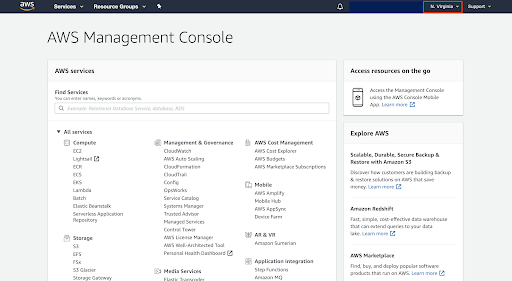
# **Section 3 AWS Setup Instructions:**

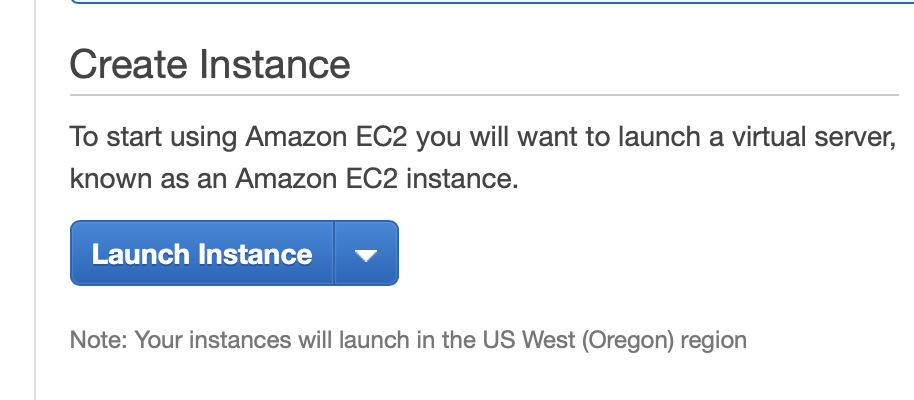
* Create an AWS account [here](https://portal.aws.amazon.com/billing/signup?nc2=h_ct&src=default&redirect_url=https%3A%2F%2Faws.amazon.com%2Fregistration-confirmation#/start) if you have not yet done so.
* Request AWS GPU credits in Ed if you have not yet done so.
  + Redeem Credits and check remaining credits by going to your Account Dropdown Menu -> My Billing Dashboard -> Credits
* Request a limit increase [here](https://console.aws.amazon.com/support/cases#/create?issueType=service-limit-increase&limitType=ec2-instances) if you have not yet done so (this may days to be approved).
  + Limit Type: “EC2 Instances”
  + Region: “US West (Oregon)”
  + Primary Instance Type: “All P Instances”
  + **New limit value: 4 (EDIT: This value used to be 1 but apparently this number now represents vCPUs rather than instances, and p2.xlarge needs 4 of them)**
  + Case description: “Training neural networks for the Stanford deep learning class (CS230 - cs230.stanford.edu).”
* Follow the tutorial for cloning a repo and running a model locally.
* Once you receive credits + limit increase:
  + 1. Set up an AWS instance.
  + 2. Follow the tutorial for cloning a repo and running a model on your AWS instance.

# **Setting Up an AWS Instance**

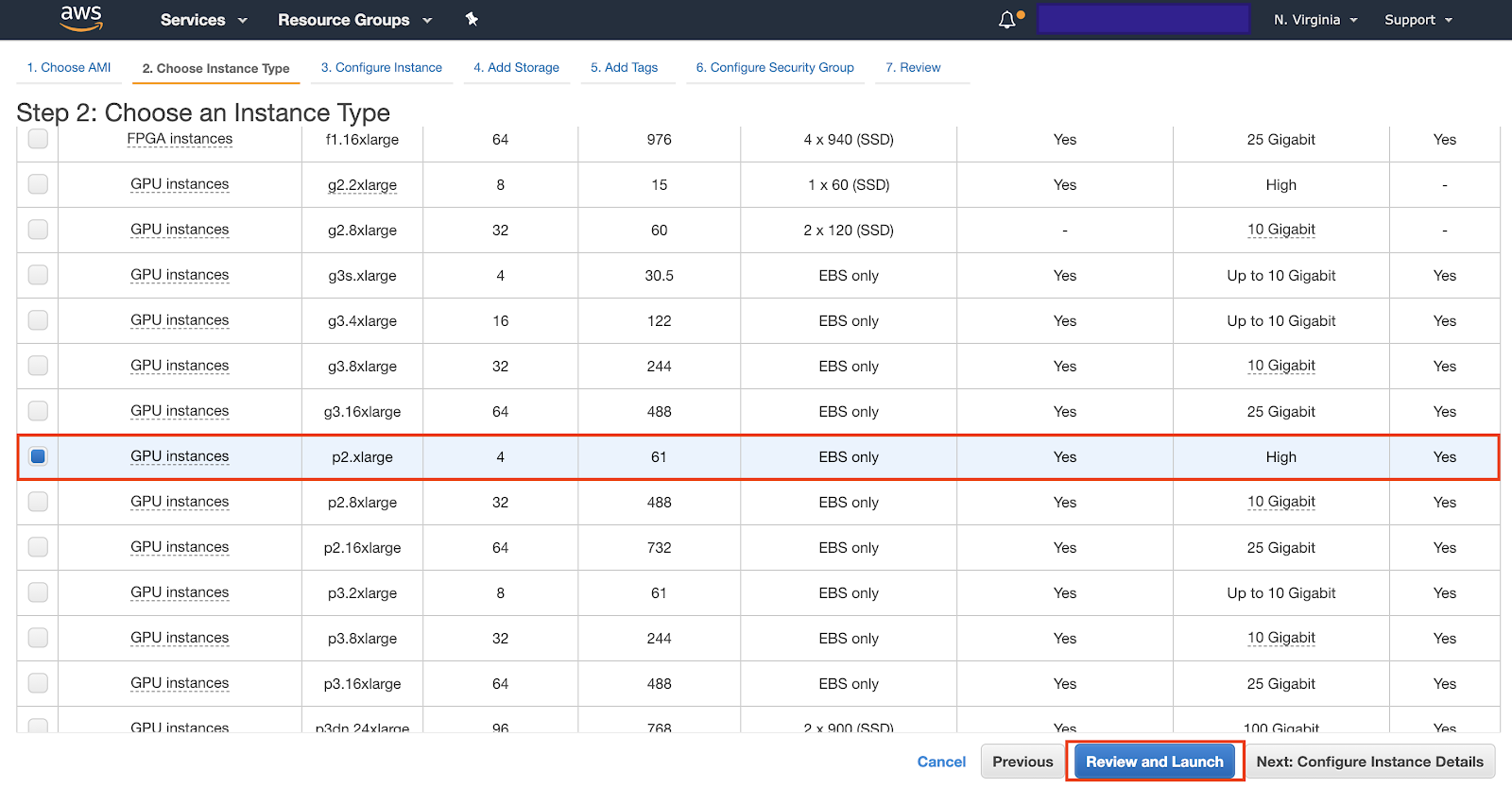
* Navigate to the [AWS console](https://us-west-2.console.aws.amazon.com/console/home?region=us-west-2)
* On the top right of the page, click the location name and set it to US West (Oregon)
* Under All Services->Compute, click EC2

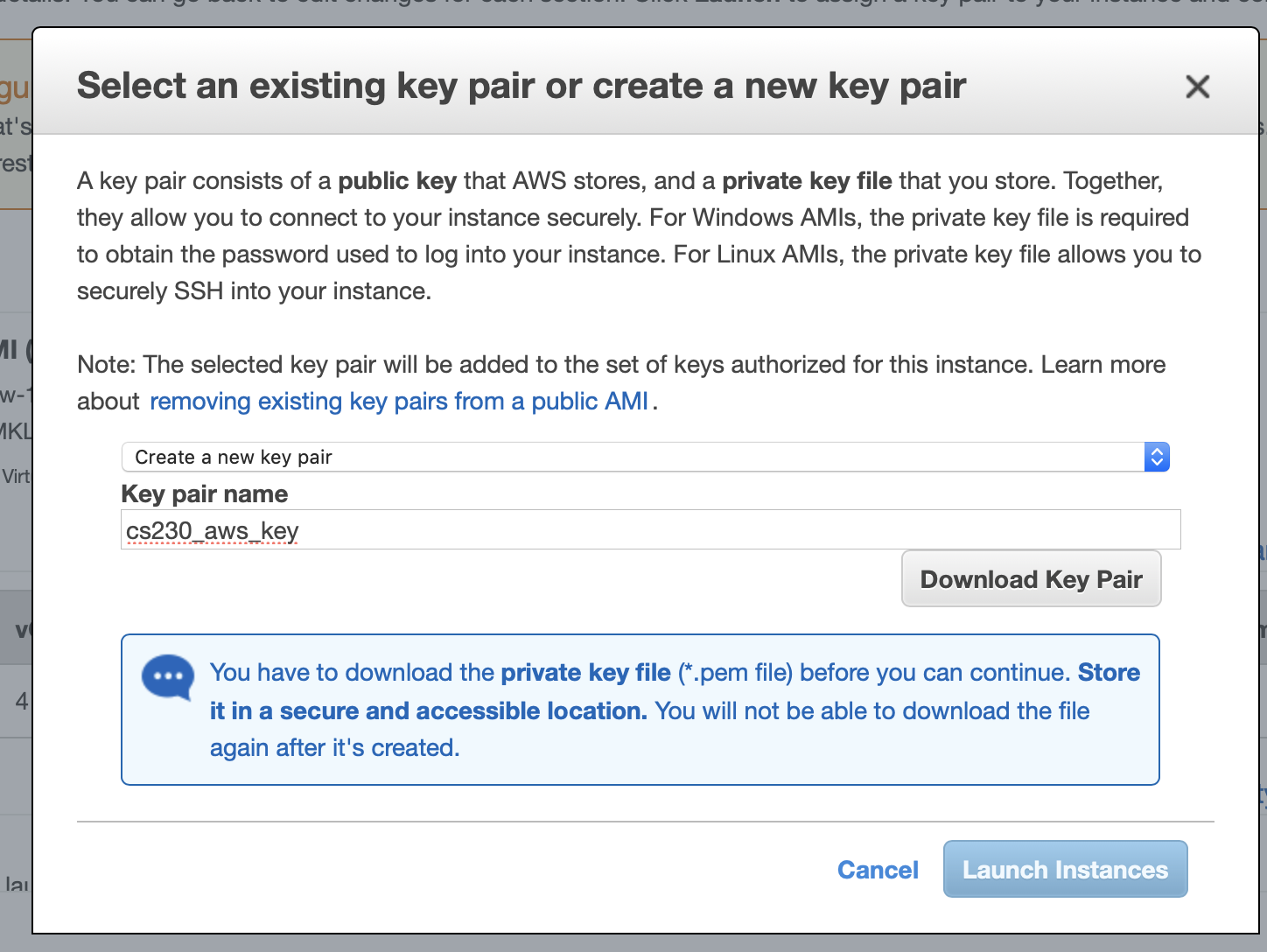


* On that page, click the Launch Instance button



* Search for and select **Deep Learning AMI (Ubuntu 18.04) Version 59.0**
* On the next page, select p2.xlarge instance, then click “Review and Launch”



* Click the blue “Launch” button
* A pop-up window will appear asking for a key pair. Select “Create a new key pair” name your key pair, and click “Download Key Pair”.****
* Take note of the path where this key pair is downloaded
  + e.g. “/Users/TA/Downloads/cs230\_aws\_key.pem”
* Click the blue “Launch Instances” button.
* Now, open up terminal so that we can move this key to a directory for private keys.
* Execute the commands:
  + mkdir ~/.ssh
  + mv <.pem file path from above> ~/.ssh/
* Now change the permissions on the key file, which now has the path ~/.ssh/<keyfilename.pem>
  + chmod 400 ~/.ssh/<keyfilename.pem>
    - e.g. chmod 400 ~/.ssh/cs230\_aws\_key.pem
  + Alternative on windows:
    - <https://superuser.com/questions/1296024/windows-ssh-permissions-for-private-key-are-too-open>

# 

# 

# **SSHing into the instance**

* **IMPORTANT! To avoid using up all of your credits accidentally, make sure to STOP your instance when you are no longer using it. Do NOT click Terminate. Click STOP. We cannot stress the importance of this enough!**
* Click on “View Instances” to check that it is “Running” and passed “2/2 status checks”. It will take some time to pass the checks but after that, you will be ready to ssh into the instance. Take note of the Public IP of the instance launched.
* SSH into your instance with the command:
  + ssh -i <keyfilename.pem> ubuntu@<Public\_IP>
  + e.g. ssh -i ~/.ssh/cs230.pem [ubuntu@ec2-54-188-166-21.us-west-2.compute.amazonaws.com](mailto:ubuntu@ec2-54-188-166-21.us-west-2.compute.amazonaws.com)

*If you don’t have conda pre-installed:*

* wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\_64.sh
* bash Miniconda3-latest-Linux-x86\_64.sh
* rm Miniconda3-latest-Linux-x86\_64.sh
* source .bashrc
* conda create -n tensorflow\_p36 python=3.5 pip
* Set up a virtual environment using conda via the following command
  + source activate tensorflow\_p36
* To deactivate, you run:
  + source deactivate

# **Running the model on the AWS instance**

* SSH into your instance and activate a virtual environment (refer to above section on “SSHing into the Instance” for a refresher)
* Clone the repo on your instance and navigate to the directory via
  + git clone <https://github.com/anishathalye/neural-style.git>
  + cd neural-style
* Download the pretrained weights for the network on your instance via:
  + curl -O [https://www.vlfeat.org/matconvnet/models/imagenet-vgg-verydeep-19.mat](http://www.vlfeat.org/matconvnet/models/imagenet-vgg-verydeep-19.mat)
* This time, run the model for 1000 iterations via:
  + python neural\_style.py --content examples/1-content.jpg --styles examples/1-style.jpg --output output.jpg
* If you get an error output "scipy.misc" has no attribute 'imread" do:
  + pip install scipy==1.1
* Exit SSH & transfer the output image from your instance to your local computer via
  + scp -i ~/.ssh/<key file.pem> ubuntu@<PUBLIC\_IP>:**~/neural-style/output.jpg** <local\_path\_to\_save>
  + scp -i ~/.ssh/<key file.pem> <local\_path\_to\_copy> ubuntu@<PUBLIC\_IP>:**~/location/to/copy/to/**
  + e.g. scp -i ~/.ssh/cs230\_aws.pem ubuntu@ec2-54-188-166-21.us-west-2.compute.amazonaws.com:**~/neural-style/output.jpg** Desktop/
  + gcloud / gdrive
* Open and view the image on your local computer by navigating to the location you saved the image at.
* **REMEMBER TO STOP YOUR INSTANCE WHEN YOU ARE DONE WITH THE TUTORIAL! :)**

# **[Optional] Running Jupyter Notebook on the AWS instance**

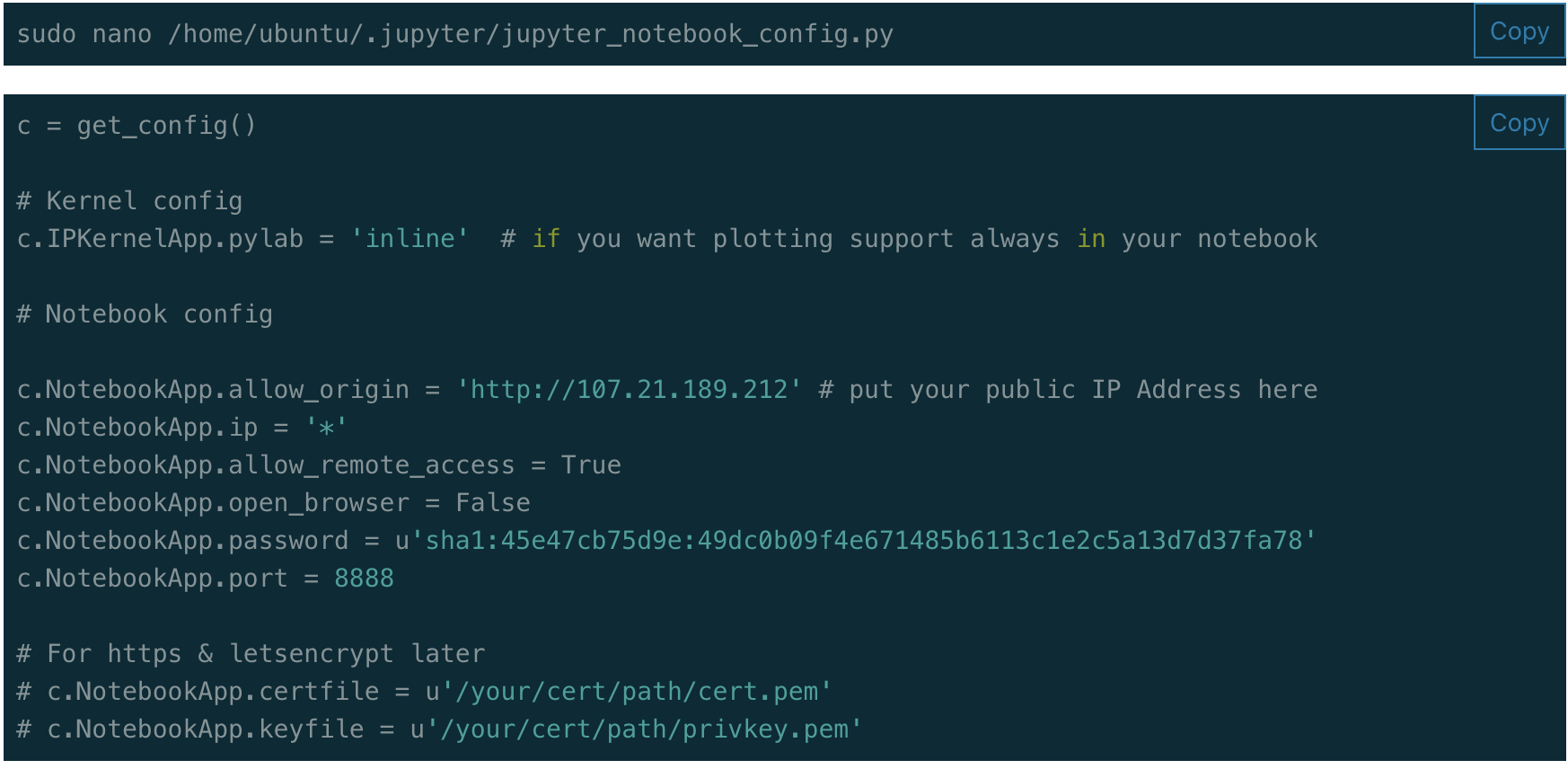
Sometimes we may find it easier to run our code in an interactive environment like Jupyter Notebook, especially for plotting and visualizing the results. Running Jupyter on AWS instance also saves you from transferring images back and forth between the remote and the local. You can directly visualize them and make code edits if needed.

Reference:

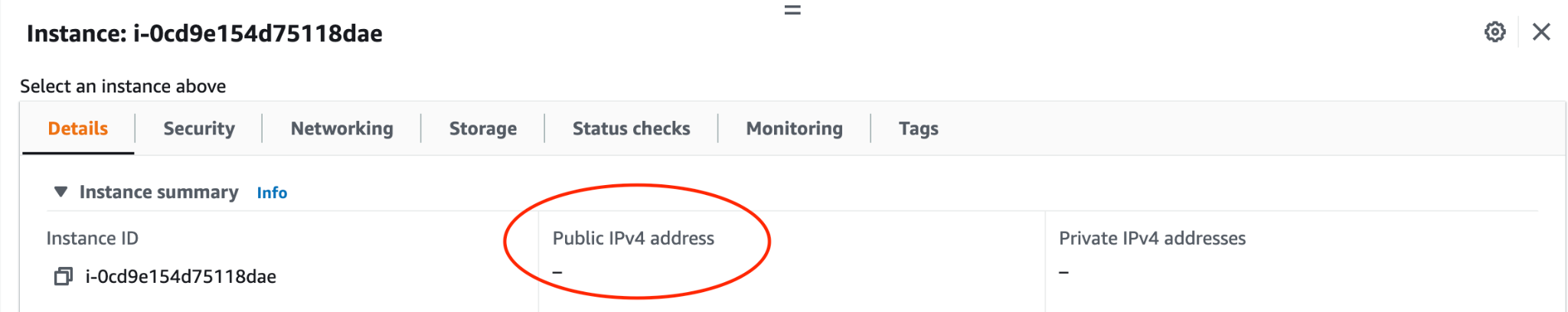
1. <https://www.codingforentrepreneurs.com/blog/jupyter-notebook-server-aws-ec2-aws-vpc>. We can directly jump to section: Provision the Jupyter Server

2. <https://www.youtube.com/watch?v=FEBYi8Ia8bk>. Can directly jump to section 10

1. Run jupyter notebook --generate-config in the instance to create a configuration file that we need to deploy a jupyter notebook server. The file should locate under /home/ubuntu/.jupyter/jupyter\_notebook\_config.py
2. Update the configuration for live running



1. c.NotebookApp.allow\_origin = ‘http://<public IP address of your instance>’. You can find the IP address here:



1. If you want to use a simple password, comment out c.NotebookApp.password = . Instead, run jupyter notebook password to set up your password. Otherwise, you need to following instructions in the first reference to set up a hashing password.
2. Edit Security Group of your instance to expose port 8888.
   1. Go to EC2 -> Security Group -> <The security group you use in your instance>
   2. Click “Edit Inbound Rules”
   3. Click “Add Rule”. Choose type = “Custom TCP”, port = 8888, Source = 0.0.0.0/0
   4. Click “Save Rule”
3. You should now be able to access jupyter notebook in your browser by going to “<public IP address of your instance>:8888”

Please check out the recorded section 3 video for demonstration

Much of this information can be found on the Section 3 page on the course website, please feel free to also consult that post.