CS234 Azure GPU Setup Tutorial

BEFORE WE GET STARTED:

Please remember to stop VM instances if you are not using them!

(by clicking the Stop button at the top of the instance page)

Azure charges you per hour when the instance is running, no matter if you are using it or not. So you will soon run out of credits if you don't stop them when done.

Table of Content

1. Accept Invitation and Credits

2. Create a VM

3. Configure and Use the VM

4. Install CUDA, Tensorflow & Other dependencies

6. Transferring Files between VM and Local

7. Tensorboard

8. [Optional] Working under Virtual Environment

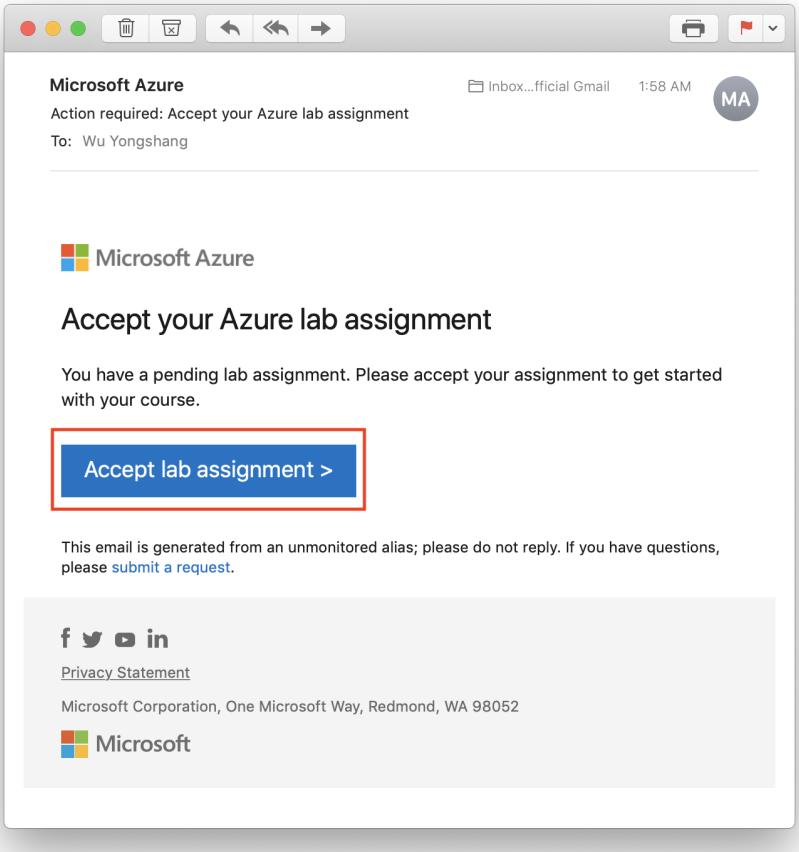
9. FAQ

The whole setup process is expected to take about **1 hour**. Most of the time is spent on downloading and installing libraries such as NVIDIA drivers, CUDA, cuDNN and Tensorflow. Don't worry as we have scripts written to do all the heavy lifting for you.

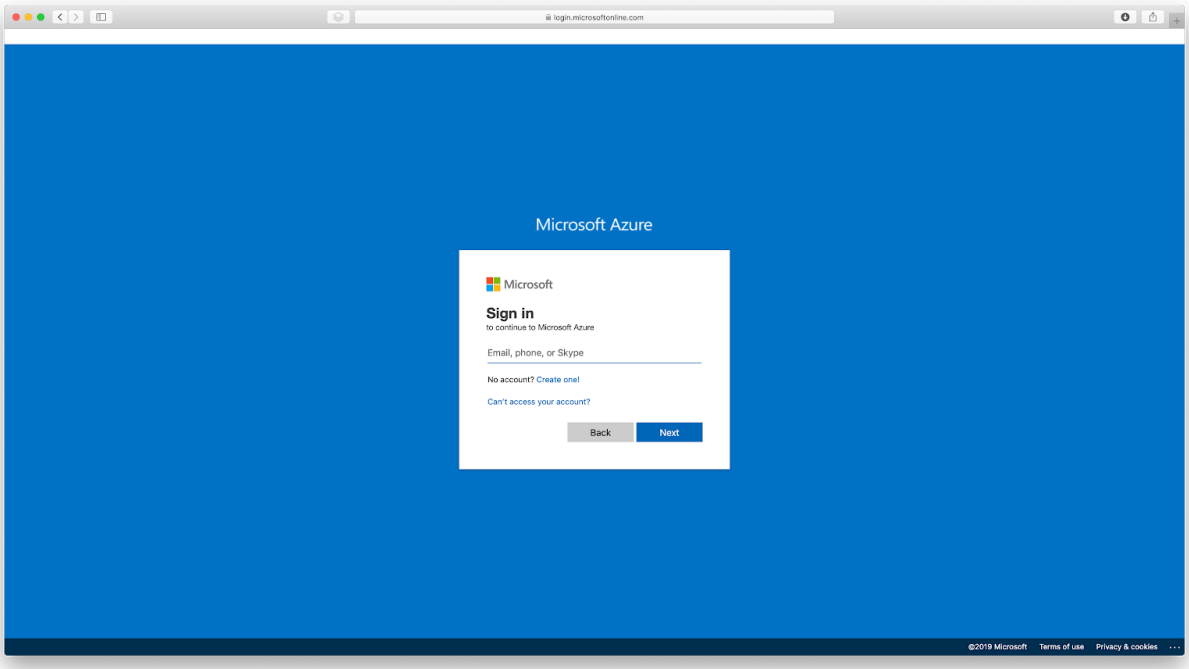
Take a deep breath and be patient :)

1. Accept Invitation and Credits

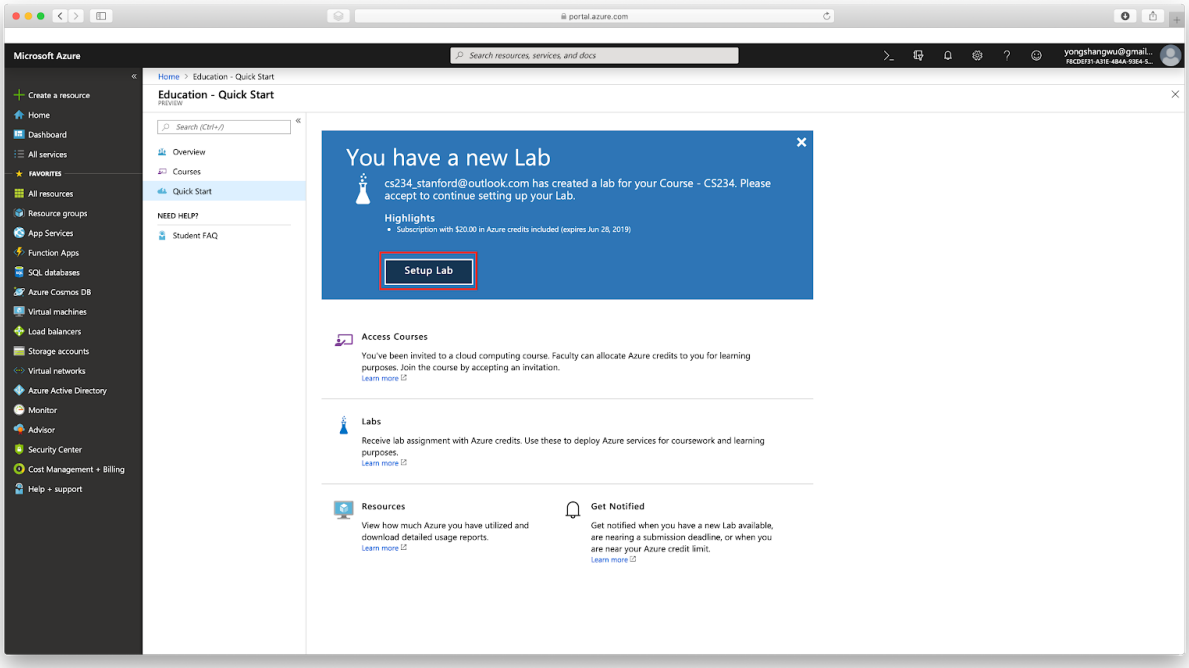
(1) You should have received an email to your Stanford mailbox, by Wed 23:59 p.m. Jan 23, with an invitation to join the Azure sponsorship subscription with $50 credits. If not, please reach out to wuy@stanford.edu directly for help, with the title starting with [CS234 Azure]. In the email you get, click “Accept lab assignment”:



(2) Sign in with **the email address that received this invitation** (if you never created an Azure account with this email before, you will need to create one first):



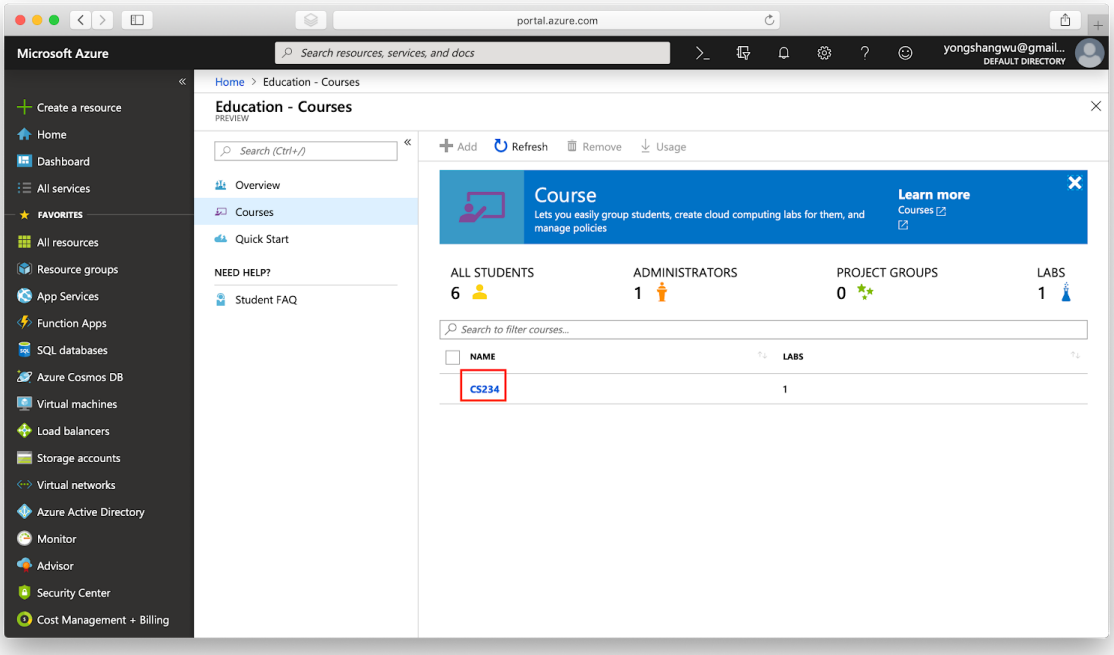
(3) Once logged in, click “Setup Lab”



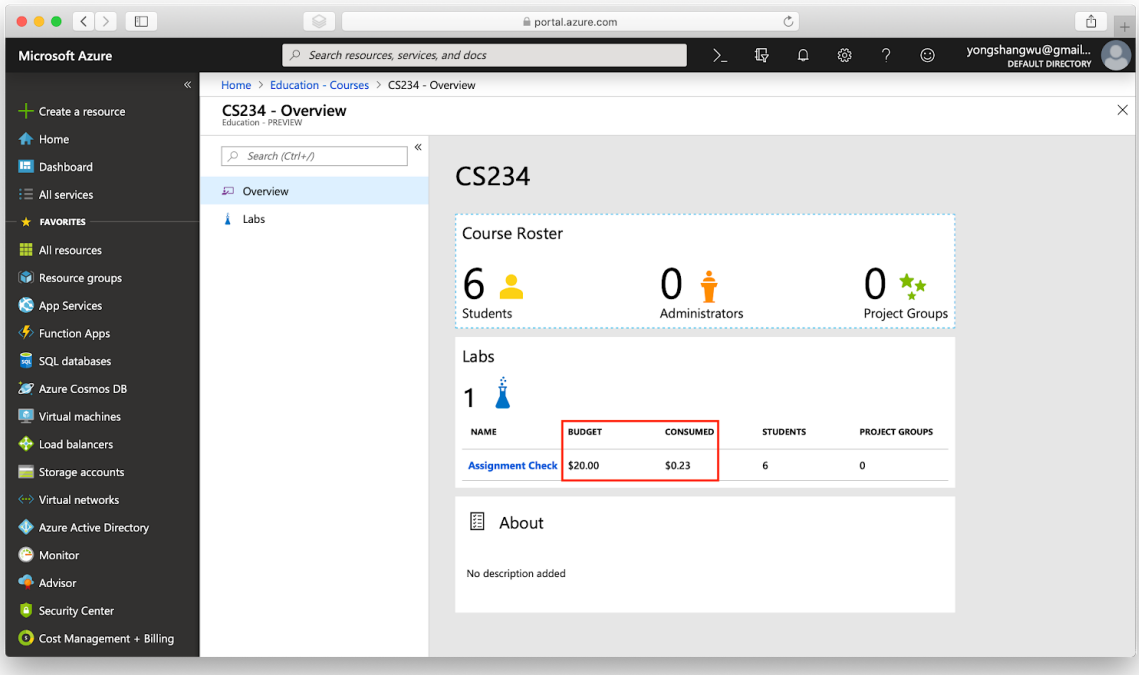
Fill out the required information and agree to the terms to continue.

(4) Check your credit balance

To check your credit balance, go to https://portal.azure.com/#blade/Microsoft_Azure_Education/EducationMenuBlade/classrooms, click course “CS234”:

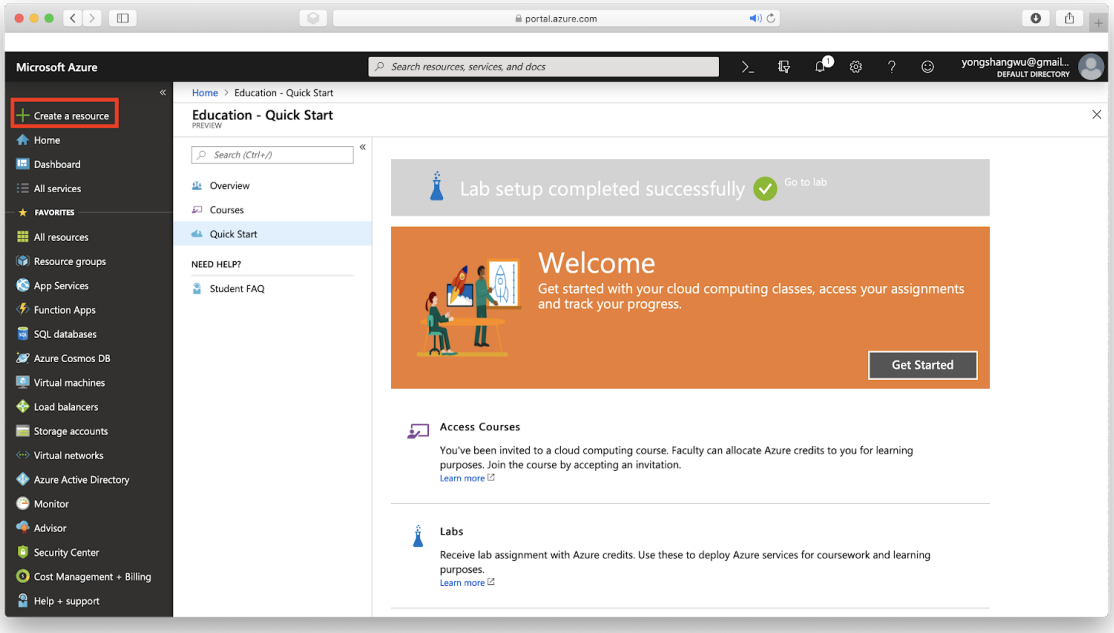


Then you can see the total budget you have and the credit you have consumed:

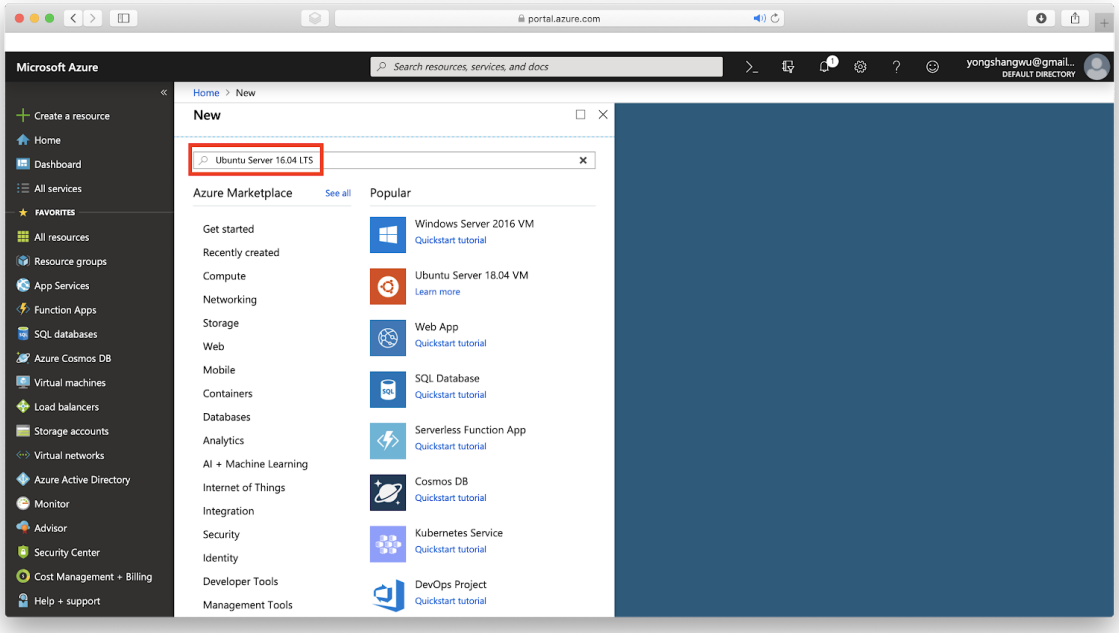


2. Create a VM

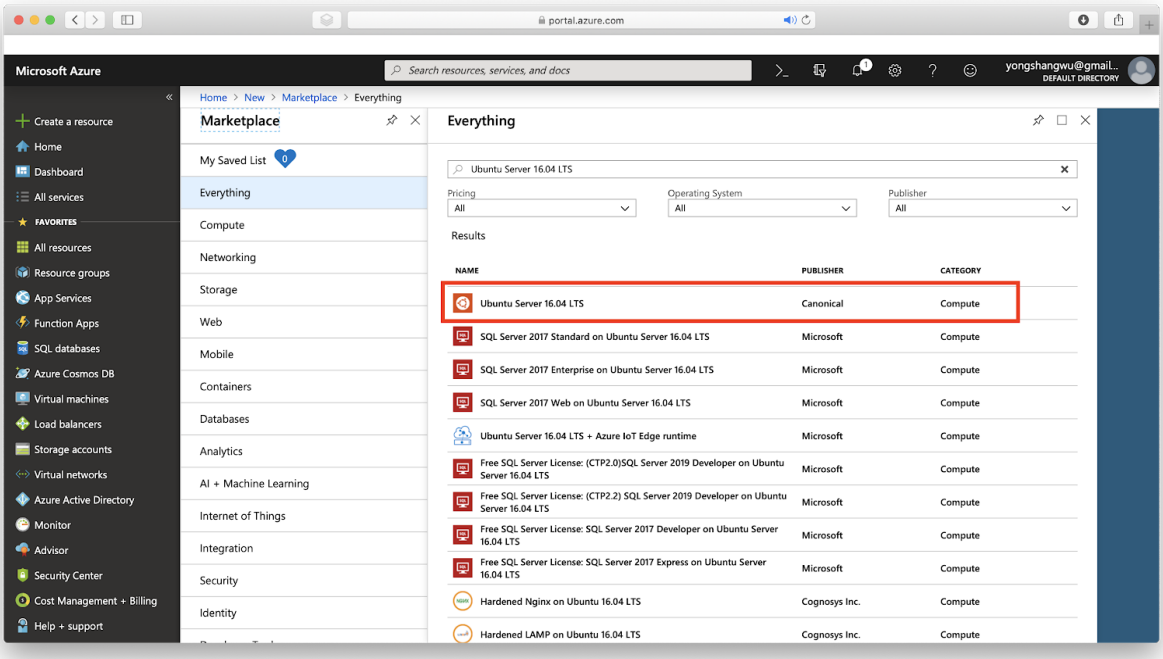
(1) After logging in, you should reach the Azure Portal page. Click the “Create a resource” button at the top of side menu.



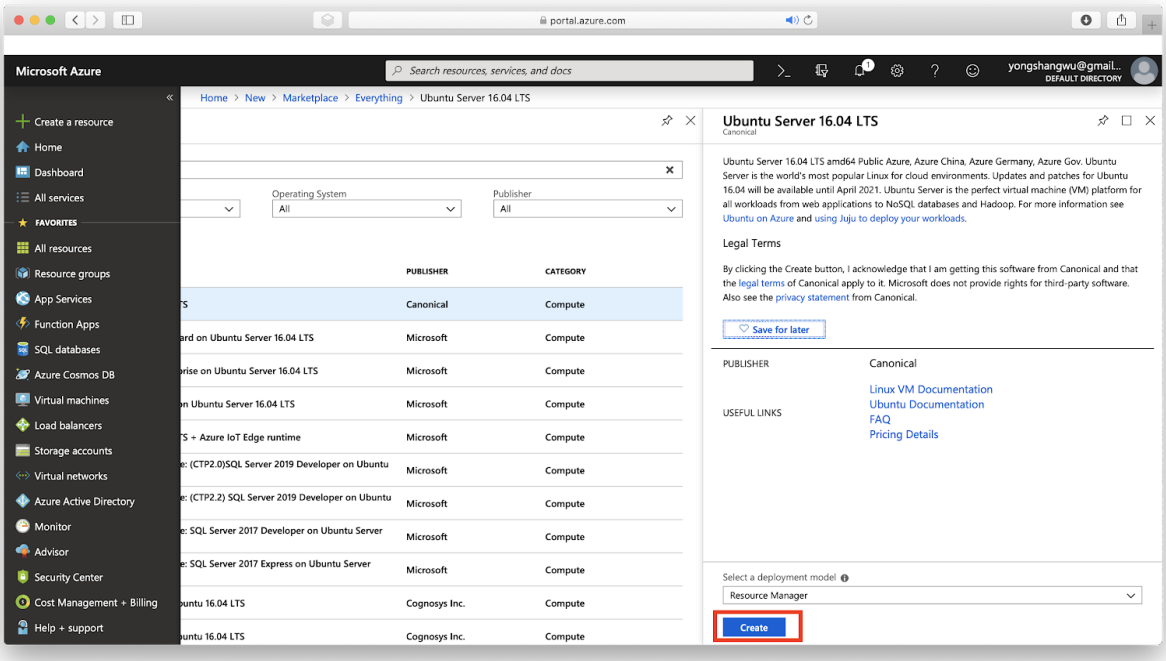
(2) Input “**Ubuntu Server 16.04 LTS**” in the search bar:



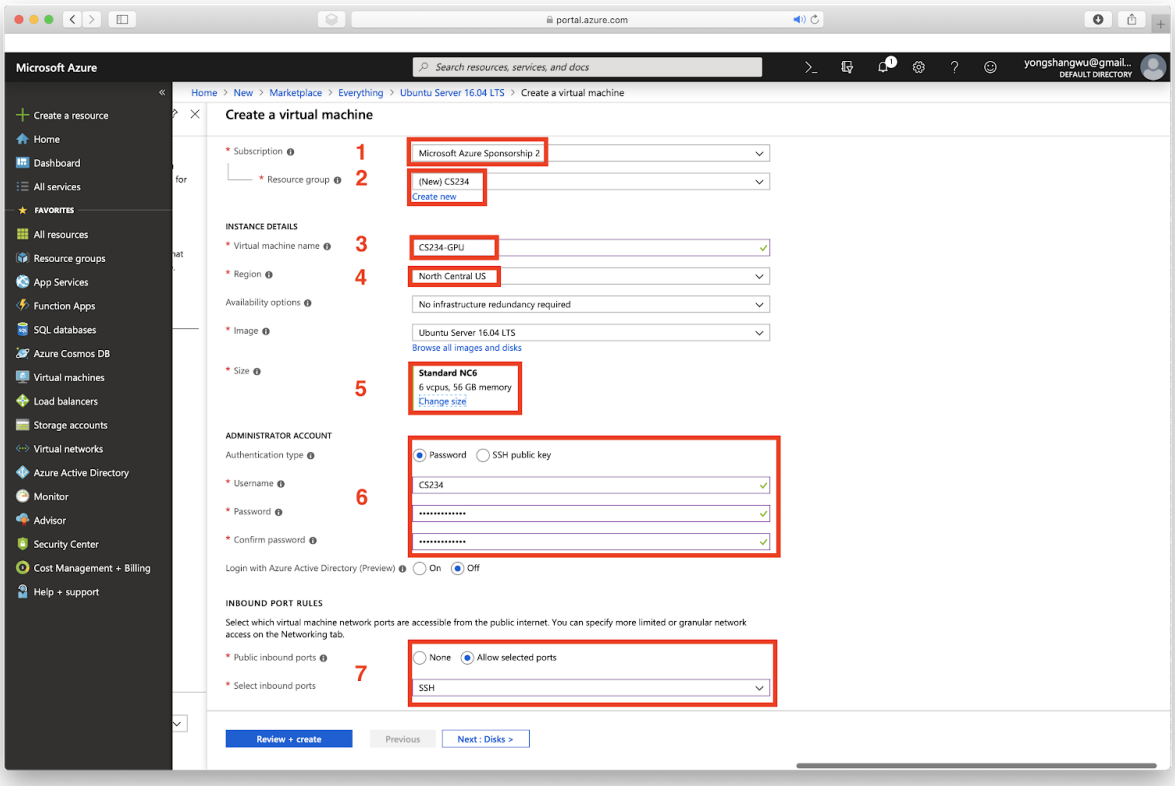
Select “**Ubuntu Server 16.04 LTS**” in the result page:



Then click “**Create**”:



(3) In the VM creation page, fill out each field as instructed below:



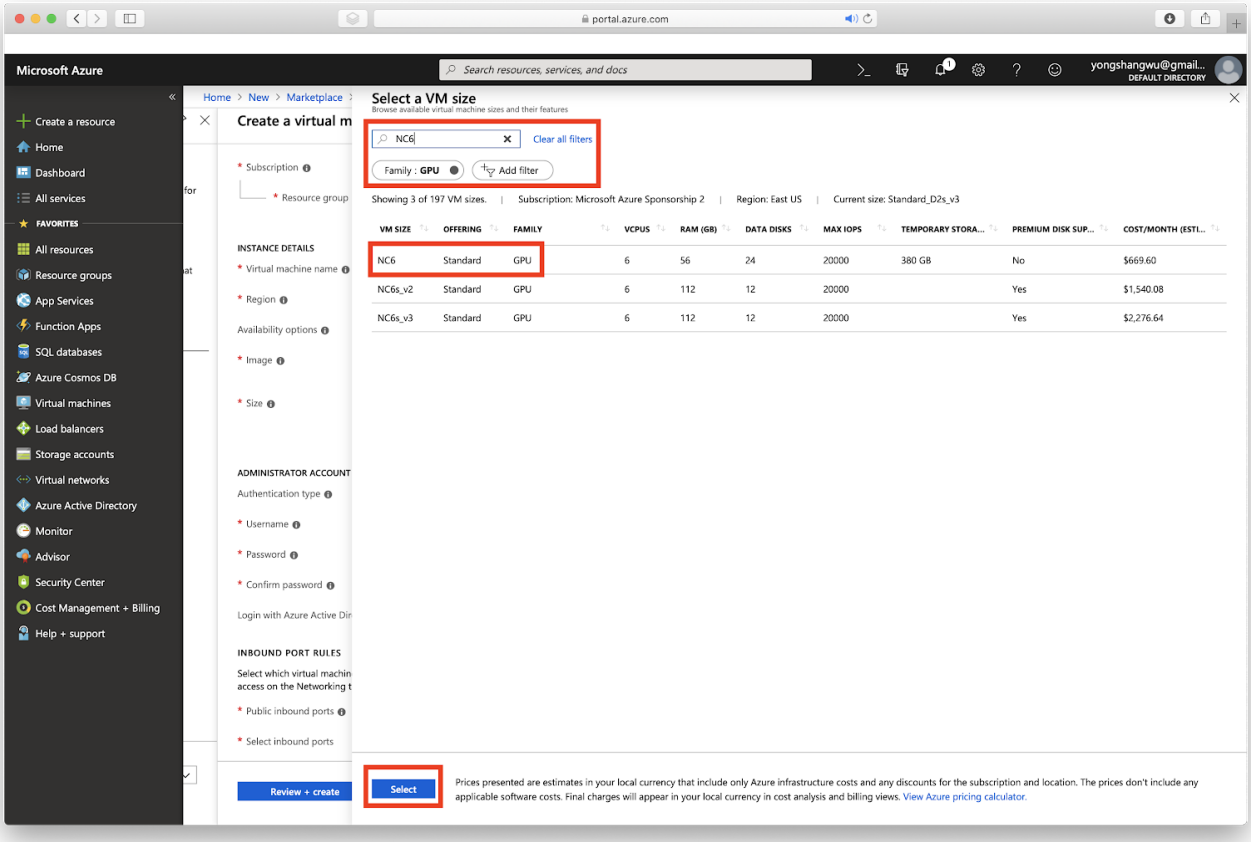
1. Subscription: select “Microsoft Azure Sponsorship 2”.

2. Resource group: click “Create new” to create a group with any name. Instances within the same resource group share a common set of rules (e.g. network, security, etc.).

3. Virtual machine name: any name you want for this particular VM.

4. **Region**: “North Central US” will be fine. You could choose other regions but **note that not all regions have VMs with GPU.**

5. **Size**: click “Change size”, in the selection page, click “**Clear all filters**” then input “**NC6**” in the search bar, click NC6 then “Select”:



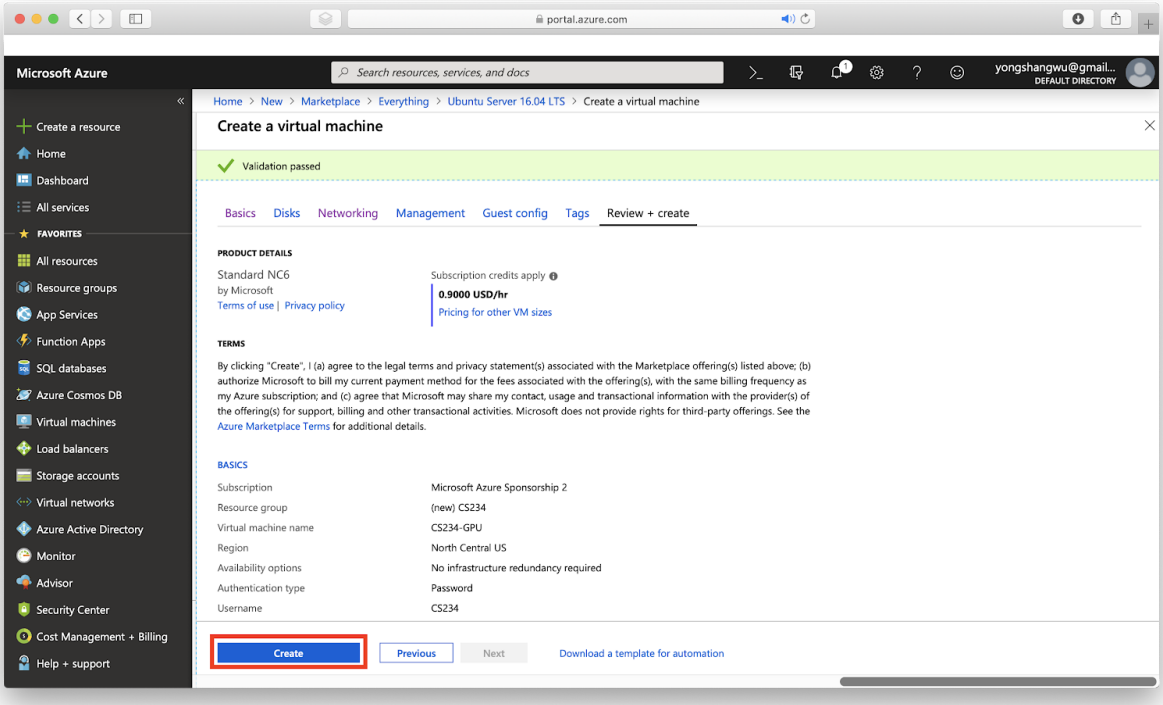
   NC6 instances are installed with an NVIDIA K80 GPU, which should be sufficient for assignment 2.

6. Administrator Account: use either Password (pick any username and password you want) or SSH public key to authenticate connection to the VM.

7. **Public inbound ports**: Select “**Allow selected ports**” and select “**SSH**” port. If you choose None, you may have trouble connecting to the VM later (in which case, see part 3.(2) of this tutorial for solution).

8. (optional) The VM comes with a 56 GB boot disk, which should be enough for assignment. If you need more disk space, click “Next: Disk” then follow the instructions. You can also add and attach new disks later if run out of space.

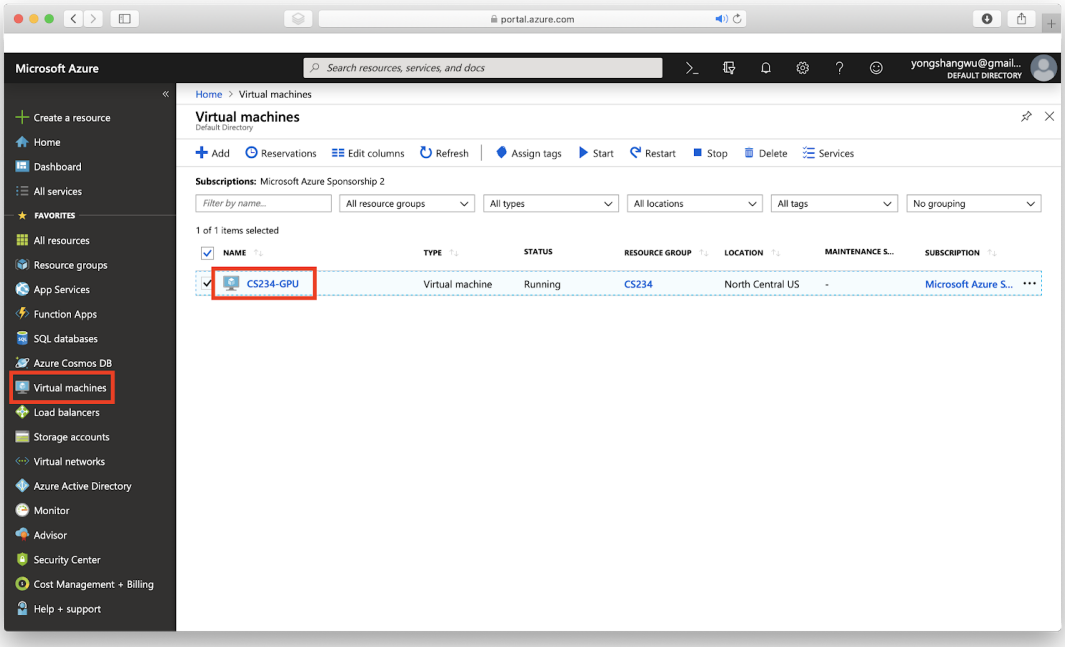
After completing all the fields, click “Review + create” then “Create”:



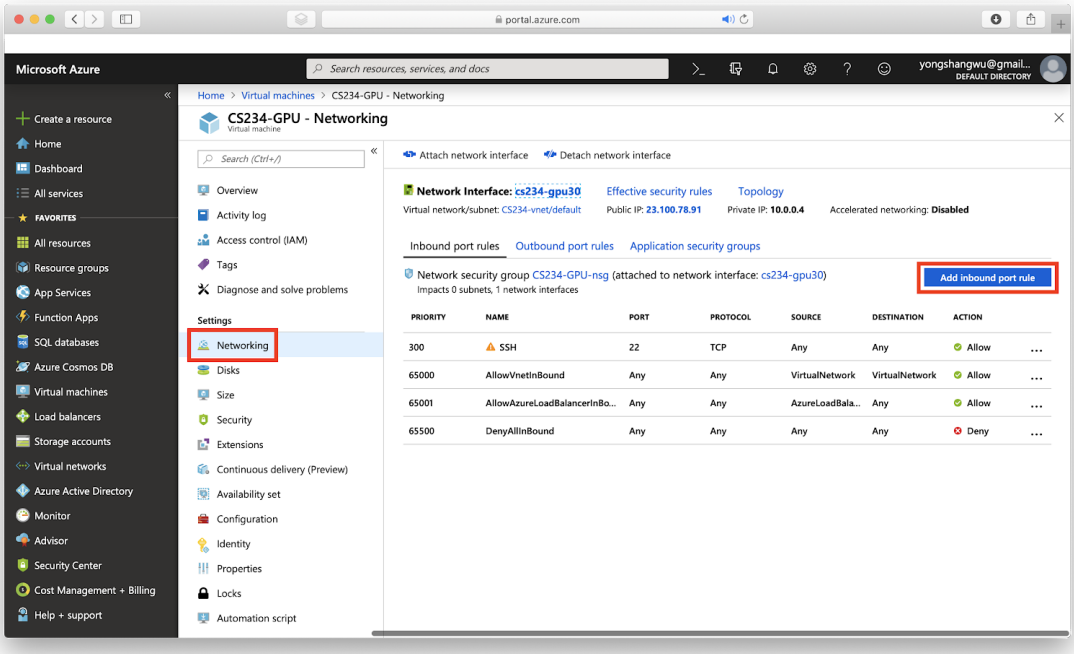
The creation process usually takes 1~5 mins. Once done, we can further configure the VM and start using it.

3. Configure and Use the VM

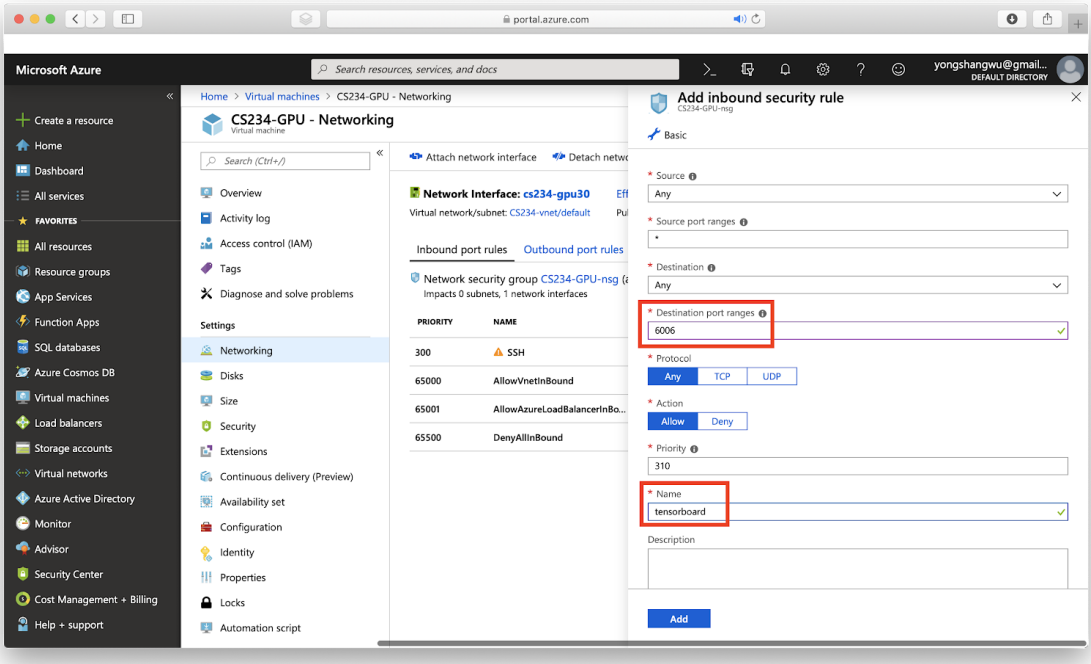
(1) Click “Virtual machines” in the side menu to see your VMs. Click the VM you just created:



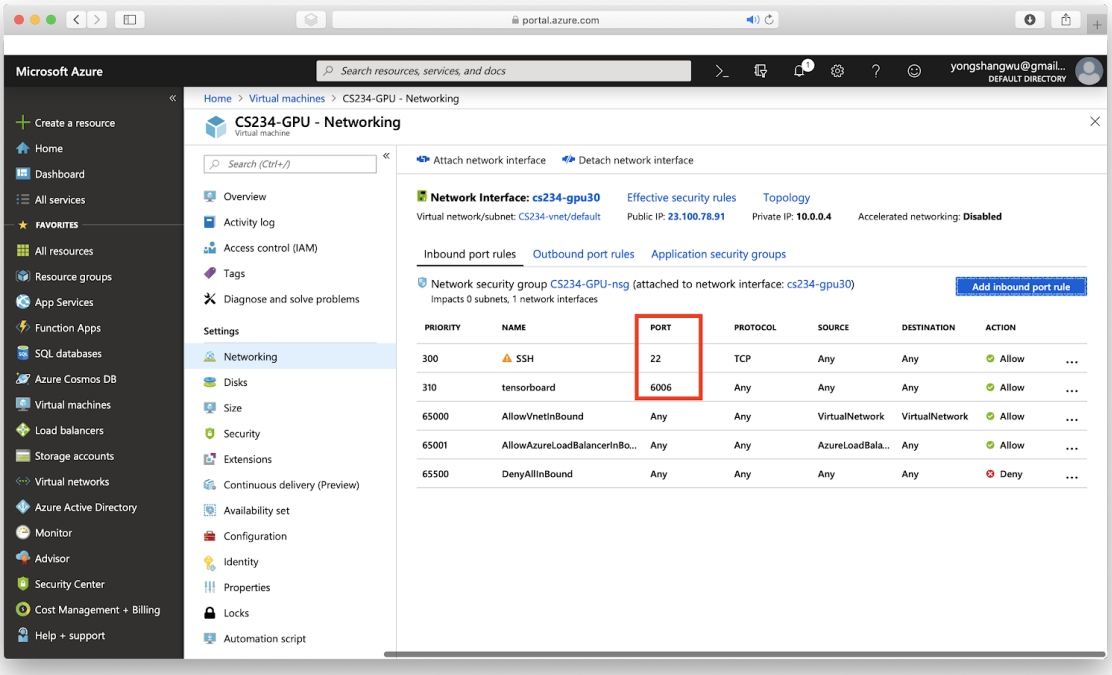
(2) Click “Networking” under “Settings”, then “Add inbound port rules”



Add an inbound port rule with “Destination port range” set to **6006** (the default tensorboard port, you can change this to other port but remember to specify “--port” option when running the tensorboard command) and “Name” to “tensorboard”(or anything you want):

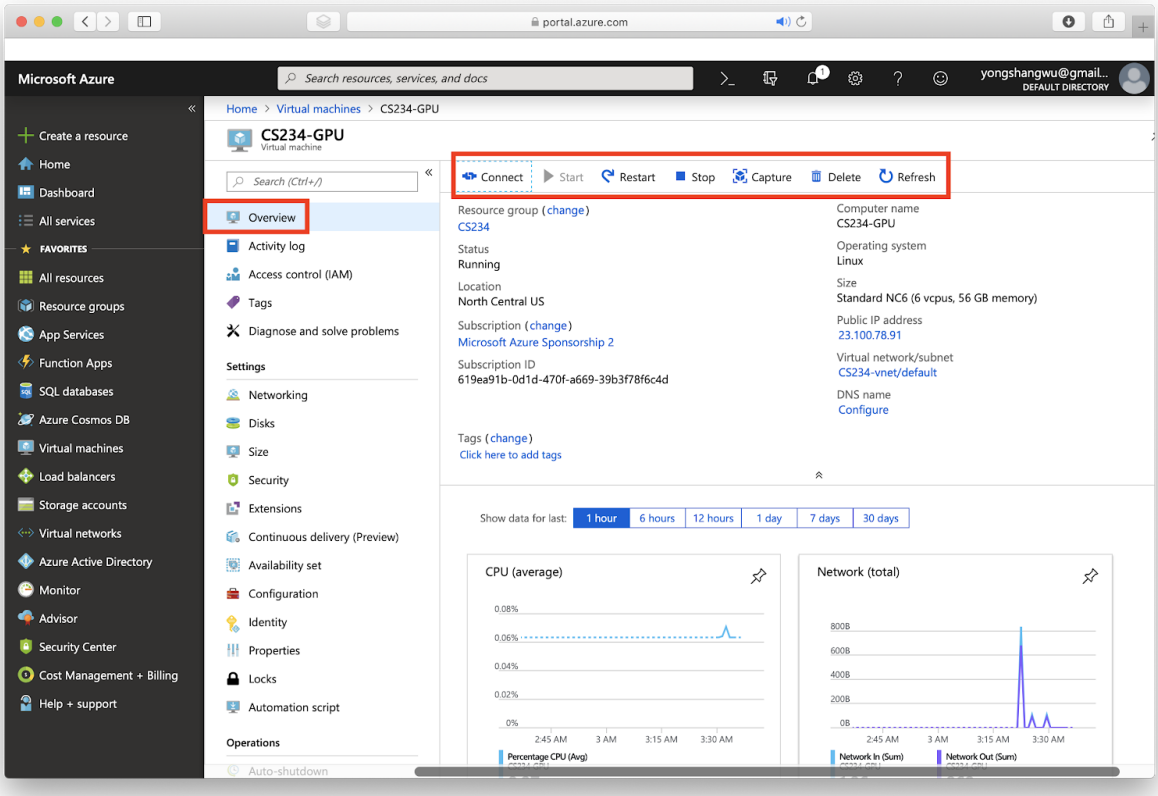


Make sure you have the following 2 networking rules (port 22 for SSH and 6006 for tensorboard):

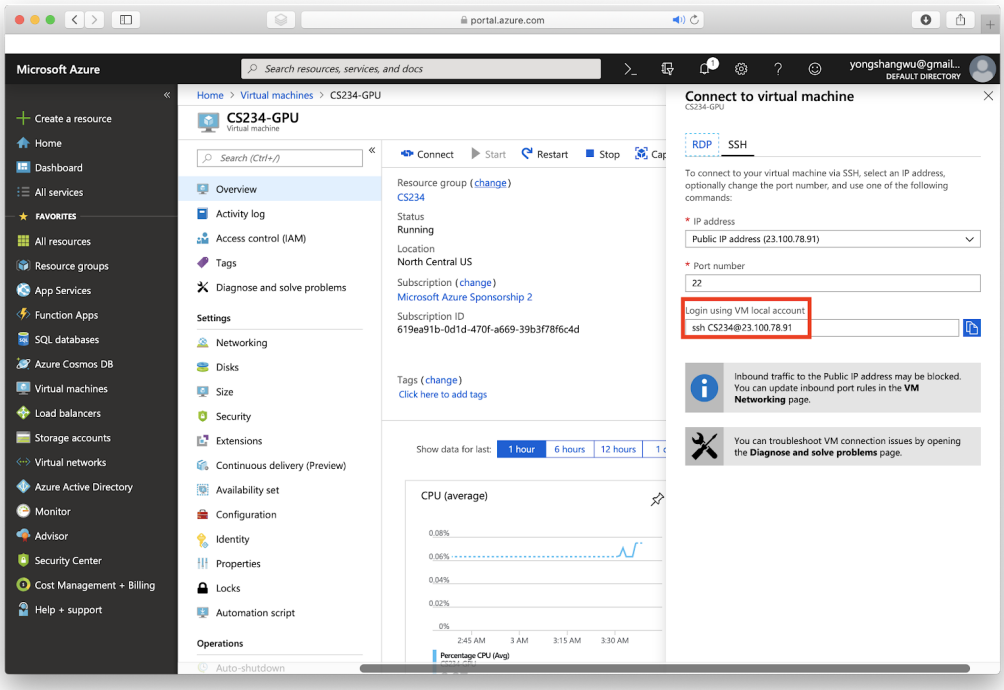


As mentioned in part 2 of this tutorial, if you forgot to allow SSH port when creating the VM, you can add an inbound rule for port 22 here, following similar procedures for the tensorboard port.

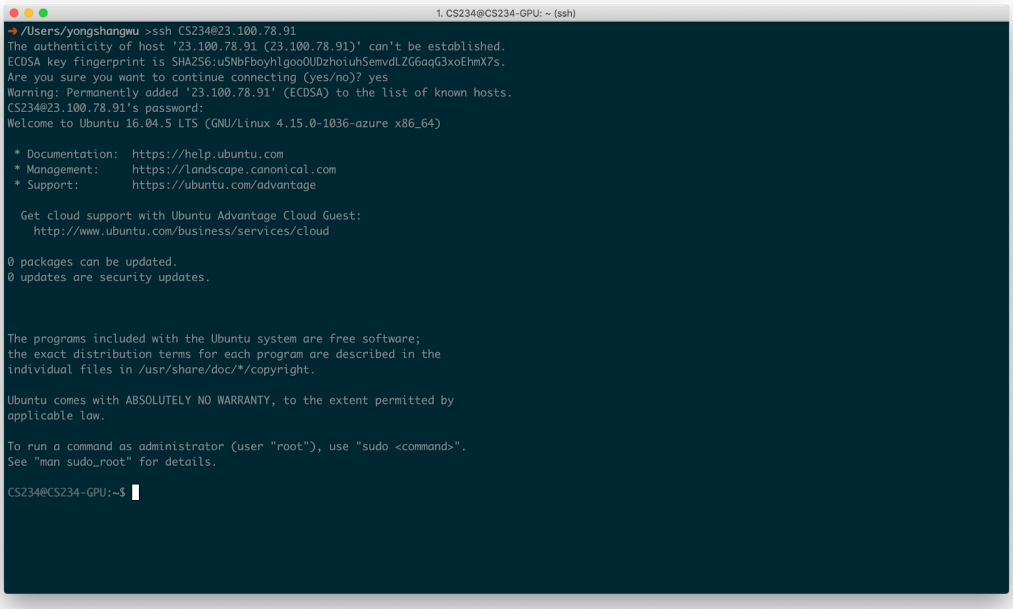
(3) Finally, we can start using the VM! In the overview tab of the instance page, you can start/**stop**/delete the VM by clicking corresponding buttons. **Again, always remember to stop the VM instance if you are not using them.** Azure also offers an auto-shutdown feature which shuts down the VM at scheduled time. Try it out!



When the VM is running, you can click “Connect” to find the SSH command to connect:



Run that command in your terminal (or [PuTTY](#) if you are using Windows), you can log into the VM we just created and configured:



4. Install CUDA, Tensorflow & Other dependencies

(1) After logging into the VM, clone the setup repo we created for you:

```
git clone https://github.com/AndyYSWoo/Azure-GPU-Setup.git
```

(2) cd into the cloned repo:

```
cd Azure-GPU-Setup
```

(3) Add execution permission for the setup scripts:

```
chmod +x *.sh
```

(4) Run the first setup script:

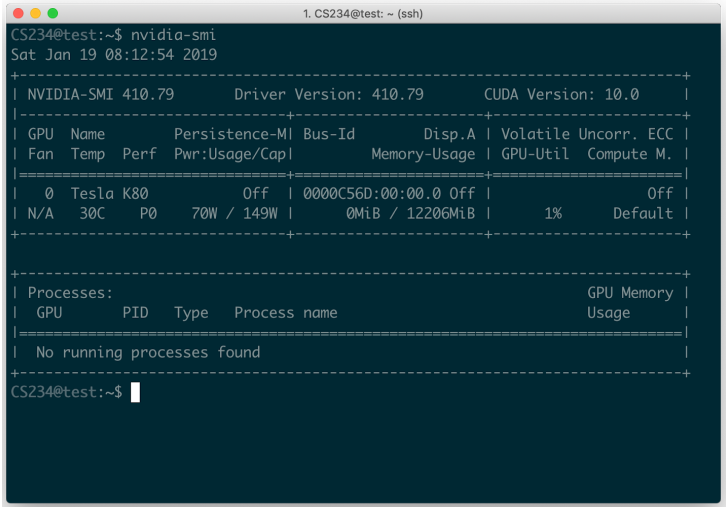
```
./gpu-setup-part1.sh
```

This script takes about **20~30 minutes** to run. Go take a well-deserved break :)

Upon completion, the VM will restart automatically. Refer to part 3.(3) of this tutorial for the command to **SSH into the VM again**. The command might **differ from last time** because the public IP address may (or may not) change after reboot.

(5) After re-logging into the VM, you can use the following command to show GPU information

```
nvidia-smi
```



(6)cd into the cloned repo again

```
cd Azure-GPU-Setup
```

(7) Run the second setup script:

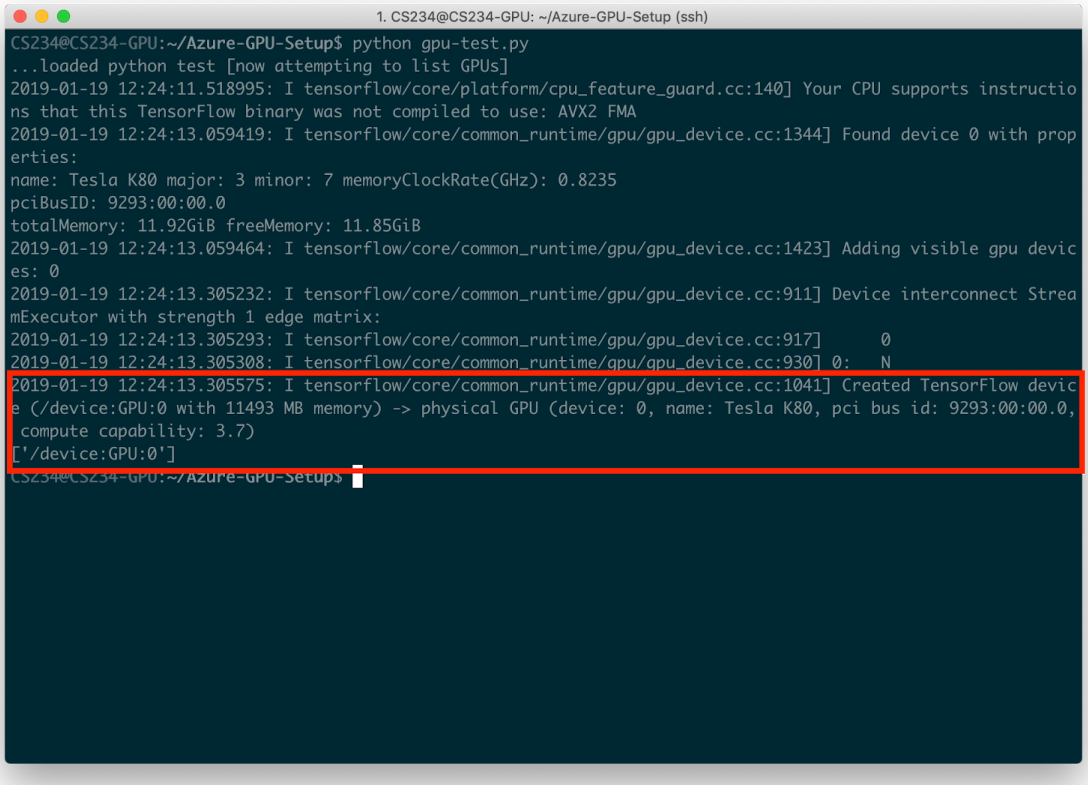
```
./gpu-setup-part2.sh
```

This script takes about **7~10 minutes** to run. Upon completion, you should see “Test passed!”, which means the cuDNN library has been installed successfully.

(8) Run the following command:

```
python gpu-test.py
```

You should see that tensorflow finds the GPU on your VM correctly:



Congratulations!!! You have successfully set up the VM with a GPU.

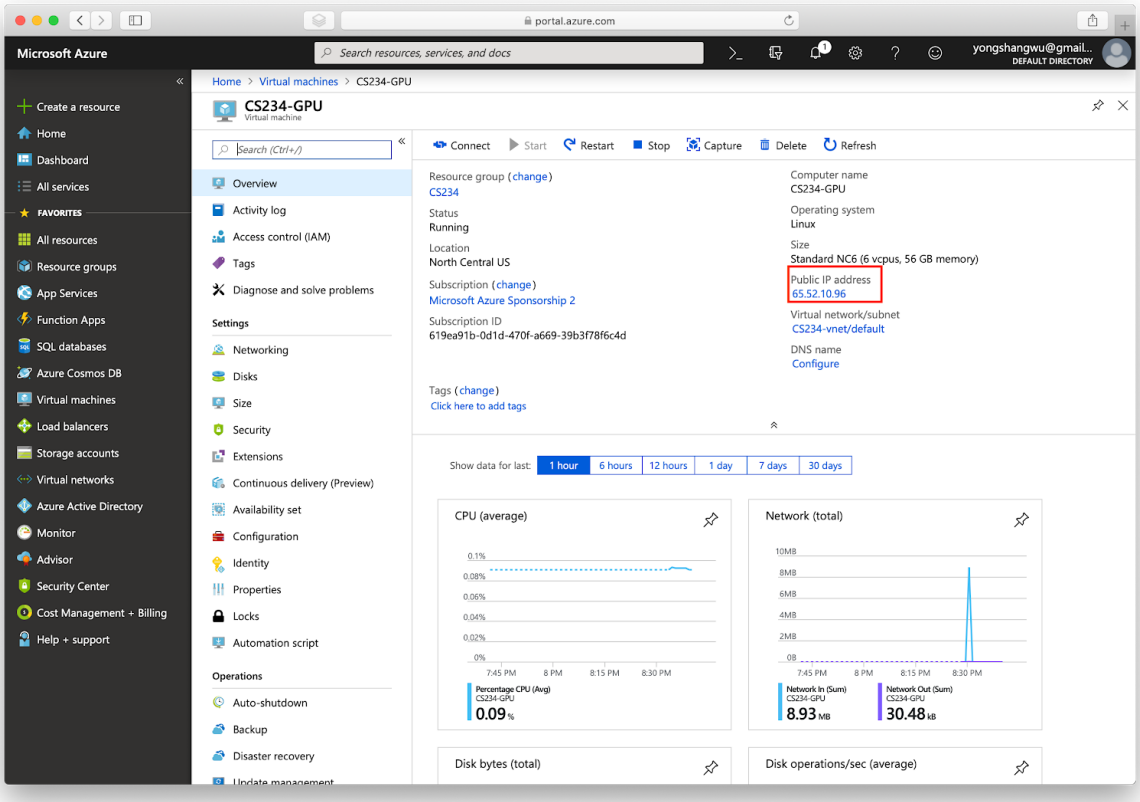
Now burn that GPU up with your amazing models!

6. Transferring Files between VM and Local

To transfer files between the VM and your local machine, you may want to use [the SCP command]. For example, the following command:

```
scp -r /local/path/to/starter_code user_name@public_ip:~/
```

copies your modified starter code folder to the home directory of your VM, `/local/path/to/starter_code` is the directory of your starter code in your local machine, `user_name` is the username you set when creating the VM in Part 2.(3) Step (6) of this tutorial, `public_ip` is the public IP address of your VM, where you can find at the instance overview page:



You can replace `~/` with any path on your VM to specify where you want to put the code.

7. Tensorboard

In assignment 2, you may want to use tensorboard to monitor the training of your models. After logging into the VM, cd in to experiment folder:

```
cd ~/starter_code/results/q5_train_atari_nature
```

Then turn on tensorboard using:

```
tensorboard --logdir=. --port=6006
```

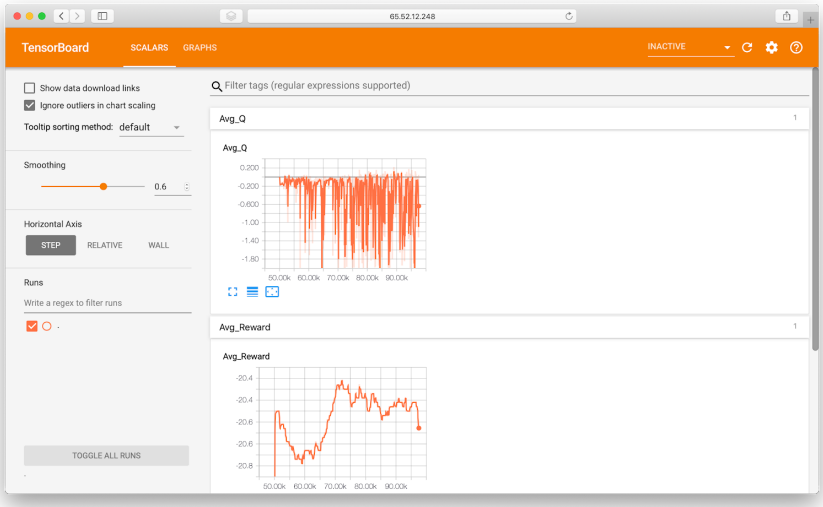
The `--port` option is optional, if you do not specify it, the default port will be 6006.
Next, open your web browser and visit

http://public_ip:port

to view the tensorboard, where `public_ip` is the public IP address of your VM and `port` is the tensorboard port you specify when running tensorboard command (default 6006). For example, if your VM's public IP address is 65.52.12.248 and the default tensorboard port is used, visiting

http://65.52.12.248:6006

gives you the tensorboard page:



Note: If you choose a different port for tensorboard, say **5678**, remember to add a corresponding inbound port rule for your VM, by following steps in Part 3.(2). of this tutorial.

8. [Optional] Working under Virtual Environment

Sometimes you may want to work in an isolated environment. For example, you may want a separate environment for assignment 2 different from your project. virtualenv serves that need conveniently. To create an isolated environment for assignment 2, cd into the start code folder, then run the following commands:

```
virtualenv .env -p python3.6
source .env/bin/activate
pip install -r requirements.txt
```

Once finished working on it, run `deactivate` to exit the virtualenv.

9. FAQ

- Do not install updates using: sudo apt-get install --upgrade
  This might break the CUDA driver installation if the kernel is updated. If you did so and the code fails to run with resulting dependencies, starting all over again with a new VM would be the fastest way out.

- What if I failed when running `gpu-setup-part1.sh` or `gpu-setup-part2.sh`
  Rerun the failed script again, and see if you can continue to next steps. If failed again, make a piazza post describing your problem with the error information output. Start your post title with [Azure].

- What if I run out of credits
  $70 should be more than enough to run the whole experiment for 4~5 times. Frequently monitor the tensorboard to detect misbehavior of your model early. Always remember to turn off your VM if not using them. If you still run out of credits, or need more to work on the extra credit, write an email to wuy@stanford.edu with [CS234 Azure] in the title. In the email, state how much you have spent, what you did with these credits and how much more you need.

- Problems connecting (SSH) to the VM
  0. Try `ping public_ip`
  1. Make sure port 22 is allowed in the networking inbound rules, see Part 3.(2).
  2. Restart the VM and make sure you are connecting to the correct public IP address.

Again, make sure your VM instances are STOPPED while not used.
