# Cloud Computing

## Overview



Training and evaluating deep neural networks is a computationally intensive task. For modestly sized problems and datasets it may be possible to train your network on the CPU in your local machine, but it could take anywhere from 15 minutes to several hours depending on the number of epochs, the size of the neural networks, and other factors. A faster alternative is to train on a [GPU (Graphics Processing Unit)](https://en.wikipedia.org/wiki/Graphics_processing_unit), which is a type of processor that supports greater parallelism.

If you do not already have a computer with a built-in NVIDIA GPU, we suggest you use an [Amazon EC2](https://aws.amazon.com/ec2/) instance. There are many cloud service providers that offer equivalent functionality, but EC2 is a reasonable default that is available to most students. In the next few sections, we'll go over the steps from nothing to running a neural network on an Amazon server.

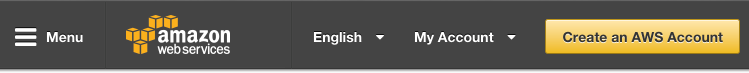
**Note: Please skip this section if you are planning to use your own GPU or CPU (or otherwise not planning to use Amazon EC2).**

## Note: We provide AWS credits to DLND students, but currently not to MLND students.

## Create an AWS Account

**Create an AWS Account**

Visit [aws.amazon.com](https://aws.amazon.com/) and click on the "Create an AWS Account" button.



If you have an AWS account already, sign in.

If you do not have an AWS account, sign up.

When you sign up, you will need to provide a credit card. But don’t worry, you won’t be charged for anything yet.

Furthermore, when you sign up, you will also need to choose a support plan. You can choose the free Basic Support Plan.

Once you finish signing up, wait a few minutes to receive your AWS account confirmation email. Then return to [aws.amazon.com](https://aws.amazon.com/) and sign in.

## Get Access to GPU Instances

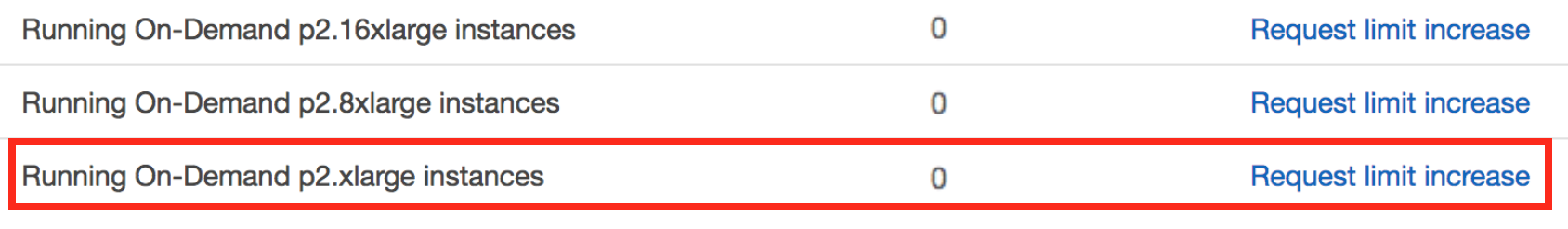
**View Your Current Limit**

Amazon Web Services has a service called [Elastic Compute Cloud (EC2)](https://aws.amazon.com/ec2), which allows you to launch virtual servers (or “instances”), including instances with attached GPUs. The specific type of GPU instance you should launch for this tutorial is called “p2.xlarge”.

We will use [this AMI (Amazon Machine Image)](https://aws.amazon.com/marketplace/pp/B01M0AXXQB#product-description) to define the operating system for your instance, and to make use of its pre-installed software. In order to use this AMI, you **must** change your AWS region to one of the following (and you are encouraged to select the region in the list that is closest to you):

* EU (Ireland)
* Asia Pacific (Seoul)
* Asia Pacific (Tokyo)
* Asia Pacific (Sydney)
* US East (N. Virginia)
* US East (Ohio)
* US West (Oregon)

**After** changing your AWS region, view your EC2 Service Limit report at: <https://console.aws.amazon.com/ec2/v2/home?#Limits>, and find your "Current Limit" for the p2.xlarge instance type.

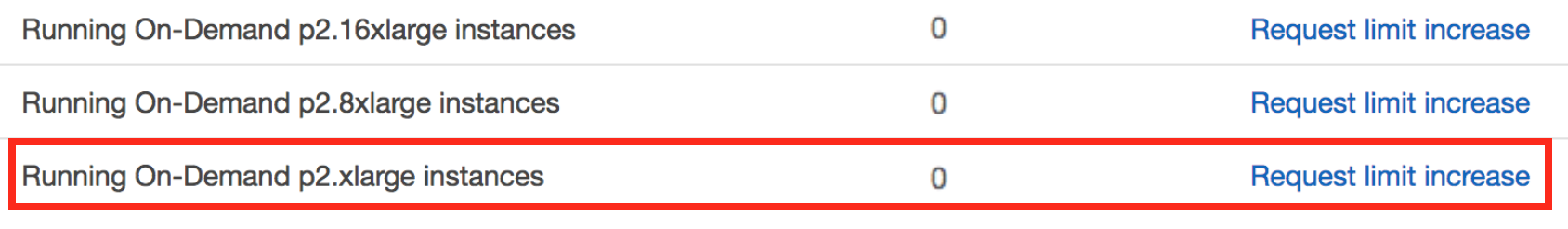


By default, AWS sets a limit of 0 on the number of p2.xlarge instances a user can run, which effectively prevents you from launching this instance.

**Submit a Limit Increase Request**

If your limit of p2.xlarge instances is 0, you'll need to increase the limit before you can launch an instance. From the EC2 Service Limits page, click on “Request limit increase” next to “p2.xlarge”.

You will not be charged for requesting a limit increase. You will only be charged once you actually launch an instance.

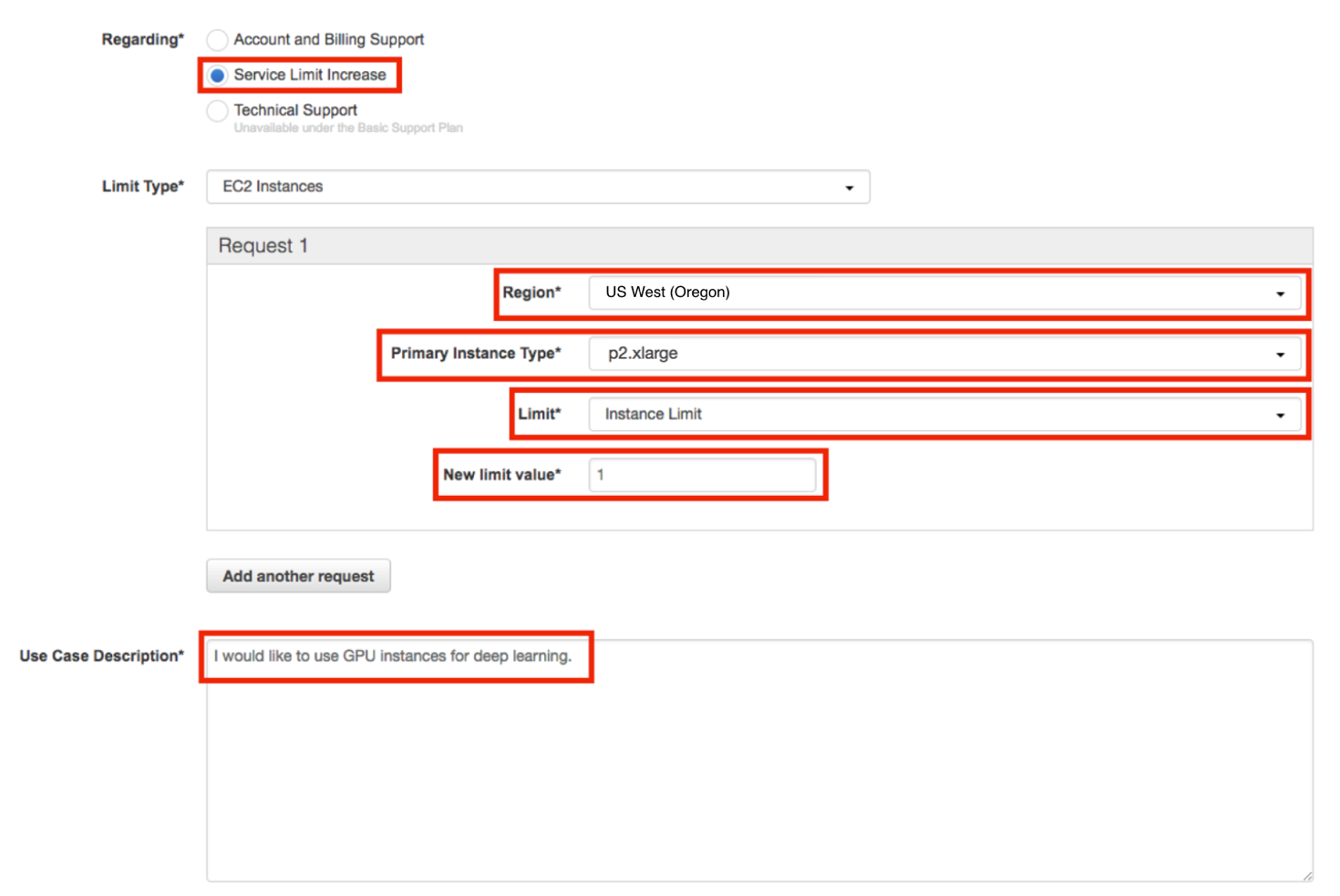


On the service request form, you will need to complete several fields.

For the “Region” field, select the AWS region you chose in **Step 2**.

For the “New limit value”, enter a value of 1 (or more, if you wish).

For the “Use Case Description”, you can simply state: “I would like to use GPU instances for deep learning.”



Note: If you have never launched an instance of any type on AWS, you might receive an email from AWS Support asking you to initialize your account by creating an instance before they approve the limit increase.

**Wait for Approval**

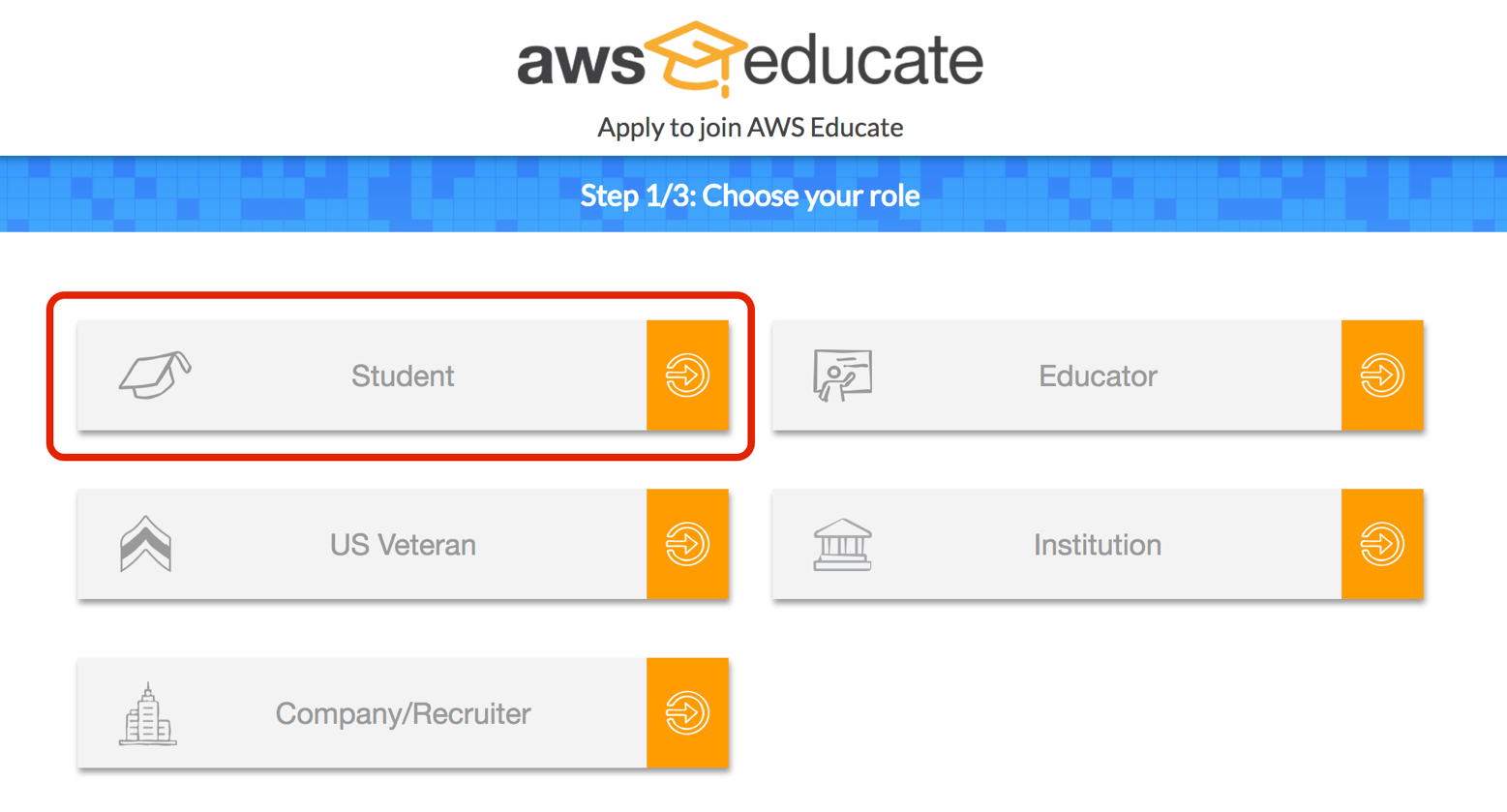
You must wait until AWS approves your Limit Increase Request. AWS typically approves these requests within 48 hours.

## Apply Credits

**Apply Credits**

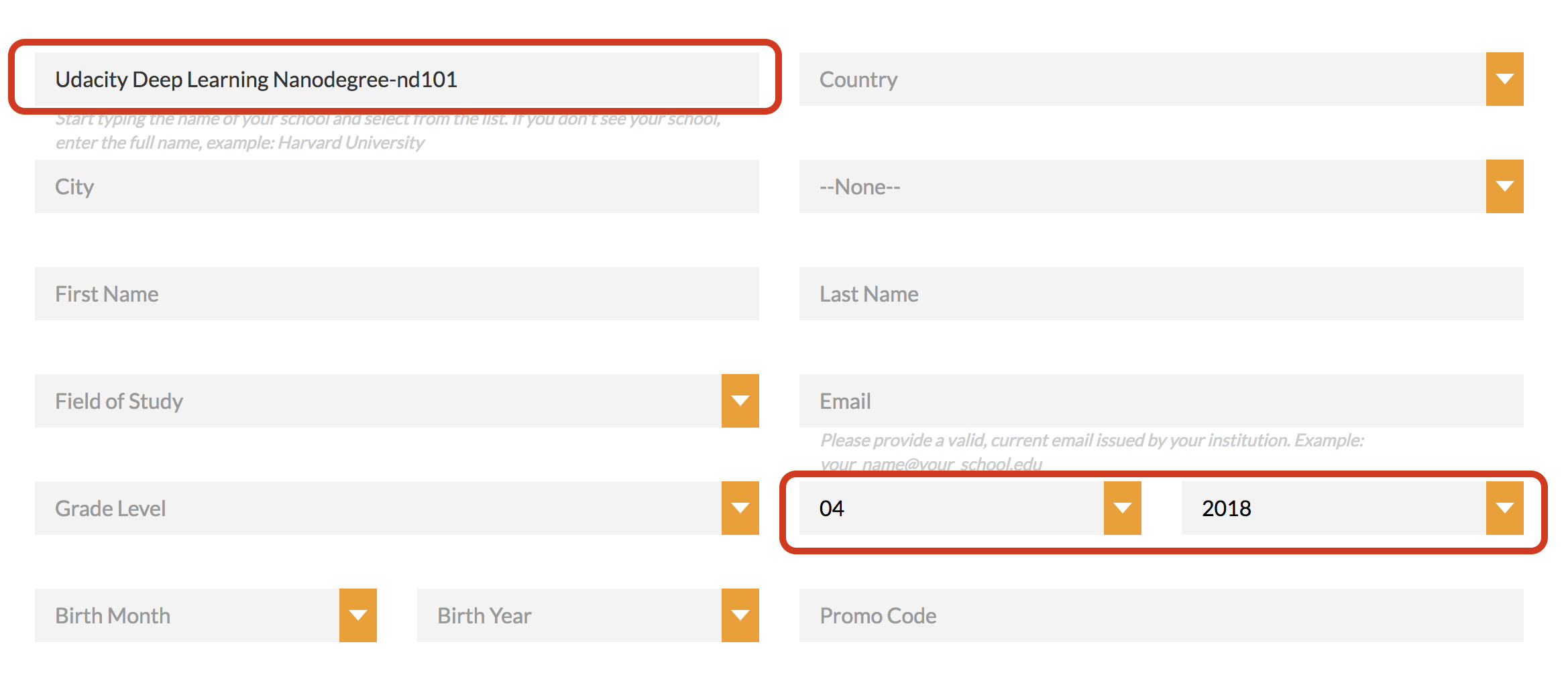
All DLND students are provided with AWS credits to train their neural networks. The link to get your AWS credit code has been emailed to you. Check the email address you used to sign up for Udacity. After clicking the link, it should take you to a page that looks like the one below.

From this page, apply as a **Student**.



The link you clicked should generate Udacity Deep Learning Nanodegree-nd101 as the **Institution Name**. Double check to make sure this is correct. If necessary, update this field accordingly.

You'll need to specify your term end date as your **Graduation Month** and **Graduation Year**, similar to the image below.



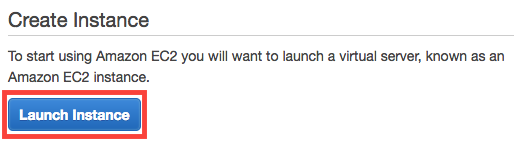
After you've gone through all the steps, you'll receive an email with your code. The email will be sent to the email address you entered on the AWS credits application. It may take 48 hours to receive this email, though it is much quicker in most cases

The code is labeled "Credit Code" in the email. Use this code on your [account credits page](https://console.aws.amazon.com/billing/home?#/credits).

## Launch An Instance

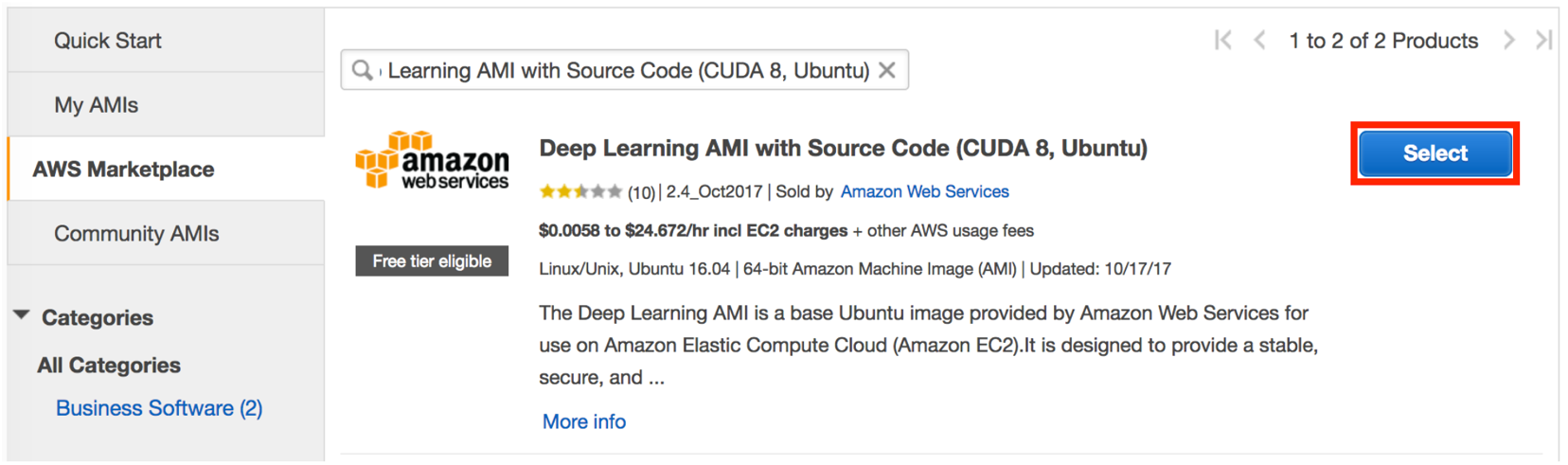
Once AWS approves your GPU Limit Increase Request, you can start the process of launching your instance.

Visit the [EC2 Management Console](https://console.aws.amazon.com/ec2/v2/home), and click on the “Launch Instance” button.



Next, you must choose an AMI (Amazon Machine Image) which defines the operating system for your instance, as well as any configurations and pre-installed software.

Click on **AWS Marketplace**, and search for **Deep Learning AMI with Source Code (CUDA 8, Ubuntu)**. Once you find the appropriate AMI, click on the "Select" button.

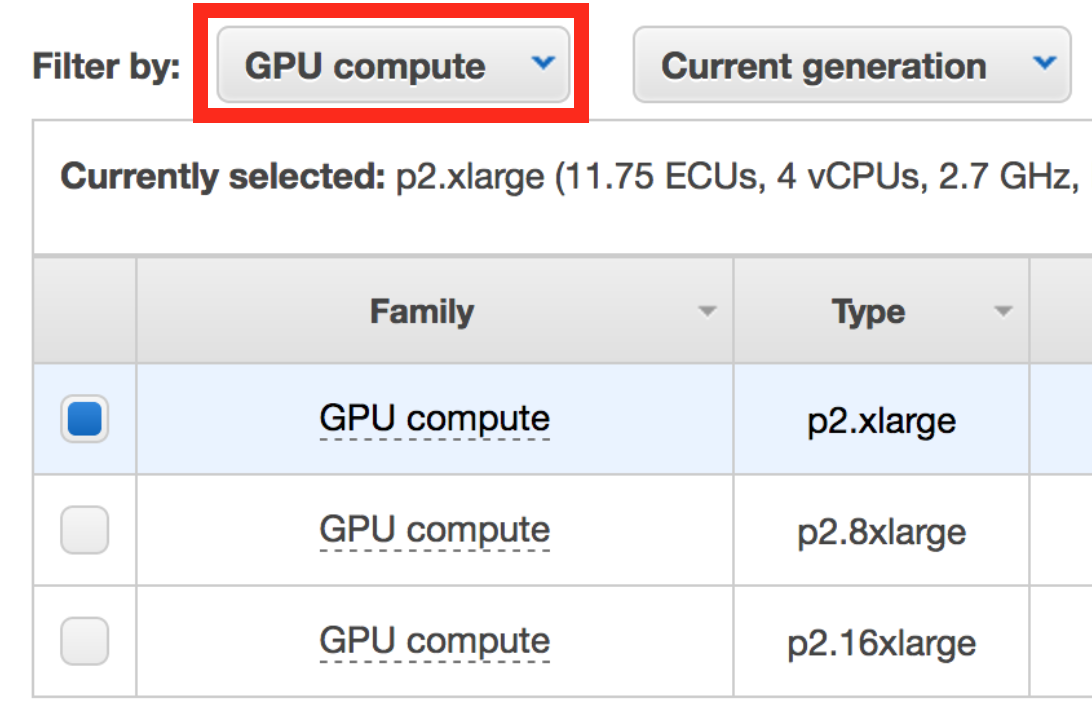


This [Amazon Machine Image (AMI)](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html) contains all the environment files and drivers for you to train on a GPU. It has [cuDNN](https://developer.nvidia.com/cudnn), and many the other packages required for this course. Any additional packages required for specific projects will be detailed in the appropriate project instructions.

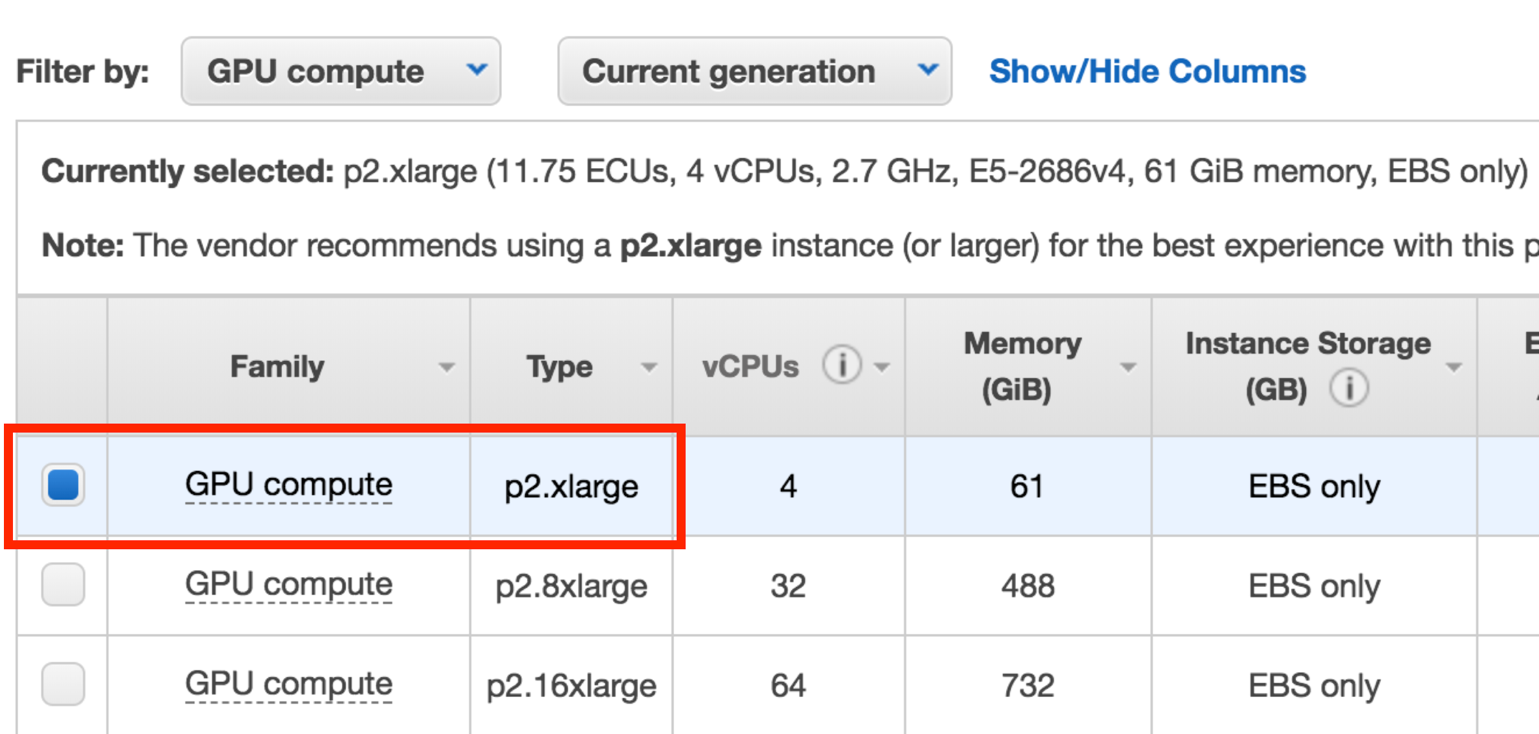
**Select the Instance Type**

You must next choose an instance type, which is the hardware on which the AMI will run.

Filter the instance list to only show “GPU compute”:



Select the p2.xlarge instance type:



Finally, click on the "Review and Launch" button:



**Configure the Security Group**

Running and accessing a Jupyter notebook from AWS requires special configurations.

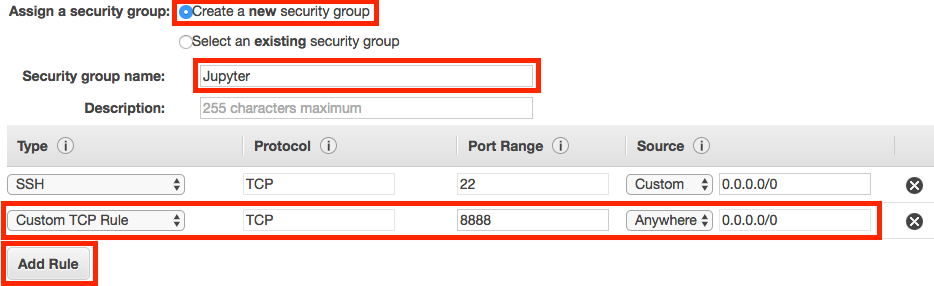
By default, AWS restricts access to most ports on an EC2 instance. In order to access the Jupyter notebook, you must configure the AWS Security Group to allow access to port 8888.

Click on "Edit security groups".



On the "Configure Security Group" page:

* Select "Create a **new** security group"
* Set the "Security group name" (i.e. "Jupyter")
* Click "Add Rule"
* Set a "Custom TCP Rule"
  1. Set the "Port Range" to "8888"
  2. Select "Anywhere" as the "Source"
* Click "Review and Launch" (again)



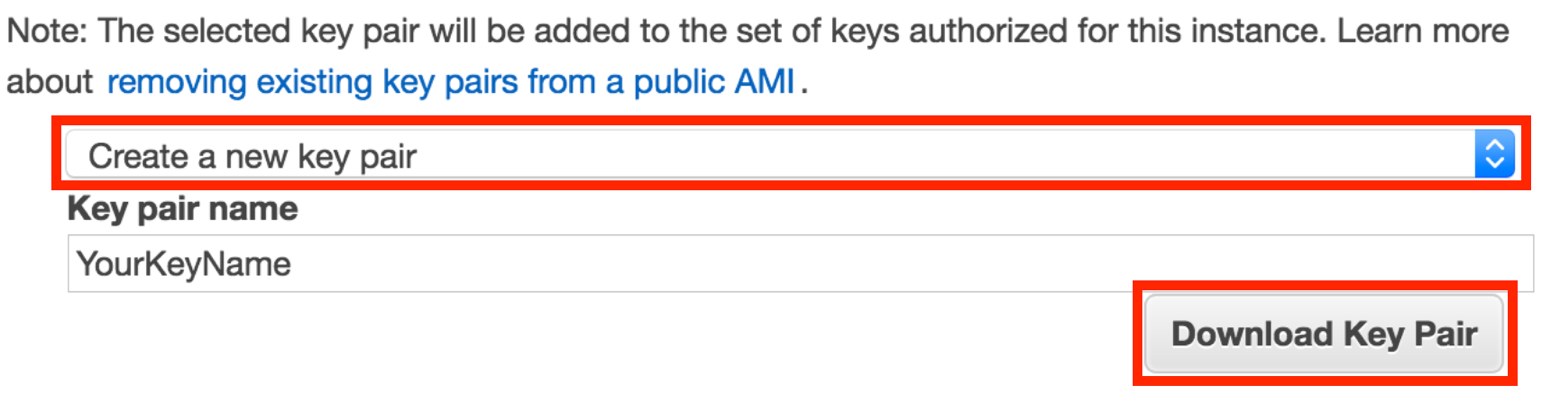
**Launch the Instance**

Click on the “Launch” button to launch your GPU instance.



**Create an Authentication Key Pair**

AWS will ask if you’d like to specify an authentication key pair. You'll need to do so in order to access your instance, so you select “Create a new key pair” and click the "Download Key Pair" button. This will download a .pem file, which you'll need to be able to access your instance.



Move the .pem file to a secure and easily remembered location on your computer; you'll need to access your instance through the location you select.

After the .pem file has been downloaded, click the “Launch Instances” button.

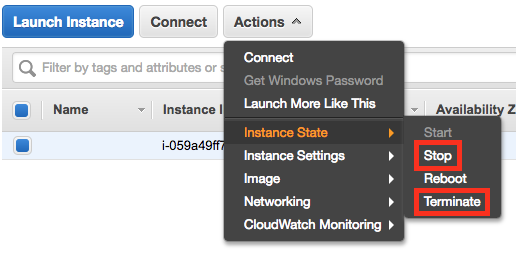
Click the “View Instances” button to go to the EC2 Management Console and watch your instance boot.

**Be Careful!**

From this point on, AWS will charge you for running this EC2 instance. You can find the details on the [EC2 On-Demand Pricing page](https://aws.amazon.com/ec2/pricing/on-demand/).

Most importantly, remember to “stop” (i.e. shutdown) your instances when you are not using them. Otherwise, your instances might run for a day, week, month, or longer without you remembering, and you’ll wind up with a large bill!

AWS charges primarily for running instances, so most of the charges will cease once you stop the instance. However, there are smaller storage charges that continue to accrue until you “terminate” (i.e. delete) the instance.



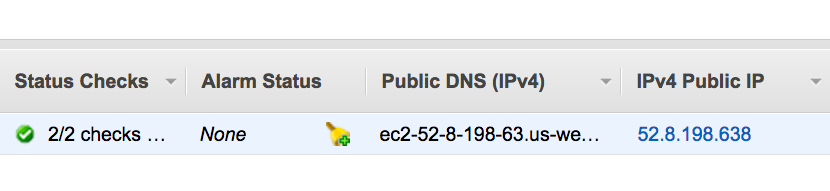
There is no way to limit AWS to only a certain budget and have it auto-shutdown when it hits that threshold. However, you can set [AWS Billing Alarms](http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/free-tier-alarms.html).

## Login to the Instance

**Login to the Instance**

After launch, your instance may take a few minutes to initialize.

Once you see “2/2 checks passed” on the EC2 Management Console, your instance is ready for you to log in.



Instance Status Check and Public IP

Note the "IPv4 Public IP" address (in the format of “X.X.X.X”) on the EC2 Dashboard.

From a terminal, navigate to the location where you stored your .pem file. (For example, if you put your .pem file on your Desktop, cd ~/Desktop/ will move you to the correct directory.)

Type ssh -i YourKeyName.pem ubuntu@X.X.X.X, where:

* X.X.X.X is the IPv4 Public IP found in AWS, and
* YourKeyName.pem is the name of your .pem file.

**Configure Jupyter notebook settings**

In your instance, in order to create a config file for your Jupyter notebook settings, type: jupyter notebook --generate-config.

Then, to change the IP address config setting for notebooks (this is just a fancy one-line command to perform an exact string match replacement; you could do the same thing manually using vi/vim/nano/etc.), type: sed -ie "s/#c.NotebookApp.ip = 'localhost'/#c.NotebookApp.ip = '\*'/g" ~/.jupyter/jupyter\_notebook\_config.py

**Test the Instance**

Make sure everything is working properly by verifying that the instance can run a Keras notebook.

**On the EC2 instance**

* Clone a GitHub repository
  + git clone https://github.com/udacity/aind2-cnn.git
* Enter the repo directory
  + cd aind2-cnn
* Install the requirements
  + sudo python3 -m pip install -r requirements/requirements-gpu.txt
* Start Jupyter notebook
  + jupyter notebook --ip=0.0.0.0 --no-browser

**From your local machine**

* You will need the token generated by your jupyter notebook to access it. On your instance terminal, there will be the following line: Copy/paste this URL into your browser when you connect for the first time, to login with a token:. Copy everything starting with the :8888/?token=.
* Access the Jupyter notebook index from your web browser by visiting: X.X.X.X:8888/?token=... (where X.X.X.X is the IP address of your EC2 instance and everything starting with :8888/?token= is what you just copied).
* Click on "mnist\_mlp" to enter the folder, and select the "mnist\_mlp.ipynb" notebook.
* Run each cell in the notebook.

**NOTE:** Windows users may prefer connecting via the GUI utility PuTTY, by following [these instructions](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html?icmpid=docs_ec2_console).

From this point on, AWS will charge you for a running an EC2 instance. You can find the details on the [EC2 On-Demand Pricing page](https://aws.amazon.com/ec2/pricing/on-demand/).

Most importantly, remember to “stop” (i.e. shutdown) your instances when you are not using them. Otherwise, your instances might run for a day or a week or a month without you remembering, and you’ll wind up with a large bill!

AWS charges primarily for running instances, so most of the charges will cease once you stop the instance. However, there are smaller storage charges that continue to accrue until you “terminate” (i.e. delete) the instance.