**Lab 7 – Jupyter Notebook and S3 Storage on AWS**

In this lab, we will install Jupyter notebook on an AWS instance and access data files stored in AWS S3.

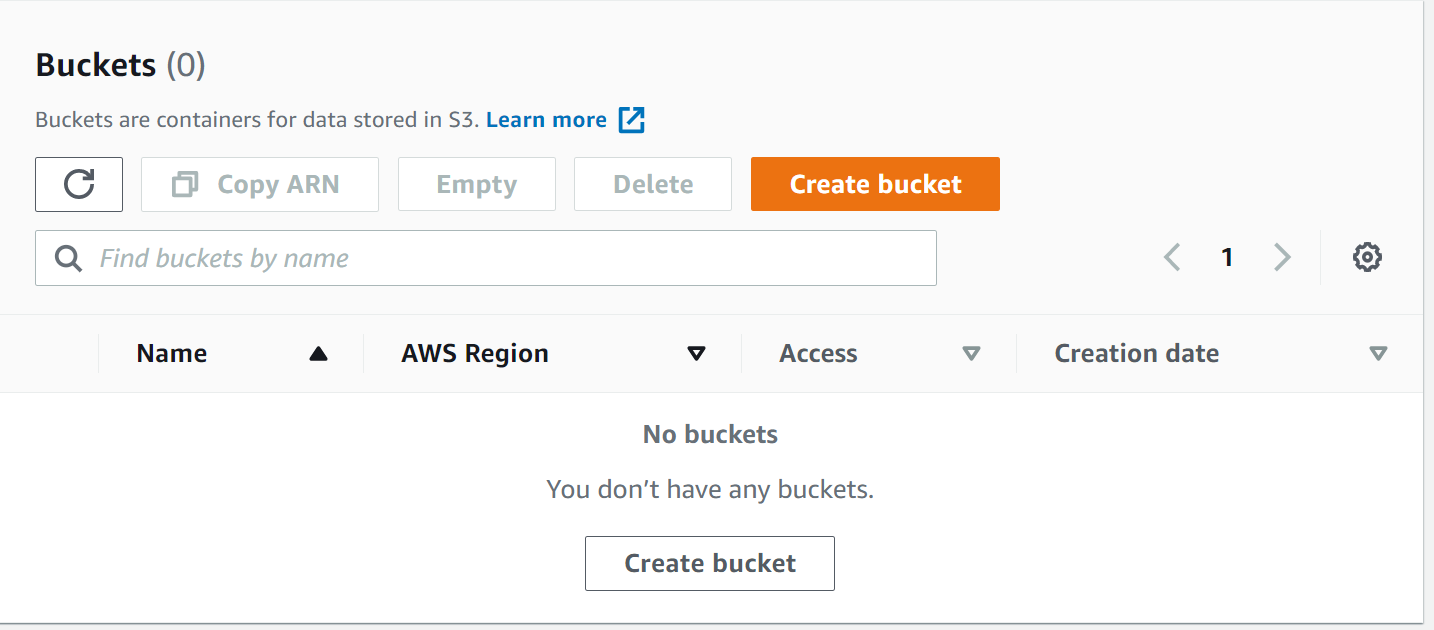
# Part1: AWS account

Apply for AWS free tier account from [AWS Free Tier (amazon.com)](https://aws.amazon.com/free/?nc1=h_ls&all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc).

# Part2: Jupyter Notebook and S3 Storage on AWS

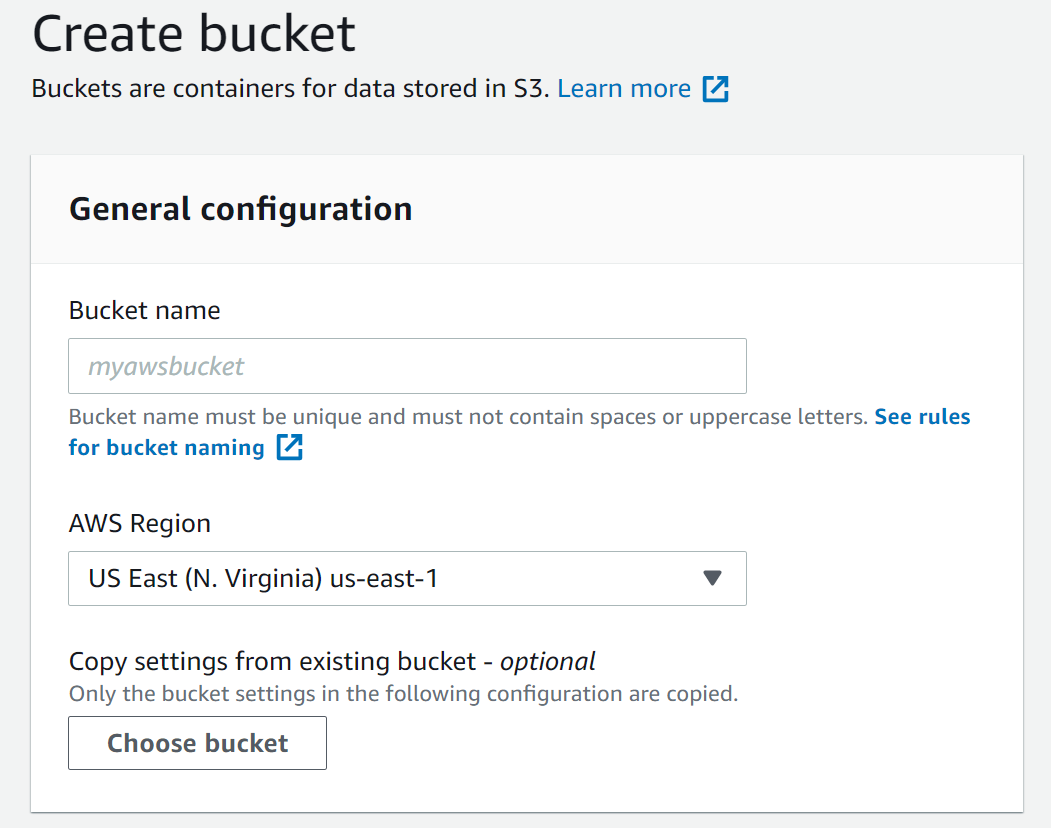
**Step 1: Create a bucket and upload data files.**

Login your AWS Educate Account, go to the AWS console. Under the Services tab, find S3, Click on Create bucket.

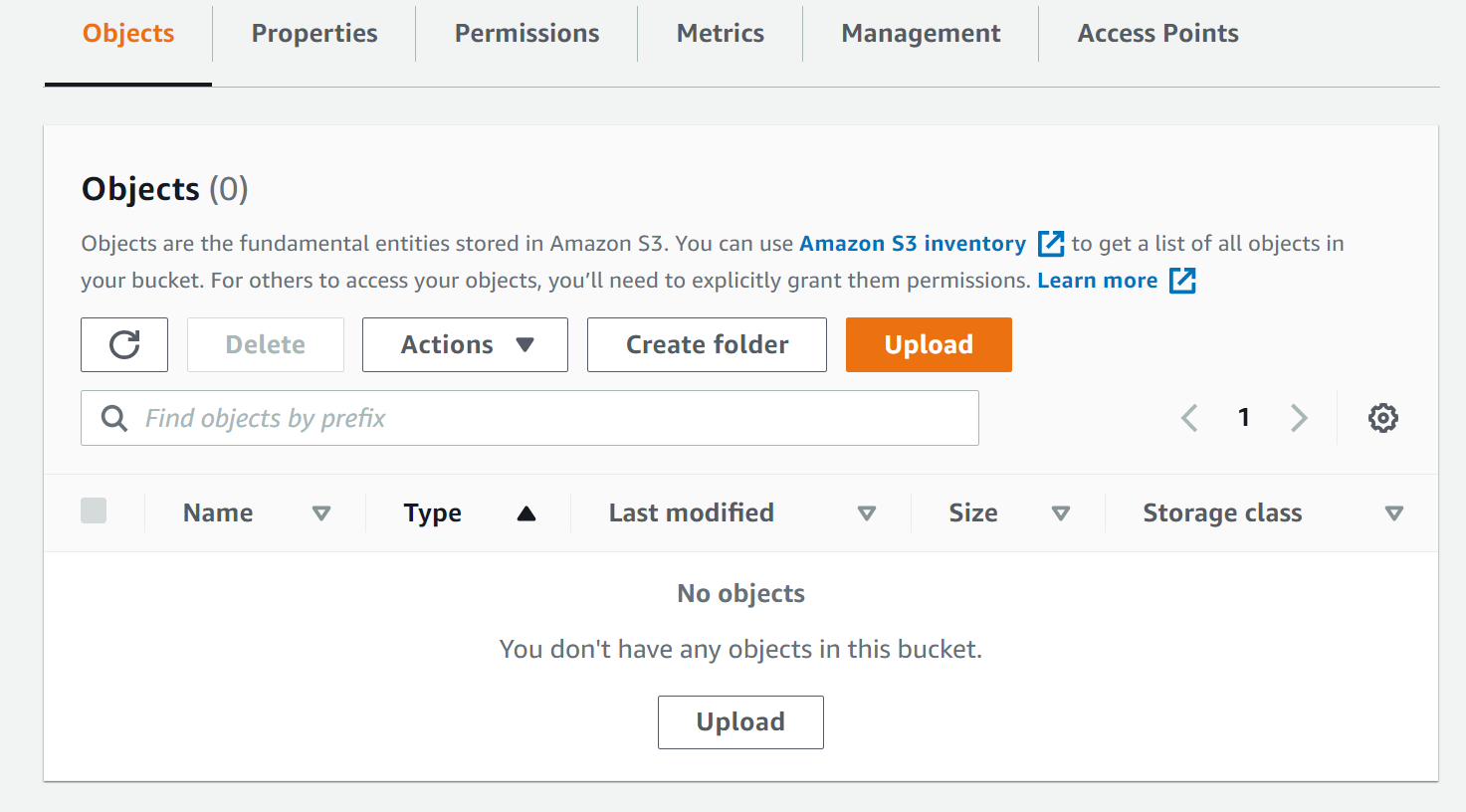


Create a bucket and give it a name. The name of the bucket has to be unique. If you failed to create a bucket, it is because the name you used for bucket was already in use.

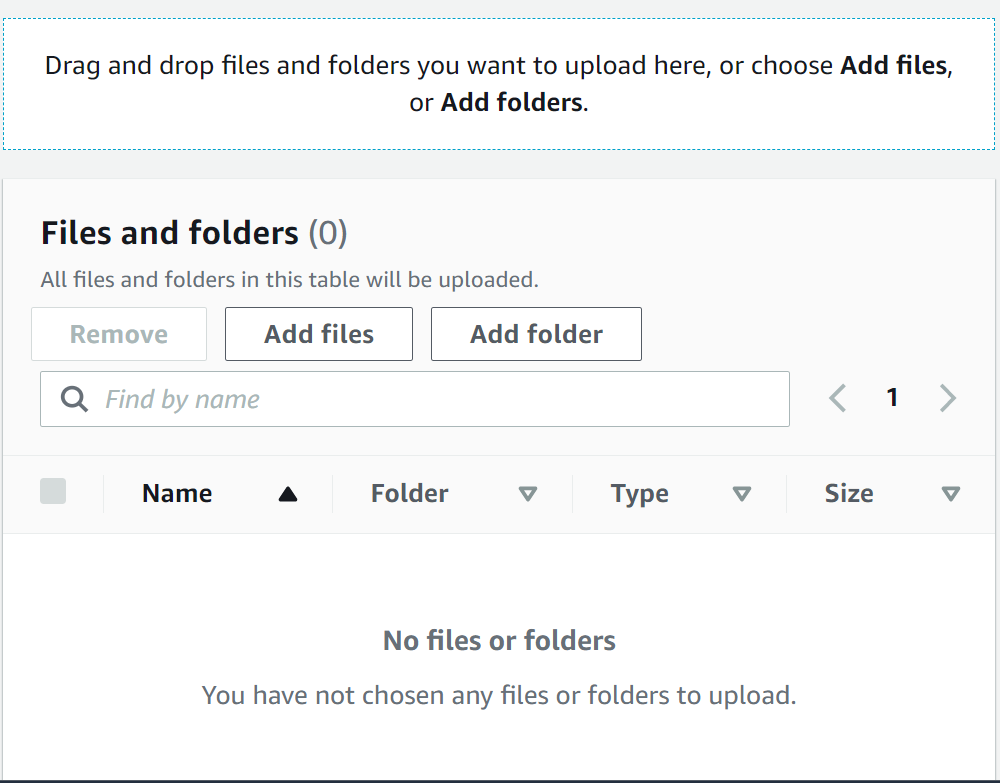
Make sure you select US East (N. Virginia) for the Region. Keep all default settings, there is no need to change anything. Scroll down and click create bucket.



Now, click on the bucket you just created to go in and upload the crime.csv file in the bucket.



Browse and fid the file in your local machine and upload it. Keep all default settings.

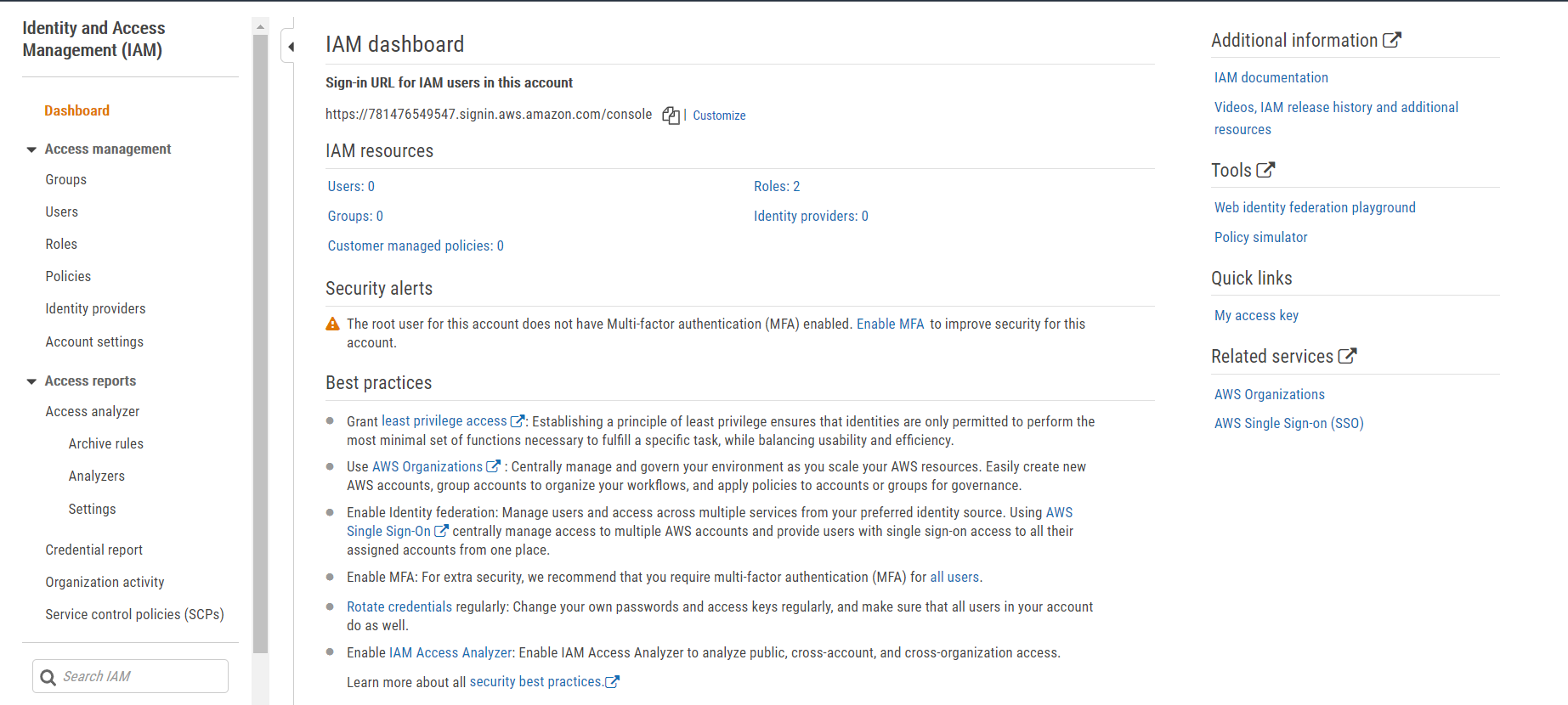


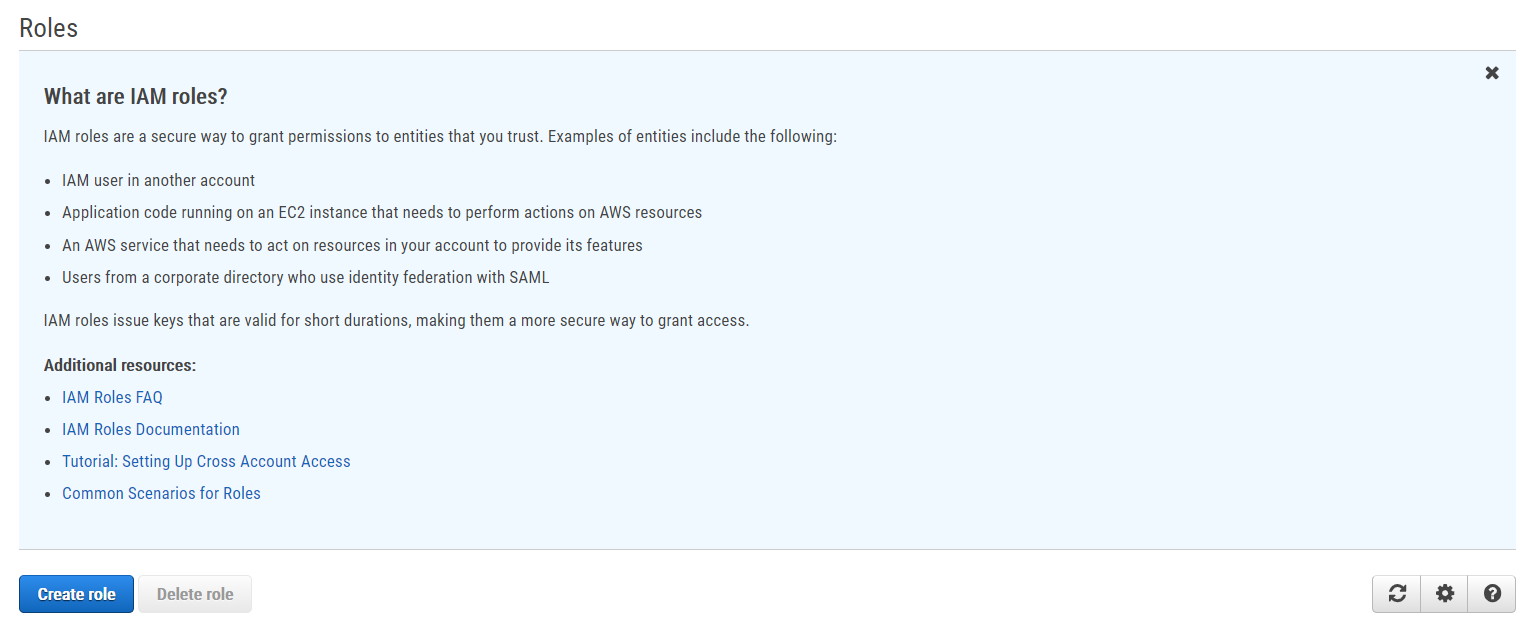
By the end of Step 1, you should have a bucket created and the file crime.csv uploaded inside the bucket.

**Step 2: Create a role in IAM to access S3.**

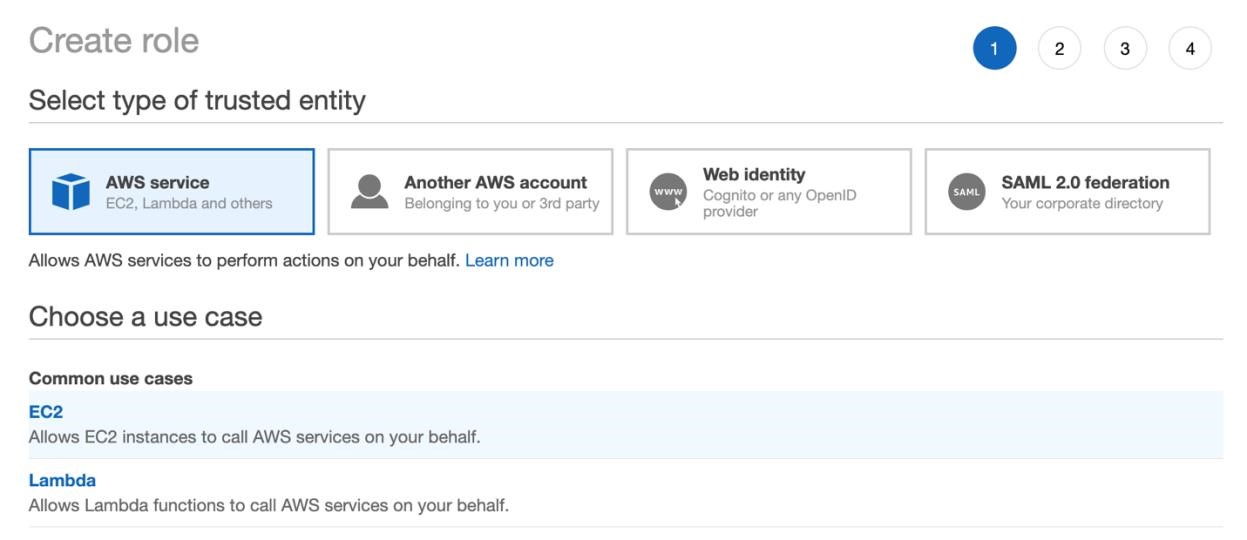
Go to the Services tab, find and go to IAM.

Under the side bar, find Roles and click.



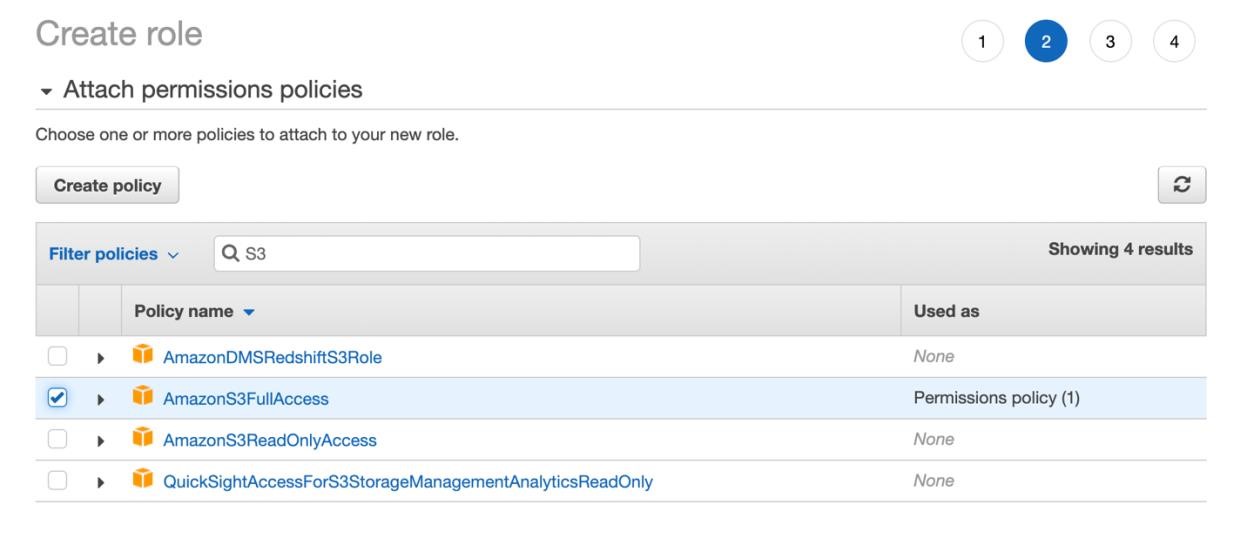


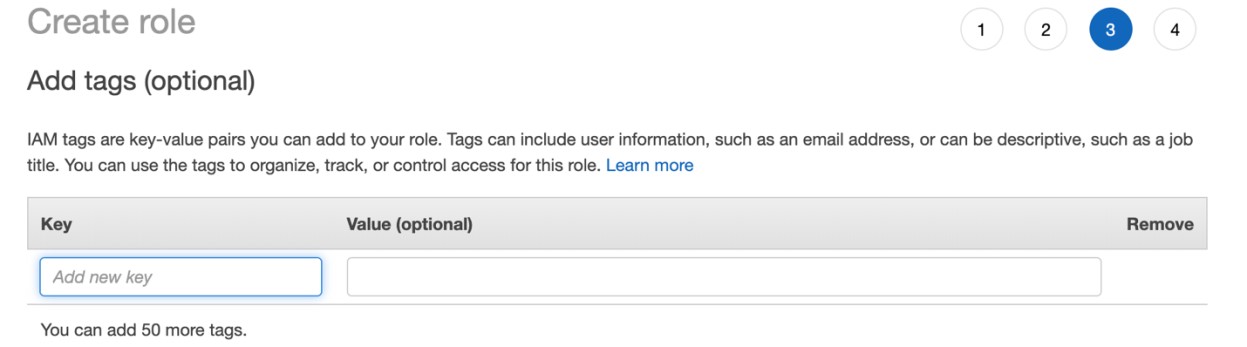
Make sure you select the EC2 under “Choose a use case”.



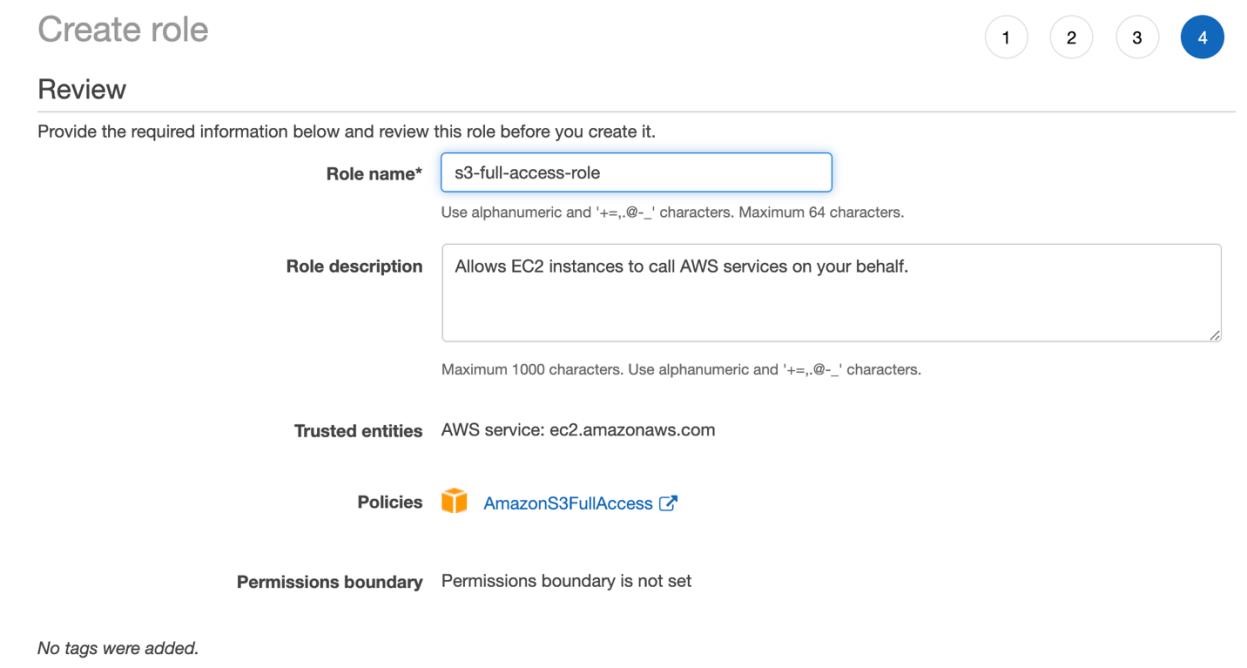
Make sure you search for “S3” and select “AmazonS3FullAccess”.

You can skip the “Add tags”.





Give your role a name and finish up the creation.



Now, you should have a new role created with full access to S3 service.

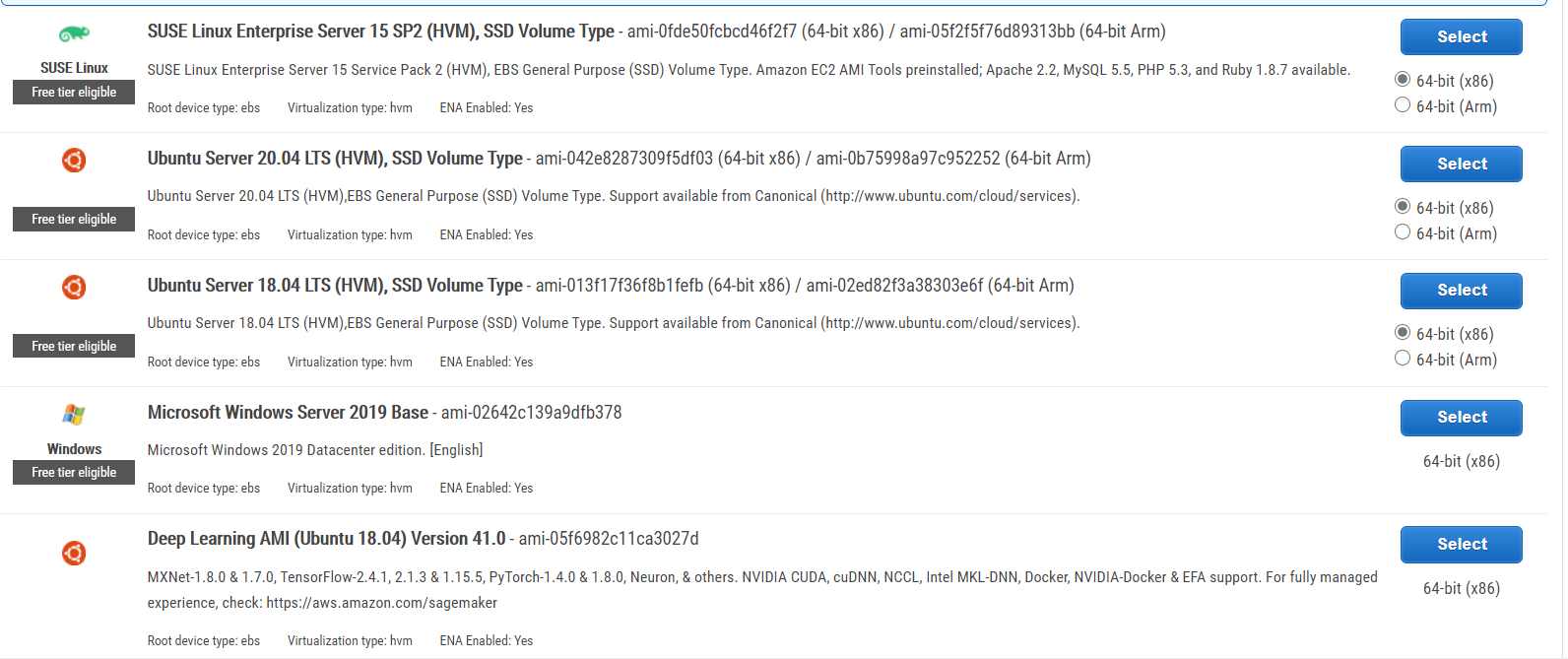
**Step 3: Create an instance using the IAM role we just created.**

Go to Services, find and go to EC2.

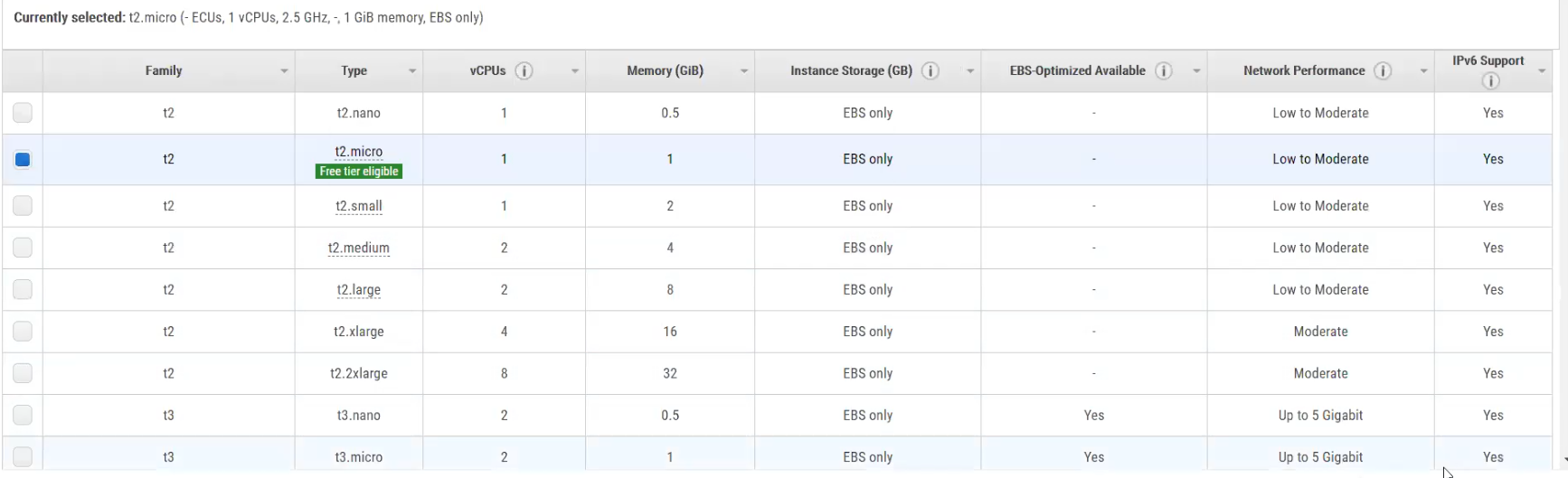
On the dashboard, you should see the “Launch Instances” button. Click on it to create a new instance.



Choose Ubuntu Server 18.04 LTS and click on Select.



Choose t2.mocro and click on Next.



Under the IAM role drill

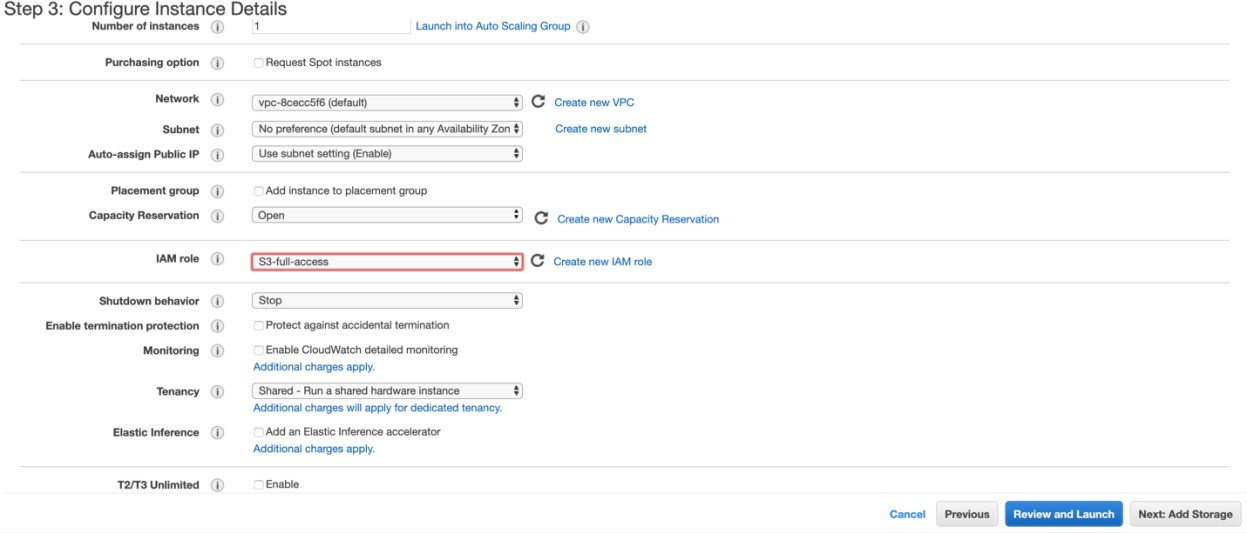
-

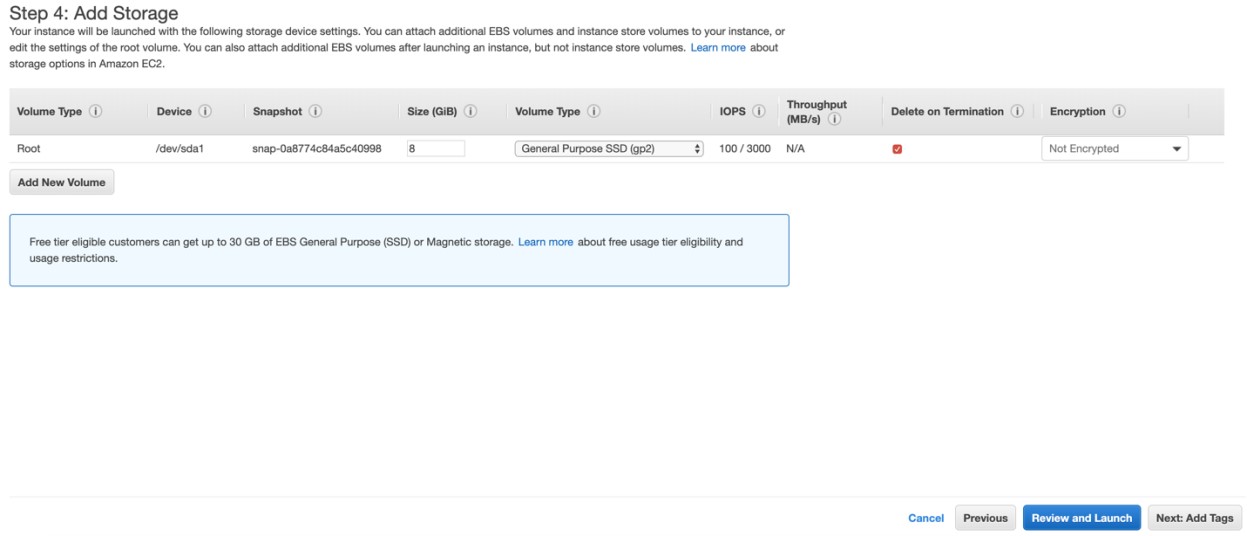
down menu, select the

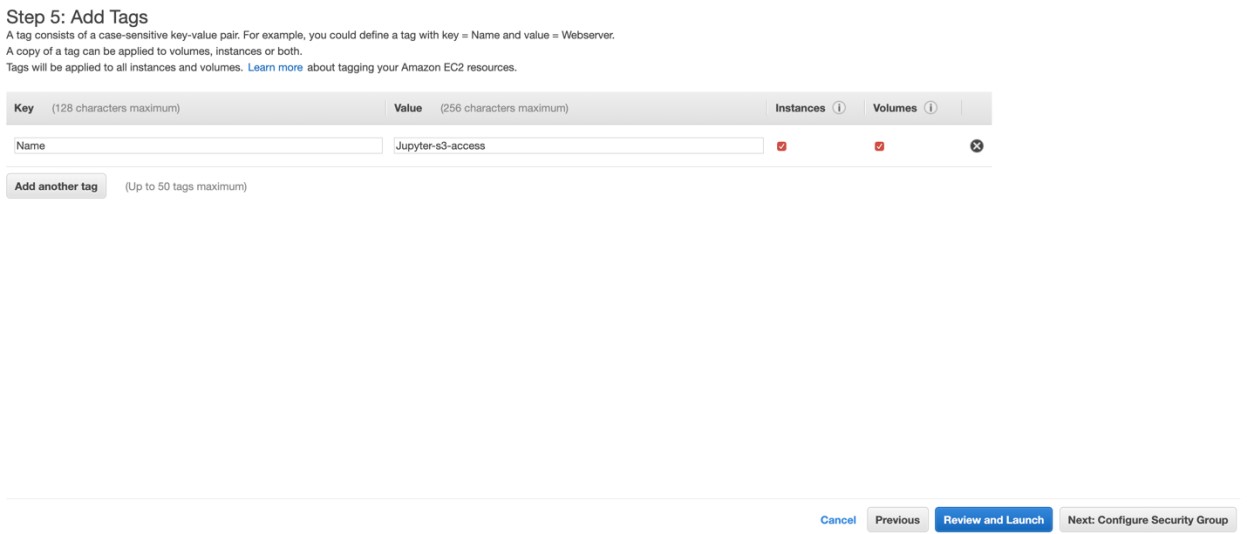
role you just created

and click on Next.

Keep everything default here, just click on next.

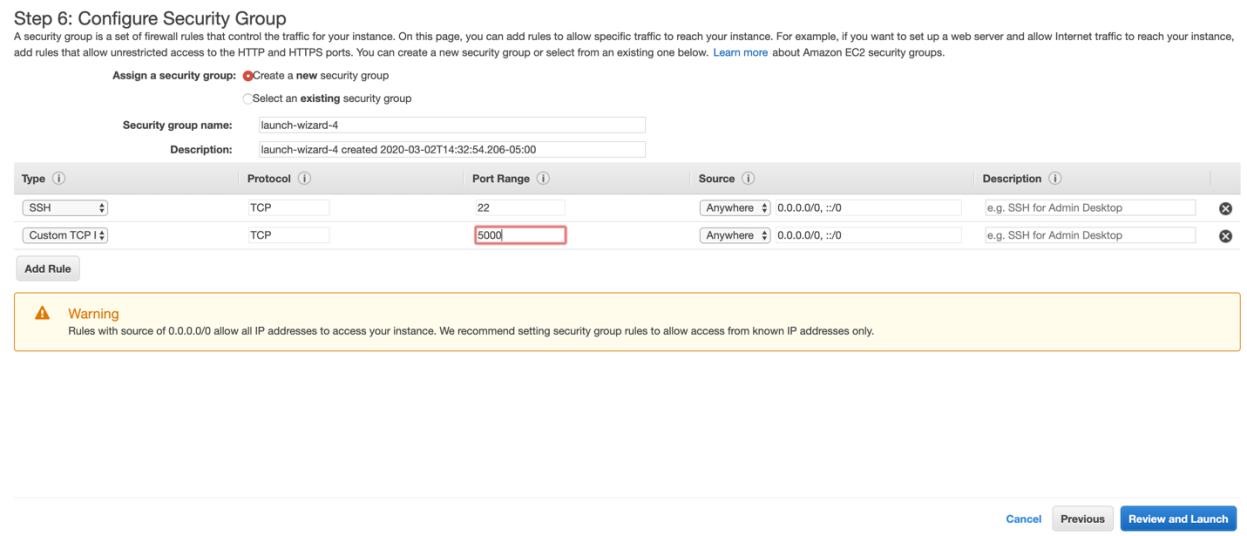


Add a tag, give your instance a name. And click on Next.

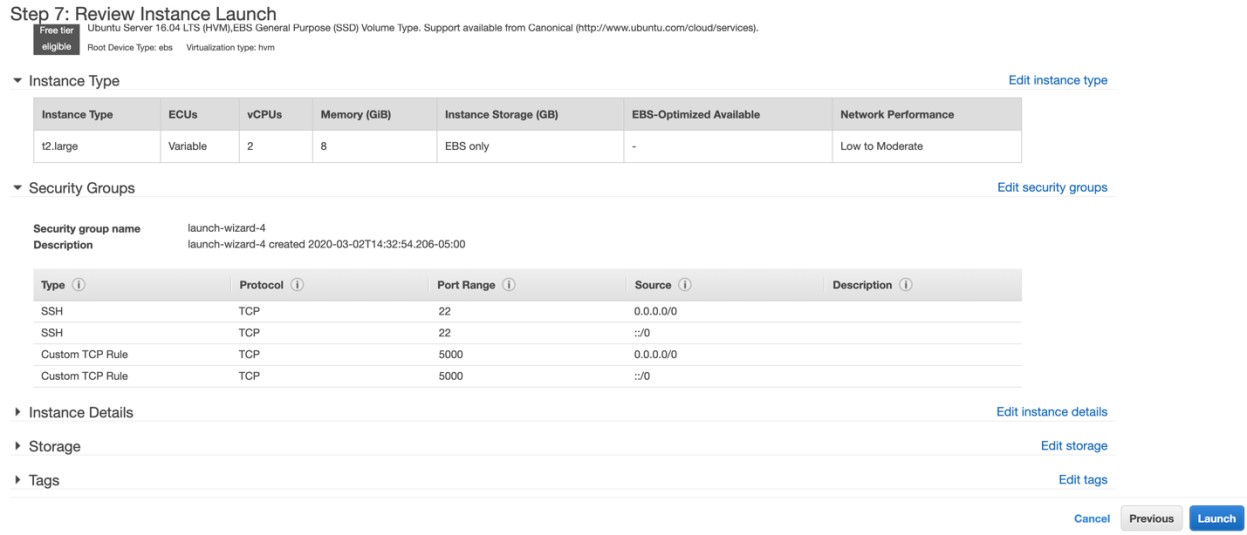


Add Rule, select Custom TCP Rule. Set “Anywhere” for the source and set Port Range to 5000.

And click on Review and Launch.



Nothing to be changed here, just click on Launch.

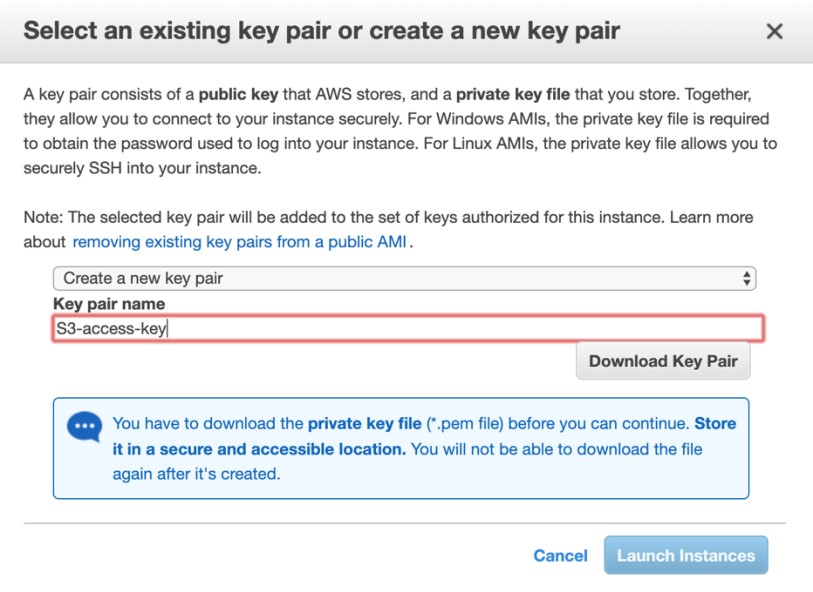


You will see a pop-up window shown below to ask you to select a ssh key pair for you to connect to the instance.

Select Create a new key pair, give it a name. **Click on Download Key Pair**.

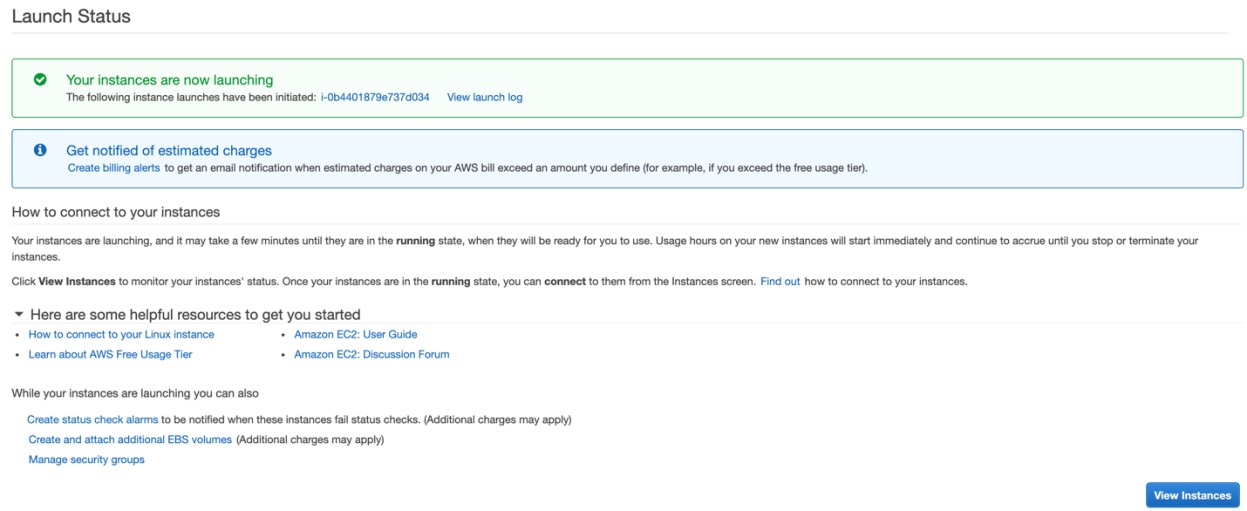
**Note:** Please click on the download button to download the key pair. You have only one chance to download the key pair. Store it somewhere safe on your local machine. Never delete it!!!!!

Then, click on Launch Instance.

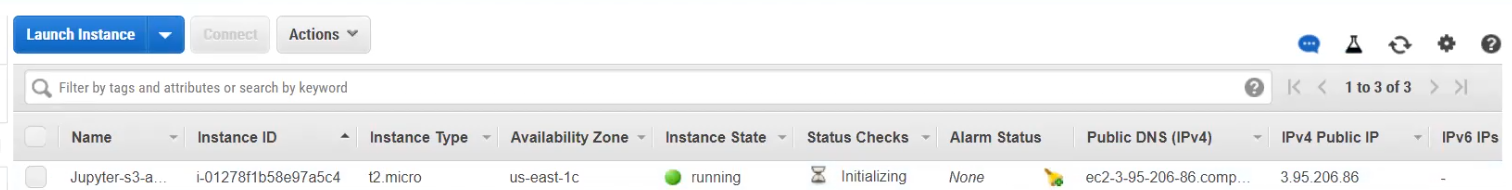


**Download your key pair!!!**

You will see the status of your instance here. Click on View Instances to see more details.

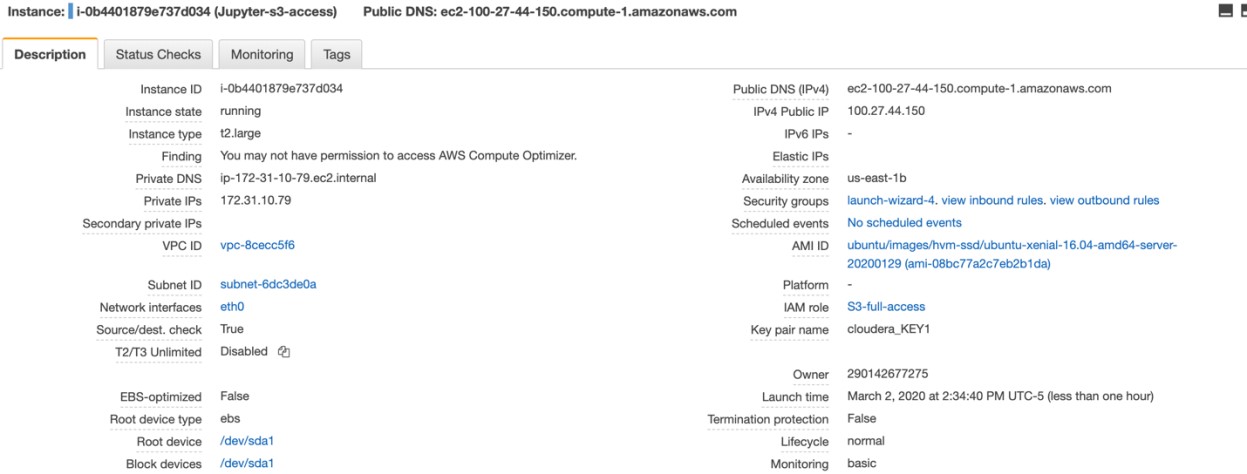


You will see the status of your instance.



All information about your instance are shown below.

Please note the **Public DNS** of your instance, we will use it later to access the web UI.

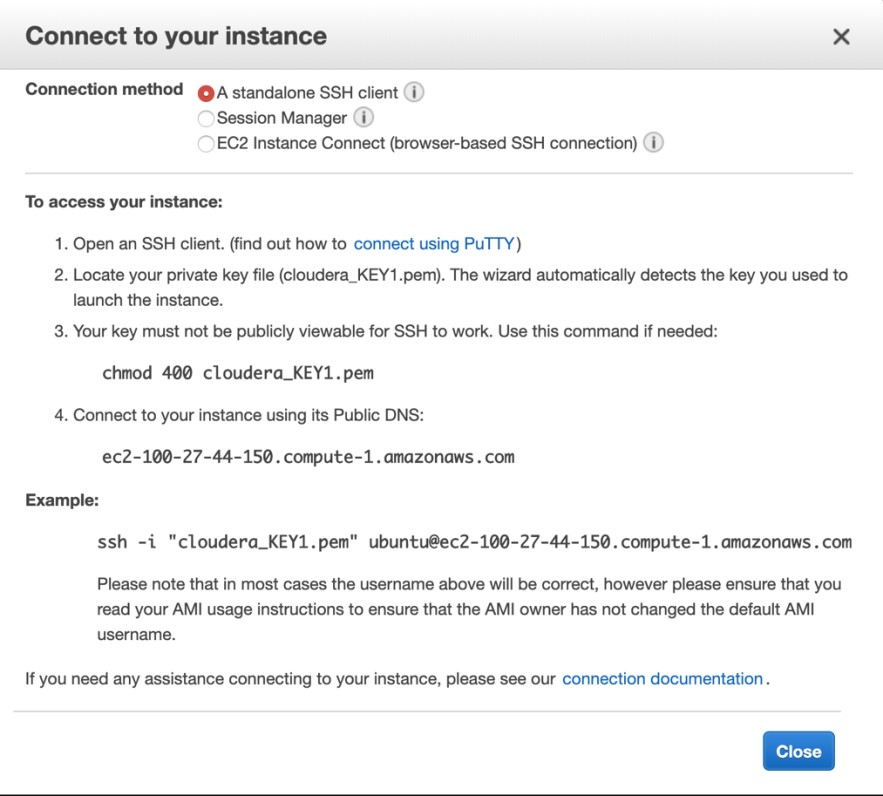


Open the Terminal (Mac)/Git Bash (Windows) from your local machine. Use **cd** command to change your working directory to the folder where you stored the ssh key pair.

Then, click on the Connect button you will see a pop-up window.

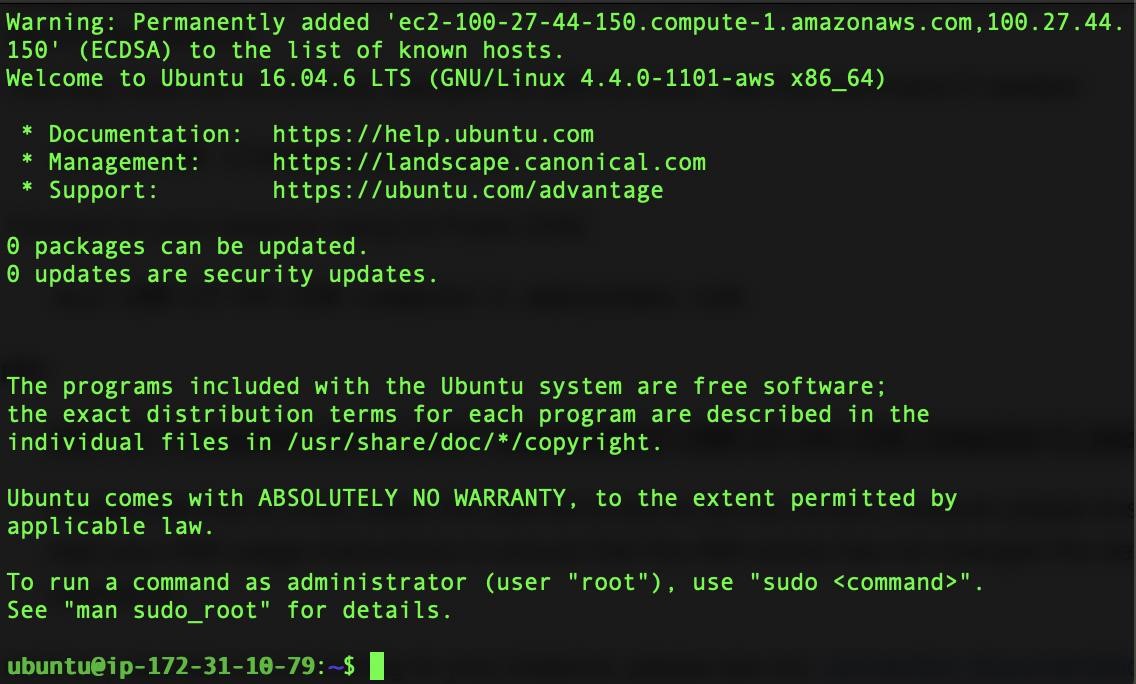
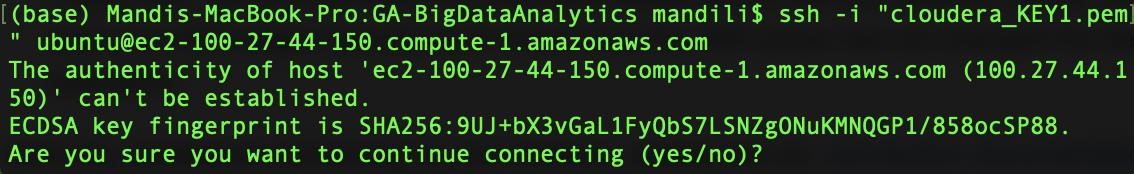


Copy the **ssh -i** command from the pop-up window and run it in your local terminal.



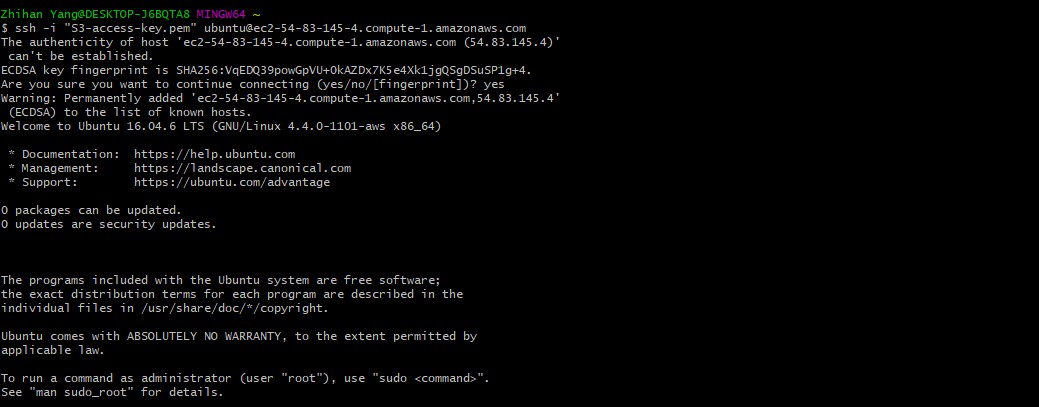
After you run the **ssh** command to connect to your Ubuntu instance, you should see something similar to this.

For Mac users:



For windows users:

You should open your Git Bash and it should be like this:

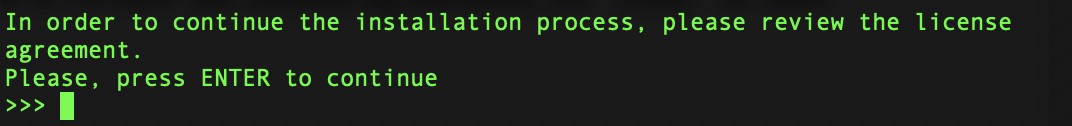


Run the following commands to install Anaconda:

**wget http://repo.continuum.io/archive/**[**Anaconda3-2019.10-Linux-x86\_64.sh**](https://repo.continuum.io/archive/Anaconda3-2019.10-Linux-x86_64.sh)

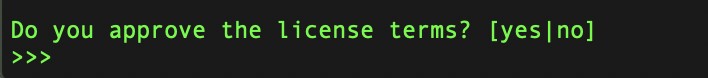
**bas**[**h Anaconda3-2019.10-Linux-x86\_64.sh**](https://repo.continuum.io/archive/Anaconda3-2019.10-Linux-x86_64.sh)

Follow the on-screen instructions. You then will be asked to review the license agreement, hit ‘Enter’ to start reviewing.

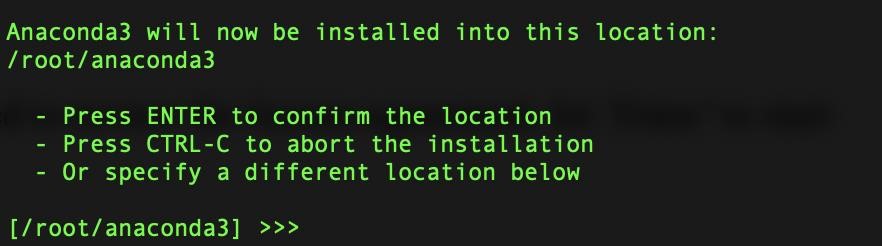




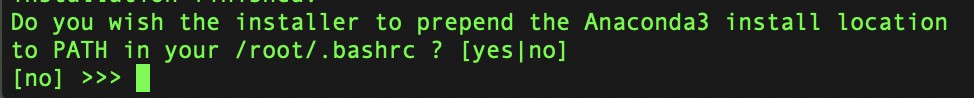
Press enter to skip.



Type in yes to accept the license.



Press enter here to agree on the location for the installation.



Type in yes to accept.

To make use of Anaconda right away, source your bashrc:

**source ~/.bashrc**

Check if you have a Jupyter configuration file:

**ls ~/.jupyter/jupyter\_notebook\_config.py**

If it doesn’t exist, create one:

# jupyter notebook --generate-config

We’re going to add a few lines to your Jupyter configuration file; the file is plain text so, you can do this via your favorite editor (e.g., vim, emacs).

The command to edit the file is:

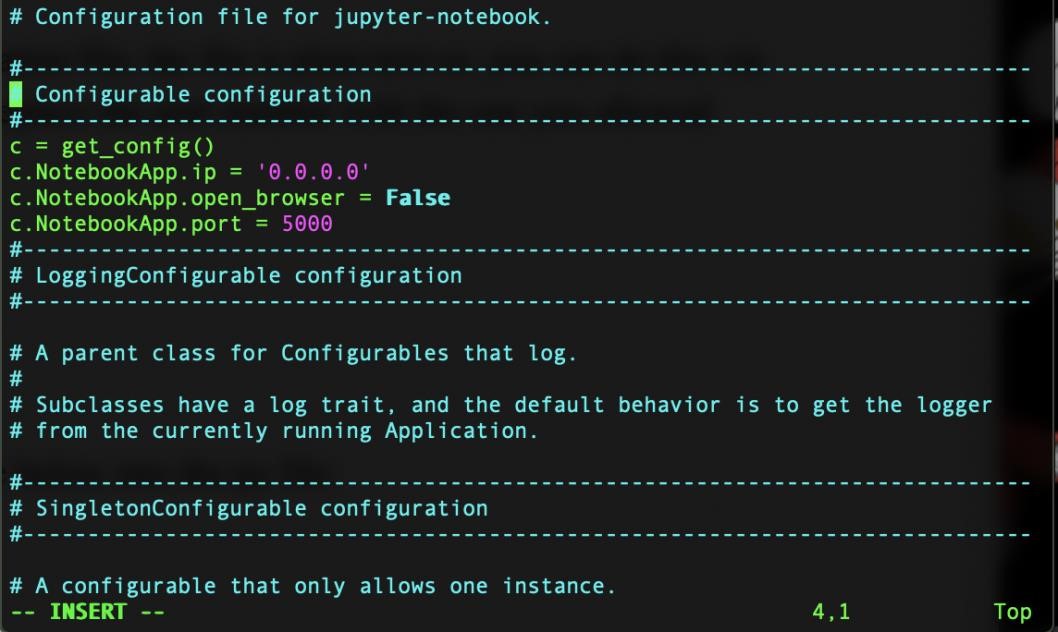
**vi ~/.jupyter/jupyter\_notebook\_config.py**

Change to Insert mode by pressing ‘i’, then type in code below into the py file:

|  |  |  |
| --- | --- | --- |
| **c = get\_config()** |  | |
| **c.NotebookApp.ip = '0.0.0.0'** | |  |
| **c.NotebookApp.open\_browser = False**  **c.NotebookApp.port = <Port Number>** | | |

In our case, the port number should be 5000.

It should look something like this:



Exit the insert mode after you add all the code, and type in “:wq!” to save the file.

To run the jupyter notebook, just type the following command in the ssh window you are in:

**jupyter-notebook --no-browser --port=<PORT-NUMBER>**

Now to launch your Jupyter notebook, just type the following in your browser:

*<Public DNS(IPv4)>:<Port Number>*

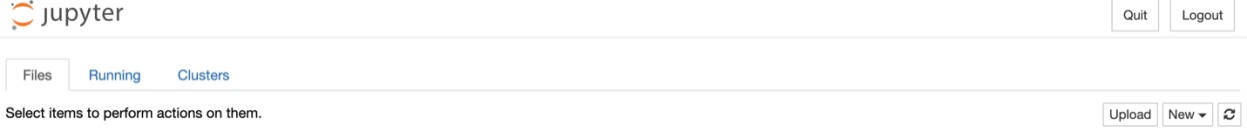
You may need to type in token to login in jupyter notebook. The token is displayed in your Terminal/git Bash where you execute commands to start jupyter notebook.

**Step 4: Access S3 from Jupyter Notebook.**

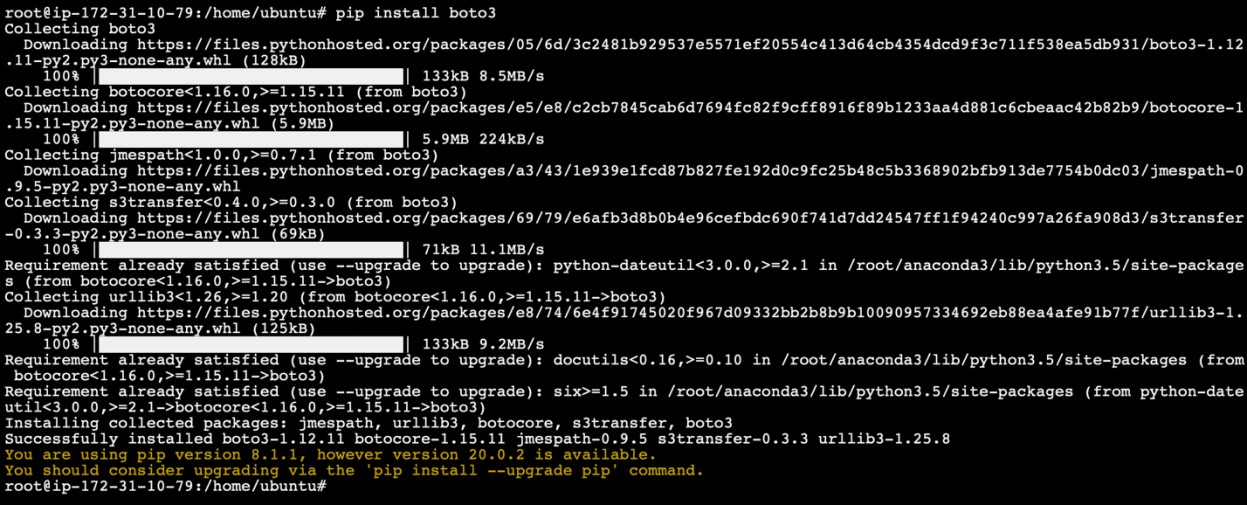
NOTE: If you have Jupyter notebook installed in your local machine, col these step might invoke Jupyter on your computer other than Jupyter on AWS(PLEASE ADD A NOTE)

Open a new terminal in Jupyter notebook and install boto3.

To open a new Terminal on Jupyter notebook, go to New and select Terminal.



Run **pip install boto3**.



Create a new Python3 file. Type in the following code to access files in the bucket and create a data frame.

import boto3

import numpy as np import pandas as pd import dask.dataframe as dd from io import StringIO, BytesIO

s3 = boto3.client("s3")

s3\_resource = boto3.resource('s3')

bucket\_name = "your-bucket-name" #substitute with your bucket name

#list all buckets in our account

response = s3.list\_buckets()

#see the bucket names for bucket in response['Buckets']:

print(bucket['Name'])

response1 = s3.list\_objects\_v2(Bucket=bucket\_name)

response2 = s3.list\_objects\_v2(Bucket=bucket\_name) response2['Contents']

obj = s3.get\_object(Bucket=bucket\_name, Key="crime.csv") df = pd.read\_csv(BytesIO(obj['Body'].read()))