Launch an AWS Deep Learning AMI

with Amazon EC2

In this step-by-step tutorial, you'll learn how to launch an AWS Deep Learning AMI. The AMIs are machine images loaded with deep learning frameworks that make it simple to get started with deep learning in minutes.

Using the AMI, you can train custom models, experiment with new algorithms, and learn new deep learning skills and techniques. The AMIs come with pre-installed open source deep learning frameworks including TensorFlow, Apache MXNet, PyTorch, Chainer, Microsoft Cognitive Toolkit, Caffe, Caffe2, Theano, and Keras, optimized for high performance on Amazon EC2 instances. The AMIs also offer GPU and CPU-acceleration through pre-configured drivers, and come with popular Python packages.

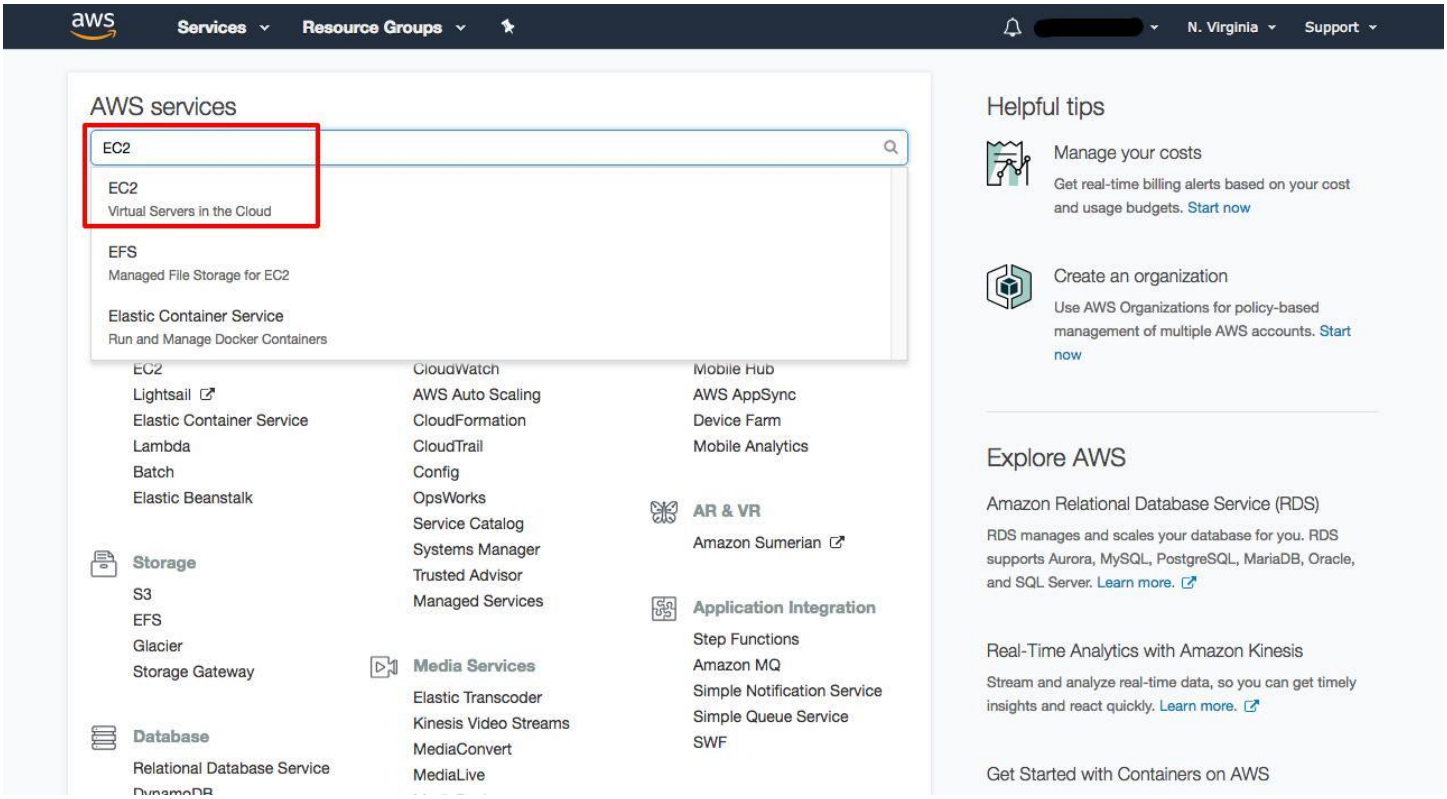
In the next few minutes, you will launch an EC2 instance using a Deep Learning AMI, connect to the instance via SSH, and access a Jupyter Notebook from your workstation.

The cost of doing this tutorial is the charge for the underlying EC2 instance. We recommend using an EC2 C5.large instance which will generate a charge of less than $0.13 per hour until you terminate it. This tutorial will instruct you how to terminate the instance to avoid unnecessary charges.

This tutorial requires an AWS account

Step 1: Open the EC2 Console

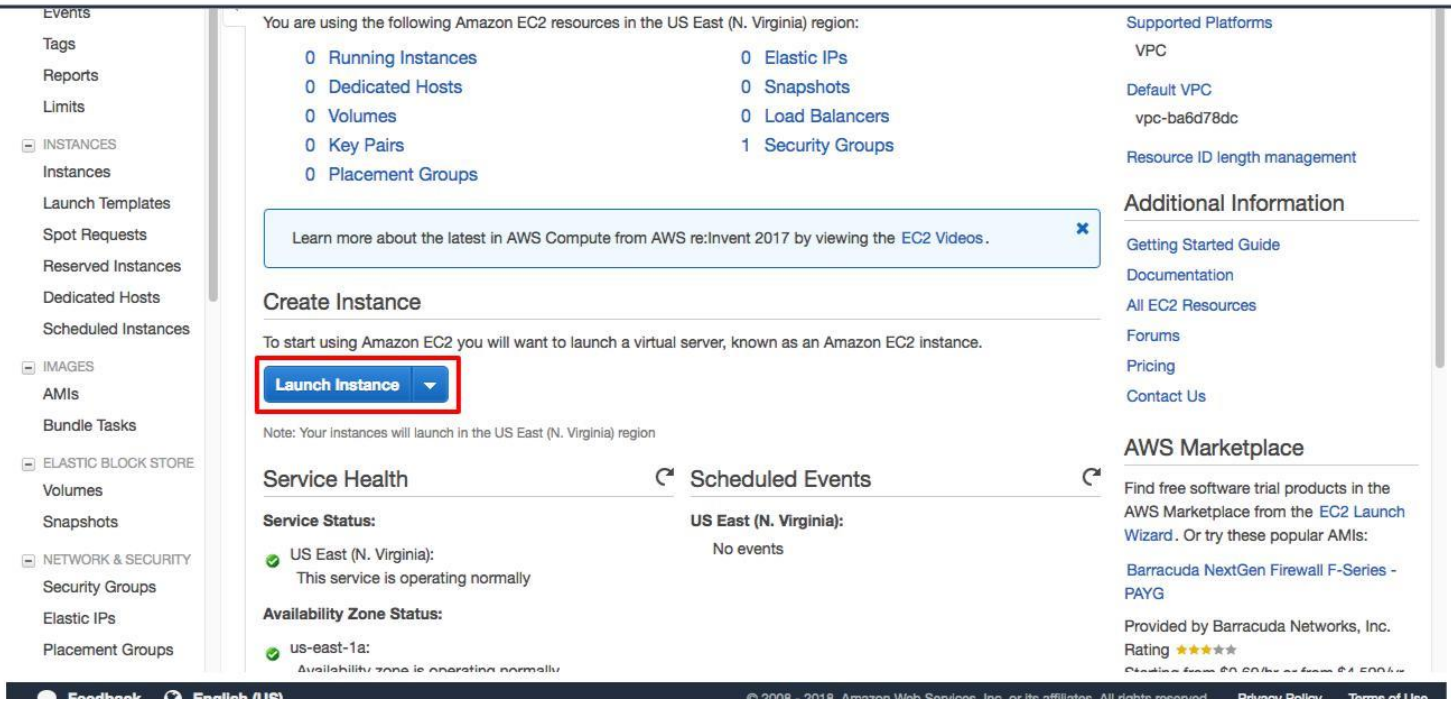
Sign into the AWS Management Console with your user name and password to get started. Then type *EC2* in the search bar and open the EC2 service console.



Step 2: Configure your instance

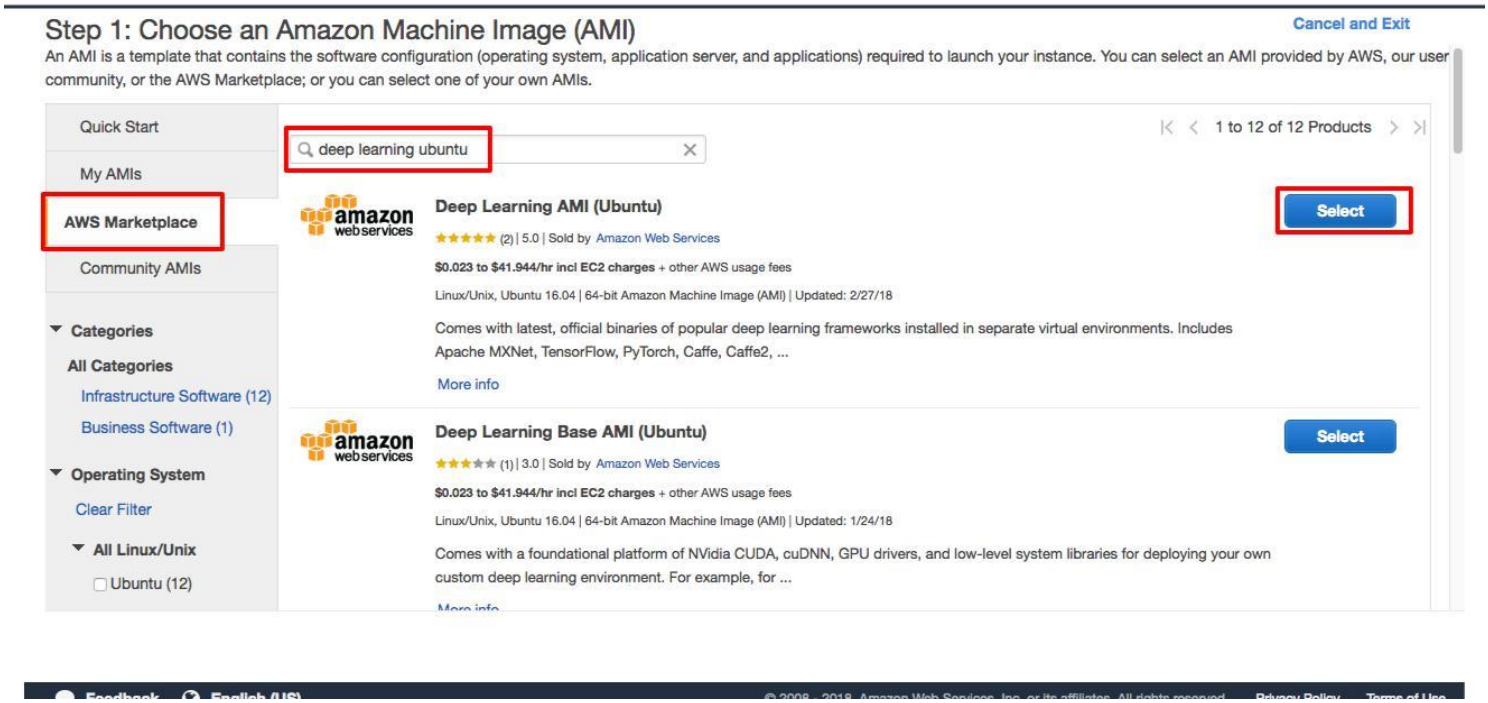
In this step, you will set up a server instance with a machine image for deep learning.

Step 2a: Choose the **Launch Instance** button

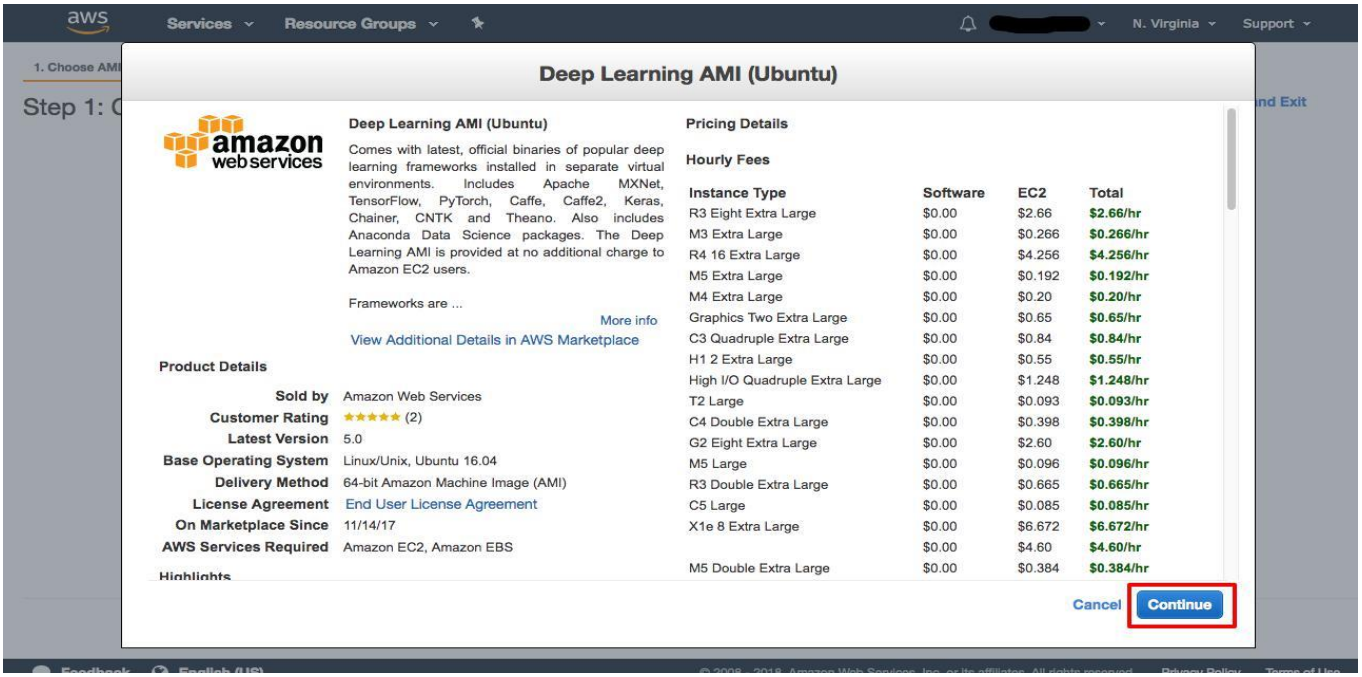


Step 2b: Select a AWS Deep Learning AMI

Choose the **AWS Marketplace** tab on the left, and then search for *deep learning ubuntu*. Select the **Deep Learning AMI (Ubuntu)**. You can also select the Base AMI to set up custom builds of deep learning frameworks. Both AMIs are available for Ubuntu or Amazon Linux.

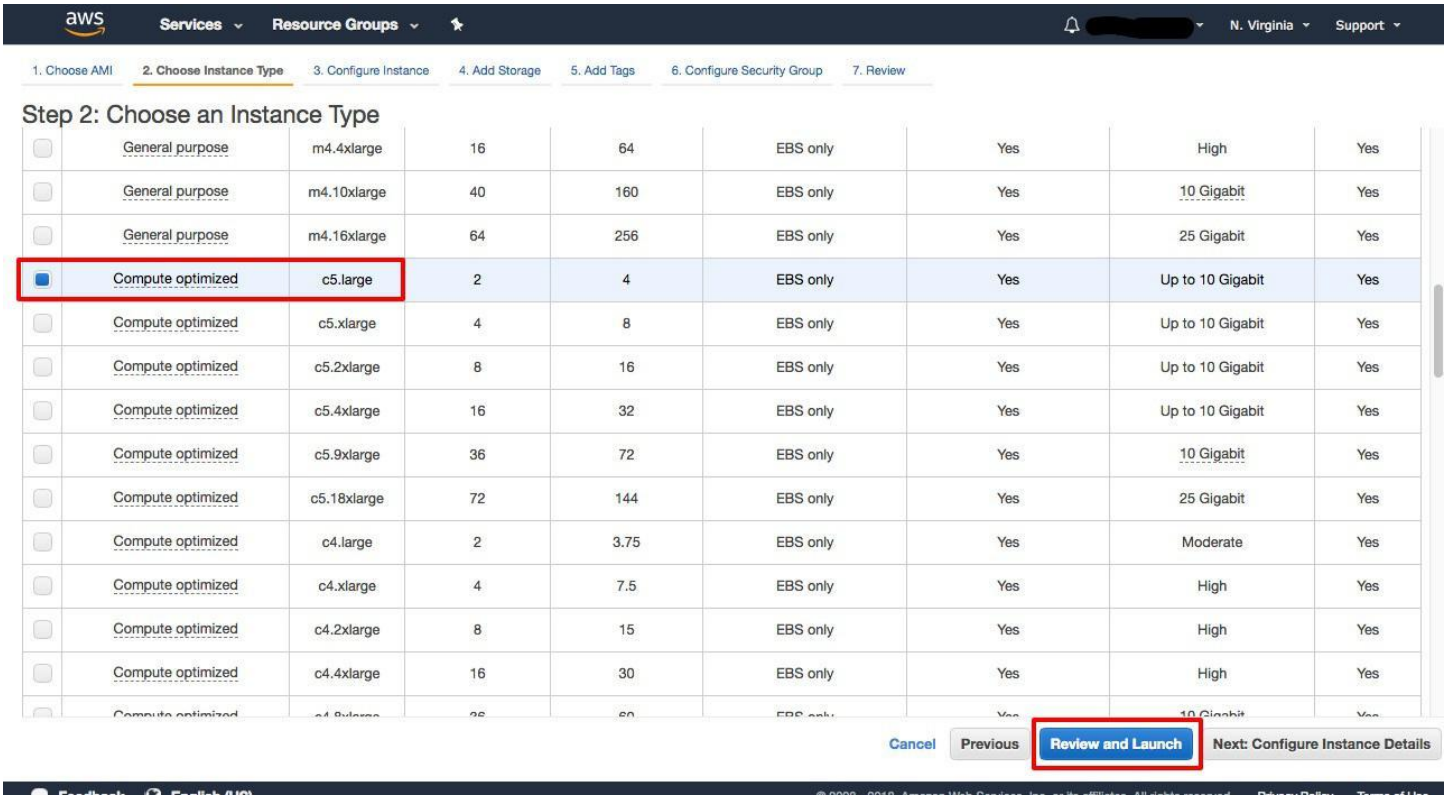


Step 2c: On the details page, choose **Continue**.



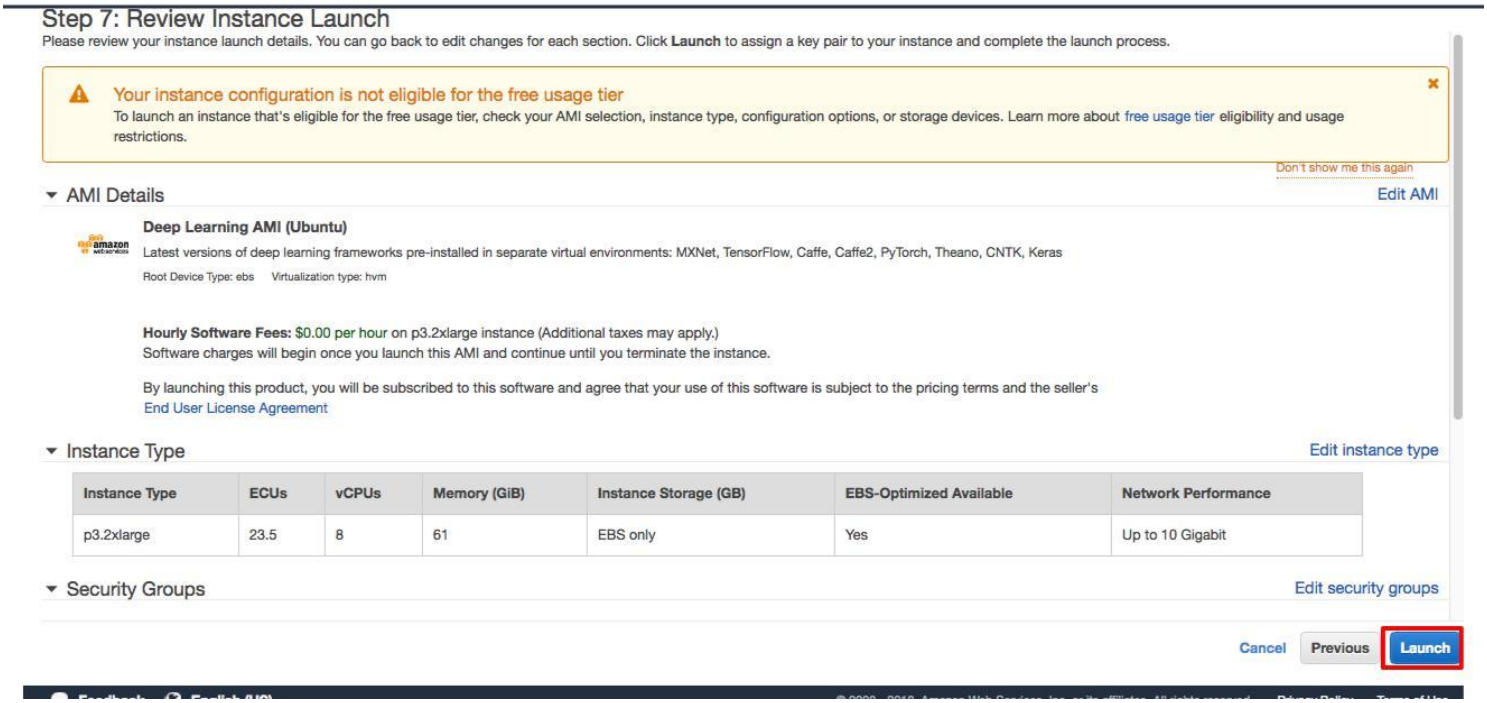
Step 2d: Select an instance type.

Choose an instance type for your deep learning training and deployment needs, and then click **Review and Launch**. Here we have used a c5.large instance, but you can choose from additional instance types including GPU-based P3 instances.

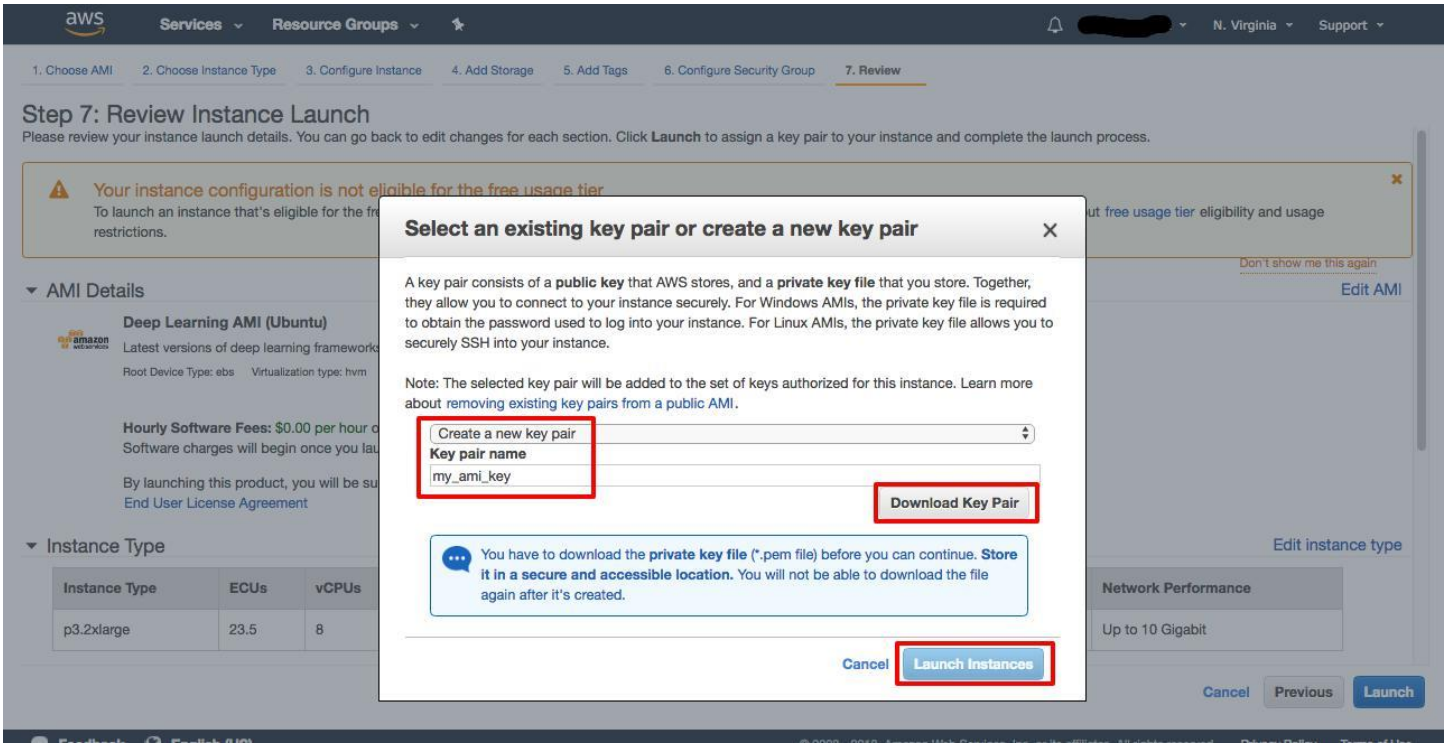


Step 2e: Launch your instance.

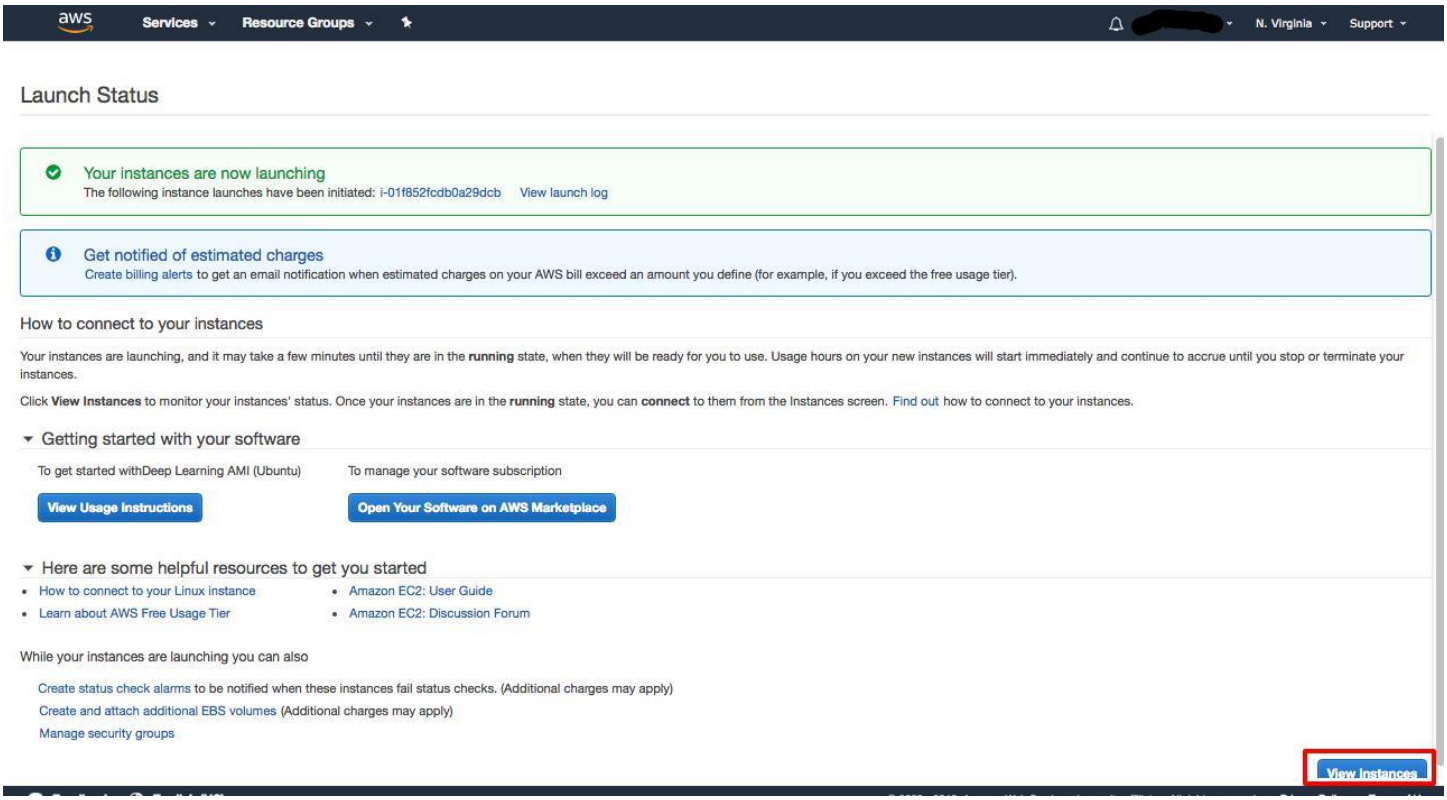
Choose **Launch** on the **Review** page.



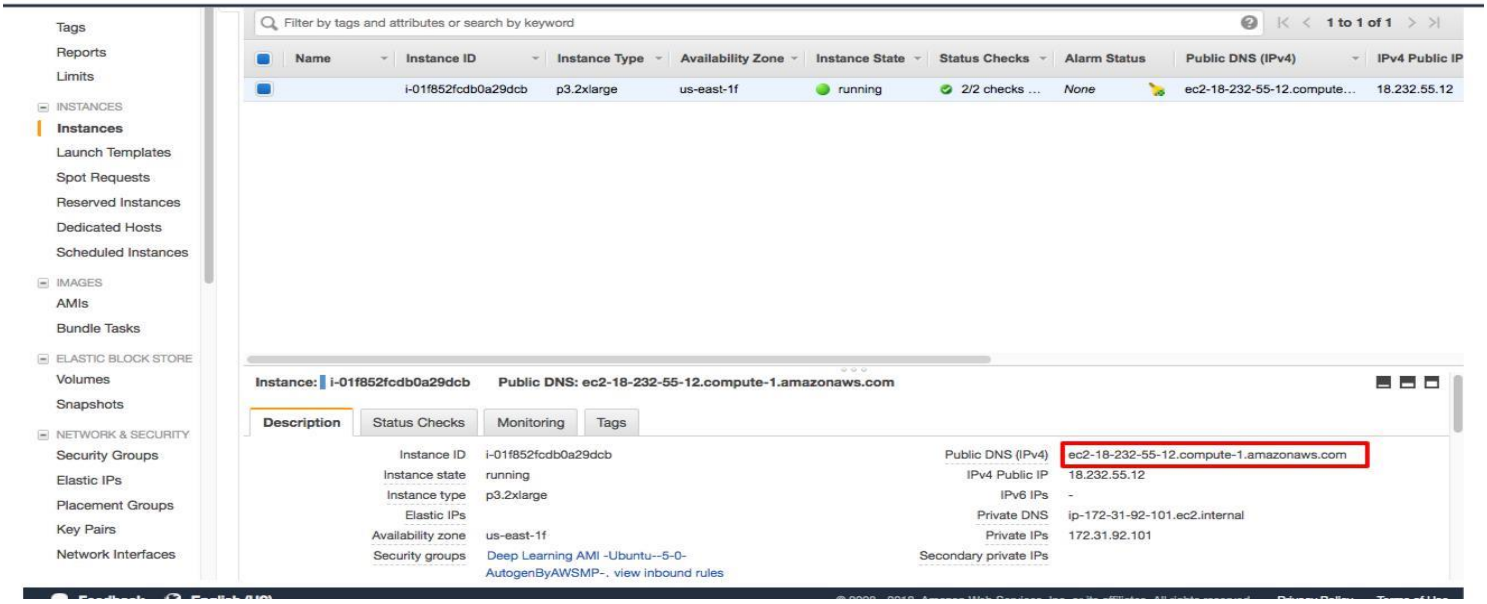
Step 2f: Create a private key file by selecting **Create a new key pair**, and download it to a safe location. Then launch your instance.



Step 2g: Click **View Instance** to see your instance status.



Step 2h: Find your instance's public DNS and copy it.



Step 3: Connect to your instance

Here you will use the command line terminal to communicate with the instance on AWS. If you are using Windows, you can use the Command Prompt or download Git for Windows.

Step 3a: Open your command line terminal.

In the terminal, use the following commands to change to the directory where your security key is located, then connect to your instance using SSH. (NOTE: Replace text below in **bold**)

cd /Users/**your_username**/**Downloads**/

chmod 0400 <**your .pem filename**>

ssh -L localhost:8888:localhost:8888 -i <**your .pem filename**> ubuntu@<**your instance DNS**>

Step 3b: In the terminal, use the command:

Install tmux

mux is a terminal multiplexer. It lets you switch easily between several programs in one terminal, detach them (they keep running in the background) and reattach them to a different terminal.

```
sudo apt-get install tmux
```

tmux
jupyter notebook --ip=0.0.0.0 --port=8888 -–no-browser

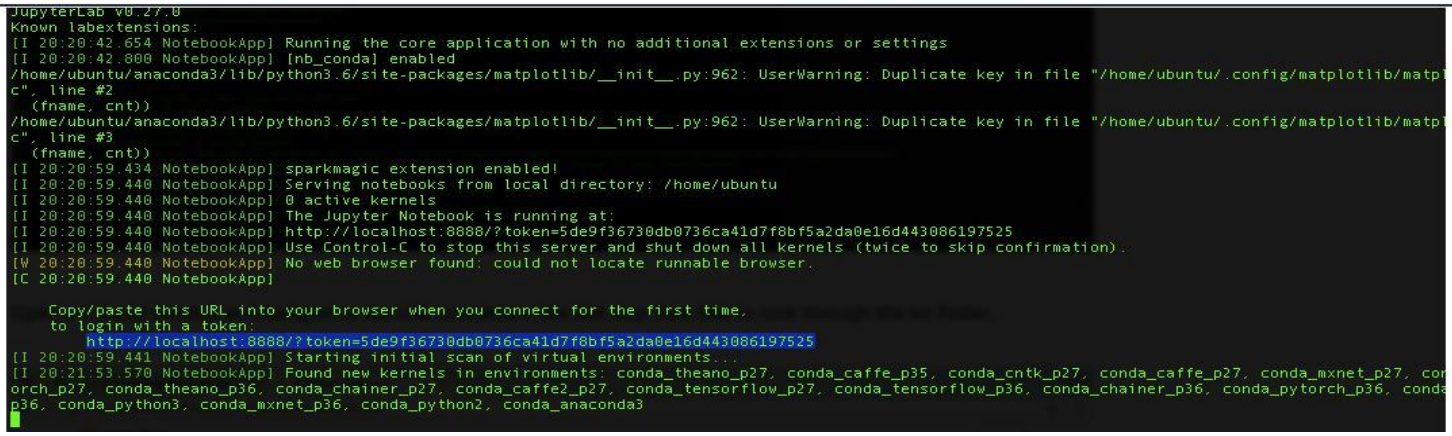
arguments:

--ip=<Unicode> (NotebookApp.ip)
Default: 'localhost'
The IP address the notebook server will listen on.

--port=<Int> (NotebookApp.port)
Default: 8888
The port the notebook server will listen on.

--no-browser
Don't open the notebook in a browser after startup.

Then copy the URL indicated.



Step 4: Access your Jupyter Notebook

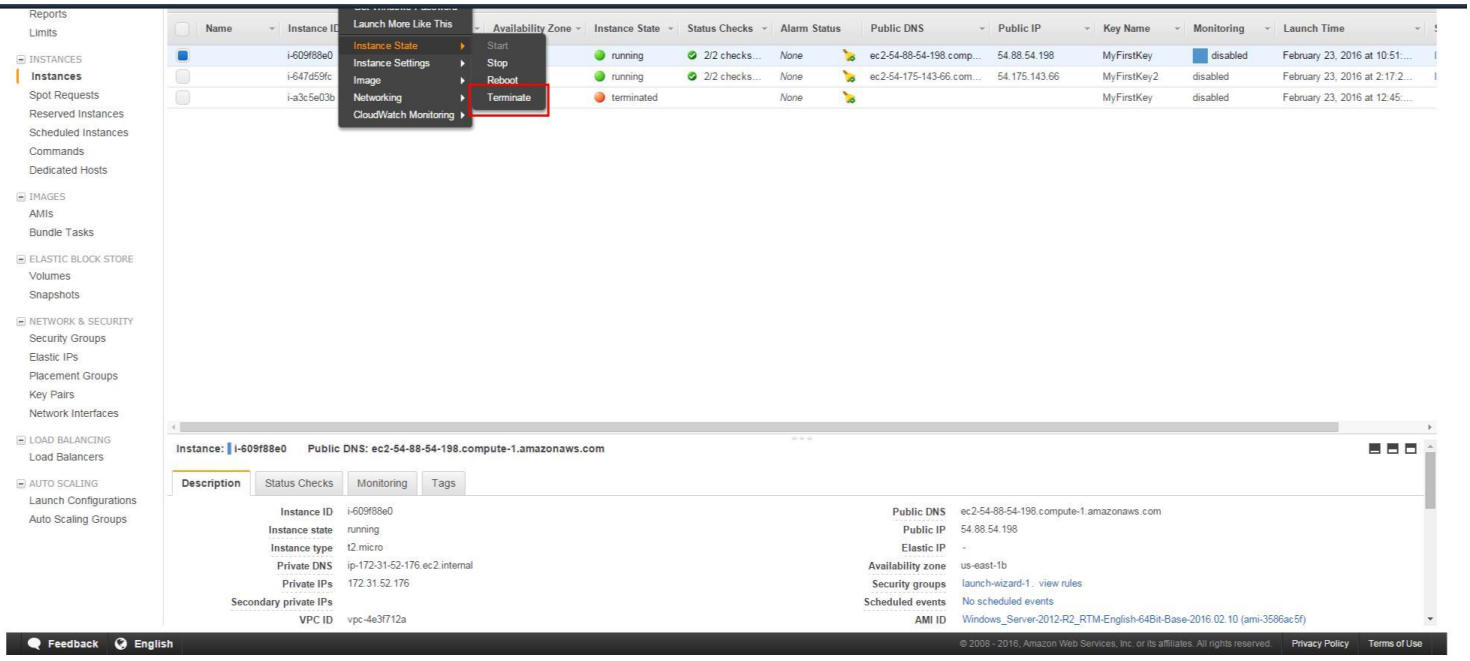
In this step, you will access your Jupyter Notebook to start using a deep learning framework. If you are connecting to a Jupyter Notebook from a Windows client, you can follow the steps listed here. (https://docs.aws.amazon.com/dlami/latest/devguide/setup-jupyter-configure-client-windows.html)

Open a browser window and navigate to the URL indicated in the last step. Choose **New** and start a new notebook using any popular deep learning framework including TensorFlow, Apache MXNet and Gluon, Torch, PyTorch, Caffe, Caffe2, Keras, Theano, Microsoft Cognitive Toolkit, and Chainer.

Step 5: Terminate your instance

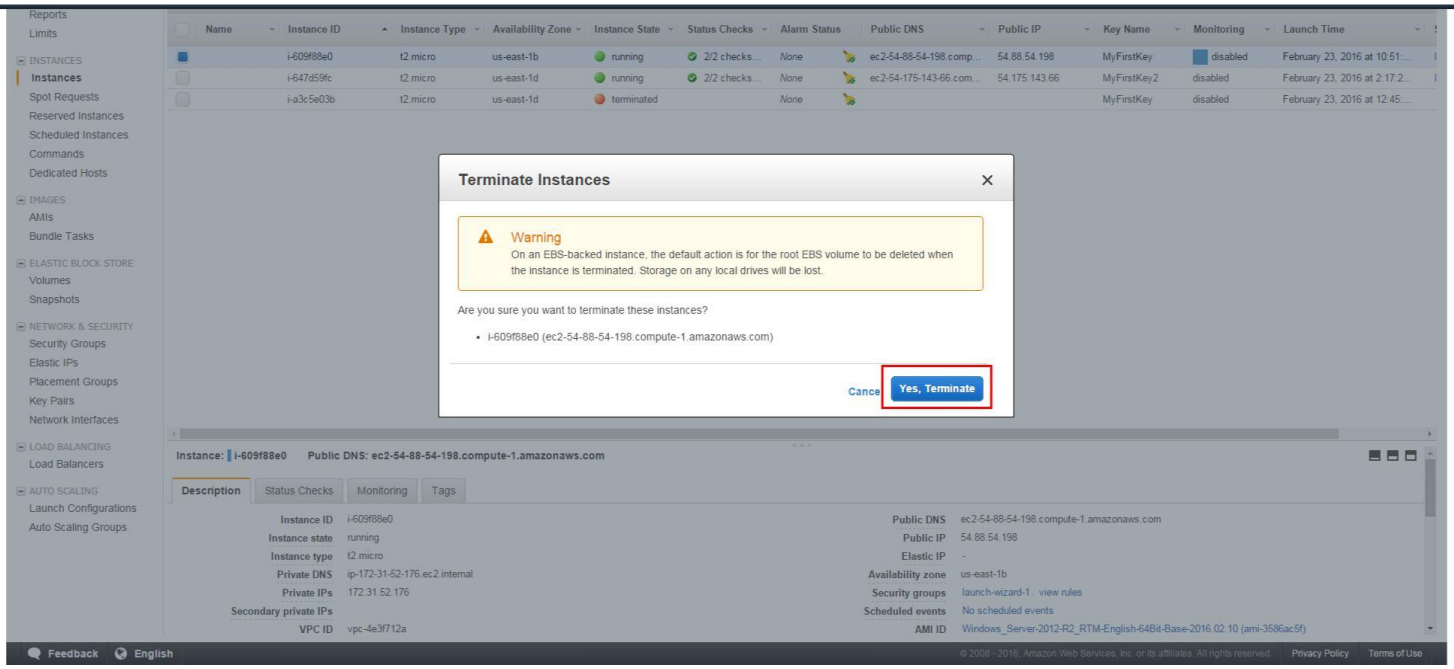
Once you're finished, you can easily terminate the instance from the EC2 console. It is a best practice to terminate instances you are no longer using so you don't keep getting charged for them.

Step 5a: Back on the EC2 Console, select the box next to the instance you created. Then click the **Actions** button, navigate to *Instance State*, and click **Terminate**.



Step 5b: You will be asked to confirm your termination - **Yes, Terminate.**

Note: This process can take several seconds to complete. Once your instance has been terminated, the Instance State will change to *terminated* on your EC2 Console.



Now that you've launched an AWS Deep Learning AMI, you can easily run tutorials for computer vision, natural language processing, recommendation systems, and more using the deep learning framework of your choice. Happy modeling!