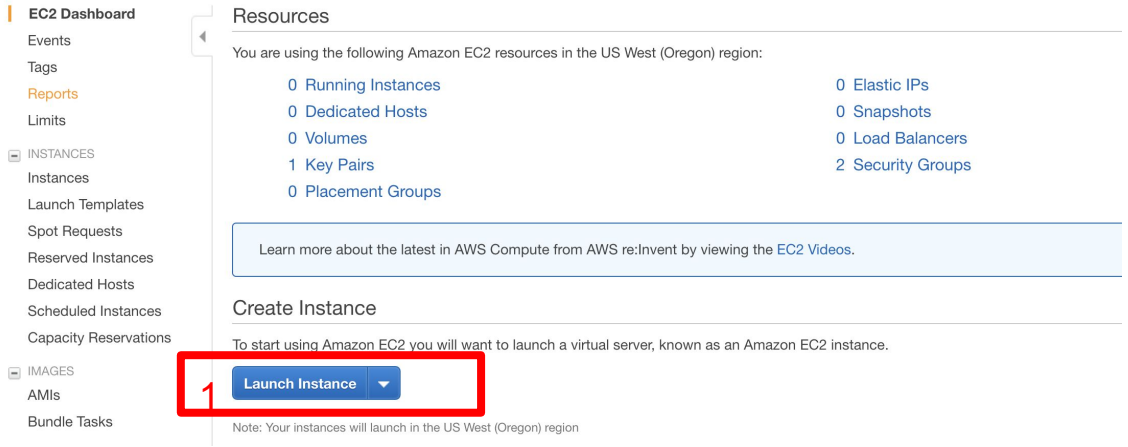
AWS

- hover the left button to “Support” in the banner,
- click the desired region, for example, Oregon



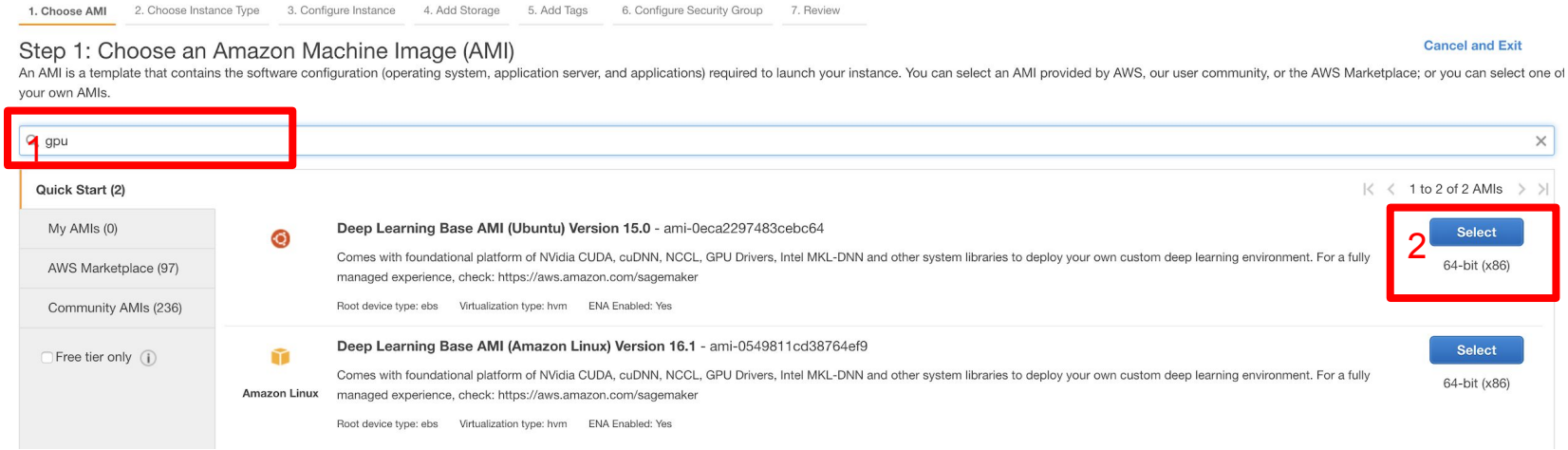
AWS

- click “Launch Instance”



AWS

- type “**gpu**” in the search bar and press the enter key
- select the “**Deep Learning Base AMI (Ubuntu) Version 15.0**”



AWS

- if (1) you have GPU limits updated and (2) you want to create a GPU machine, select **p2.xlarge** or p2.8xlarge or p2.16xlarge
  - Otherwise, you can choose other instances such as “t2.small”
- Click “**Review and Launch**”

☑	GPU instances	p2.xlarge	4	61	EBS only	Yes	Hi
☐	GPU instances	p2.8xlarge	32	488	EBS only	Yes	10 Gi
☐	GPU instances	p2.16xlarge	64	732	EBS only	Yes	25 Gi
☐	GPU instances	p3.2xlarge	8	61	EBS only	Yes	Up to 10
☐	GPU instances	p3.8xlarge	32	244	EBS only	Yes	10 Gi

Cancel Previous Review and Launch

AWS

- review and if everything is fine, click “**Launch**”

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

Your instance configuration is not eligible for the free usage tier

To launch an instance that's eligible for the free usage tier, check your AMI selection, instance type, configuration options, or storage devices. Learn more about free usage tier eligibility and usage restrictions.

Don't show me this again

AMI Details

Edit AMI

Deep Learning Base AMI (Ubuntu) Version 15.0 - ami-0eca2297483cebc64

Comes with foundational platform of NVidia CUDA, cuDNN, NCCL, GPU Drivers, Intel MKL-DNN and other system libraries to deploy your own custom deep learning environment. For a fully managed experience, check: https://aws.amazon.com/sagemaker

Root Device Type: ebs Virtualization type: hvm

Instance Type

Edit instance type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
p2.xlarge	11.75	4	61	EBS only	Yes	High

Security Groups

Edit security groups

Security group name launch-wizard-2
Description launch-wizard-2 created 2019-02-12T15:08:11.912-08:00

Type	Protocol	Port Range	Source	Description

This security group has no rules

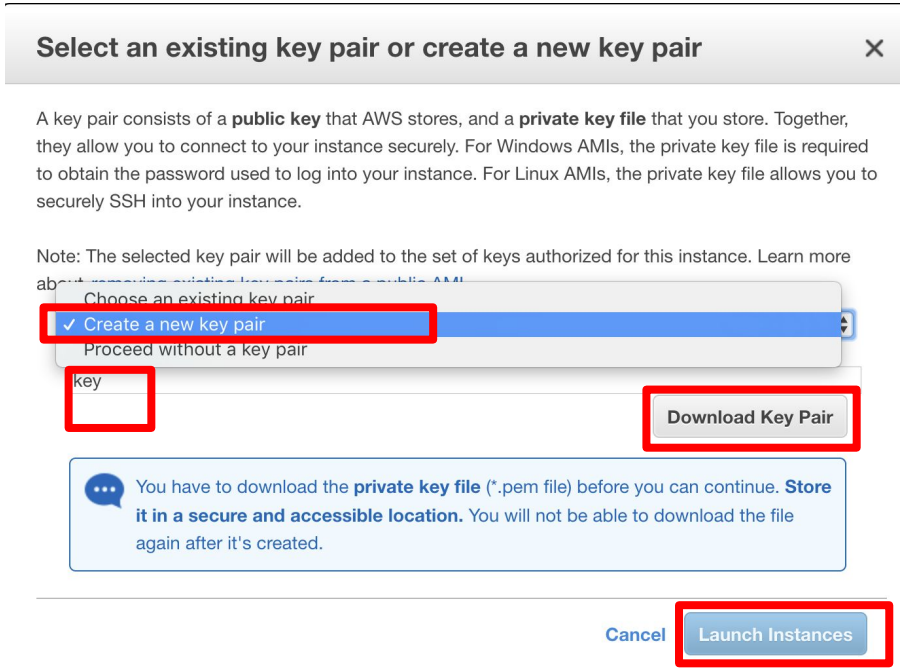
Instance Details

Edit instance details

Cancel Previous Launch

AWS

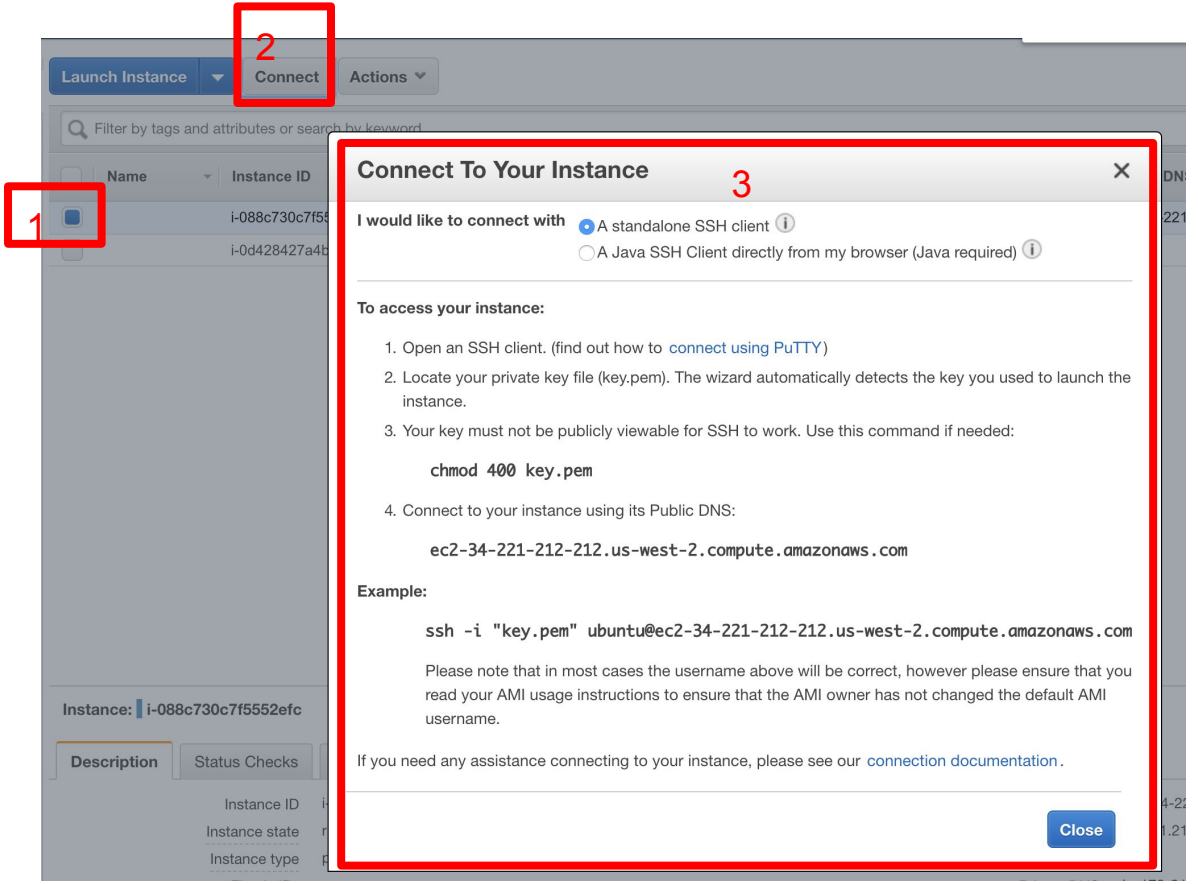
- choose “**Create a new key pair**”
- type the key pair name, e.g. “key”
- click “**Download Key Pair**”
- click “**Launch Instances**”



Then you will see your instance is launching, click “**View Instances**”

AWS

- select the instance you just create
- click “**connect**”
- choose a way suggested to connect your server



AWS

If you choose to use terminal SSH, assume your key file is stored in “~/Downloads/key.pem”

- first, go to the folder where your keys are,
  - e.g. “cd ~/Downloads”
- then, change the permission of the key file
  - e.g., “chmod 400 key.pem”
- then, connect through SSH
  - e.g., `ssh -i "key.pem" ubuntu@ec2-34-221-212-212.us-west-2.compute.amazonaws.com`
-

AWS

After you connect successfully,

- type “nvidia-smi” if you are using a GPU instance and you can see
- type “python3 -V” to check the python version
- then you can upload your code and run your scripts.

```
ubuntu@ip-172-31-42-170:~$ nvidia-smi
Tue Feb 12 23:18:23 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 410.79       Driver Version: 410.79       CUDA Version: 10.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla K80           On   | 00000000:00:1E.0 Off |                    0 |
| N/A   43C    P8    27W / 149W |      0MiB / 11441MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

```
ubuntu@ip-172-31-42-170:~$ python3 -V
Python 3.5.2
```

AWS

you can upload your files through “scp”, “rsync”, or “Filezilla”, or other programs.

Useful links:

- https://angus.readthedocs.io/en/2014/amazon/transfer-files-between-instance.html
- https://stackoverflow.com/questions/18169455/uploading-file-to-aws-from-local-machine
- https://www.google.com/search?q=how+to+upload+files+to+aws+ec2&oq=how+to+upload+files+to+aws+ec2&aqs=chrome..69i57j0l2.4679j0j4&sourceid=chrome&ie=UTF-8

Tips

- It’s highly recommended to run your code in **tmux or screen**, so that you can detach the window and reattach to the terminal window later.
- It’s highly recommended to save checkpoints when you train your model, so that they can be resumed after unexpected program halts.

- If you want to setup Jupyter Notebook, you can check the AWS’s tutorial https://docs.aws.amazon.com/dlami/latest/devguide/setup-jupyter.html

AWS & GCP are not free!
GPU Instances are expensive!
You are responsible for all the billings!
Even if you are not running processes, they will charge you if your machine is running.
Remember to shutdown/terminate your machines when not using them.