

An NVIDIA GPU.

Depending on the size of your training set and the speed of your CPU, you might be able to train your neural network on your local CPU. Training could take anywhere from 15 minutes to several hours if you train for many epochs.

A faster alternative is to train on a GPU.

It's possible to purchase your own NVIDIA GPU, or you may have one built into your machine already.

If not, it’s easy ([**although not free**](https://aws.amazon.com/ec2/pricing/on-demand/)) to access a GPU-enabled server (also known as an "instance") through Amazon Web Services.

**1. 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.

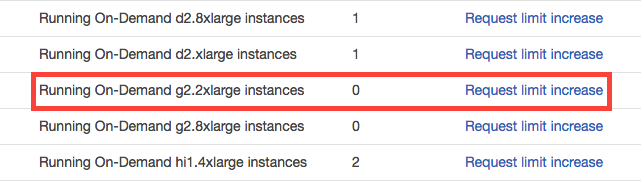
Furthermore, when you sign up, you will also need to choose a support plan. You can choose the free Basic Support Plan.

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.

**2. View Your Limit**

View your EC2 Service Limit report at: [**https://console.aws.amazon.com/ec2/v2/home?#Limits**](https://console.aws.amazon.com/ec2/v2/home?#Limits)

Find your "Current Limit" for the g2.2xlarge instance type.



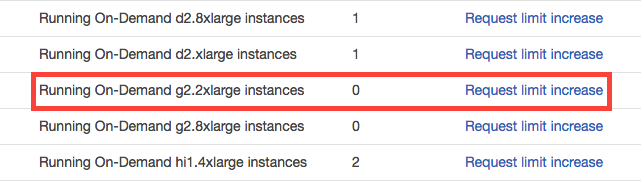
Amazon Web Services has a service called [**Elastic Compute Cloud (EC2)**](https://aws.amazon.com/ec2), which allows you to launch virtual servers (or “instances”), including instances with attached GPUs. The specific type of GPU instance you should launch for this tutorial is called “g2.2xlarge”.

By default, however, AWS sets a limit of 0 on the number of g2.2xlarge instances a user can run, which effectively prevents you from launching this instance.

**3. Submit a Limit Increase Request**

From the EC2 Service Limits page, click on “Request limit increase” next to “g2.2xlarge”.

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

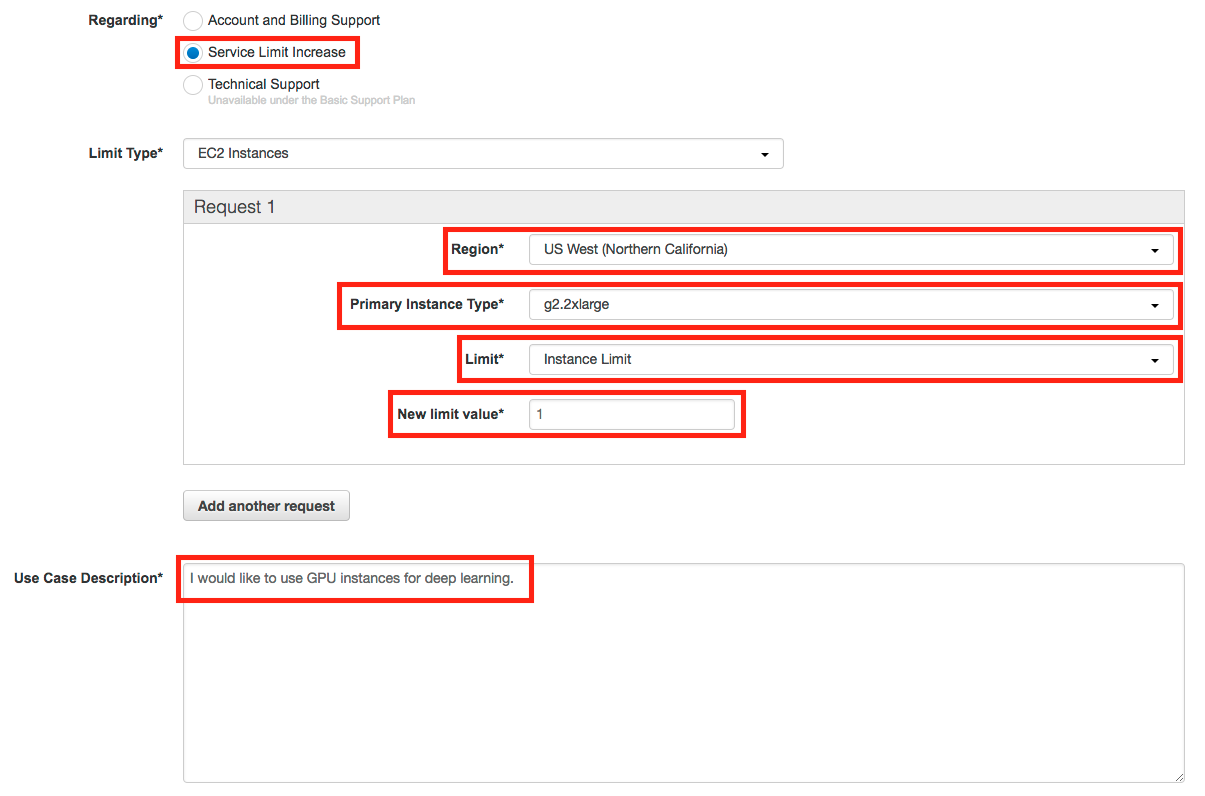


On the service request form, you will need to complete several fields.

For the “Region” field, select the region closest to you.

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.

**4. Wait for Approval**

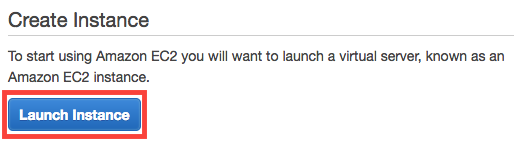
You must wait until AWS approves your Limit Increase Request. AWS typically approves these requests within 48 hours.

**5. Launch an Instance**

Once AWS approves your 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**](https://console.aws.amazon.com/ec2/v2/home)

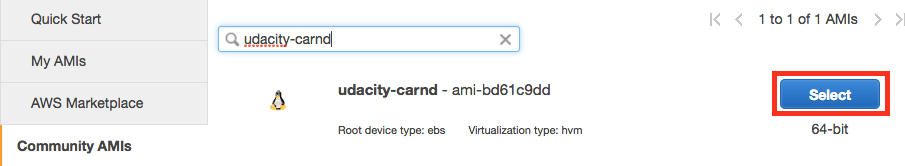
Click on the “Launch Instance” button.



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

We’ve created an AMI for you!

Search for the “udacity-carnd” AMI.

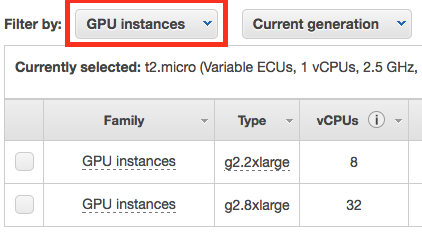


Click on the “Select” button.

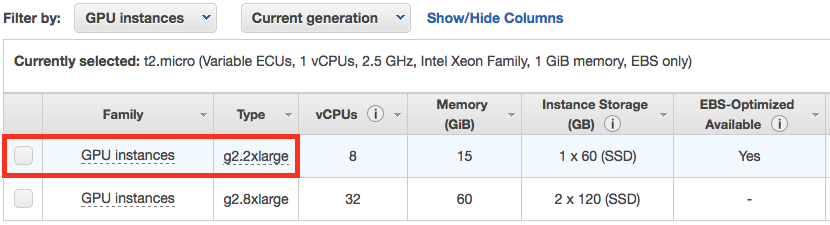
**6. Select the Instance Type**

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 instances”:



Select the g2.2xlarge instance type:



Finally, click on the “Review and Launch” button:



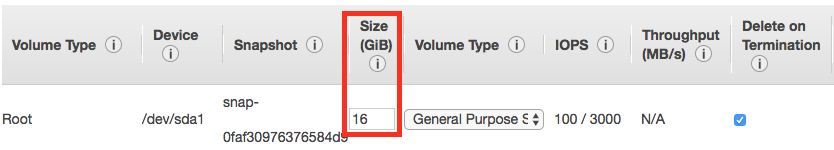
**7. Add Storage**

Your instance is now configured and ready for launch, but it would probably help to add storage above and beyond the 8GB that come with the g2.2xlarge instance by default.

Click on “Edit Storage”:



Increase the storage size to 16 GB (or more, if necessary):



Click on the “Review and Launch” button again.

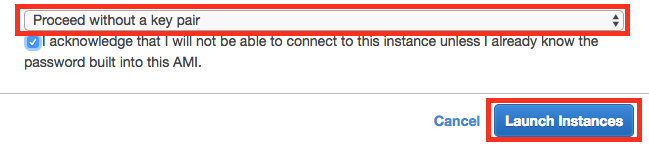
**8. Launch the Instance**

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



**9. Proceed Without a Key Pair**

Oops. Before you can launch, AWS will ask if you’d like to specify an authentication key pair.



In this case the AMI has a pre-configured user account and password, so you can select “Proceed without a key pair” and click the “Launch Instances” button (for real this time!).

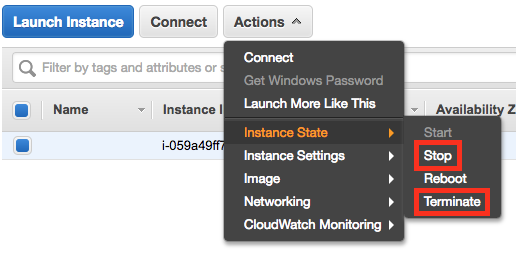
Next, click the “View Instances” button to go to the EC2 Management Console and watch your instance boot.

**10. Be Careful!**

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.

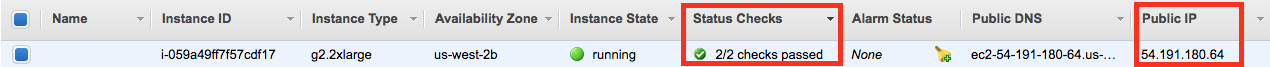


There is no way to limit AWS to only a certain budget and have it auto-shutdown when it hits that threshold. However, you can set [**AWS Billing Alarms**](http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/free-tier-alarms.html).

**11. Log In**

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.



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

From a terminal, SSH to that address as user “carnd”:

ssh carnd@X.X.X.X

Authenticate with the password: carnd

**12. Test TensorFlow**

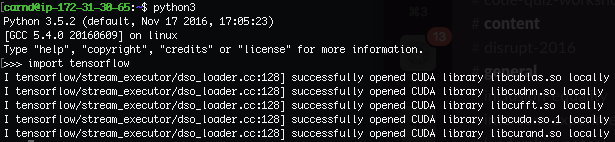
Congratulations! You now have a GPU-enabled server on which to train your neural networks.

Make sure everything is working properly by testing the TensorFlow installation.

Run python3 (**not** python) at the terminal prompt.

Then type import tensorflow from within the Python interpreter.

You should see TensorFlow import successfully, and include several CUDA libraries, which are necessary for the GPU.



**13. Train a Network**

Finally! It’s time to see how fast your GPU instance can train a network.

Exit the Python interpreter by running the exit() command.

Next, download the CNN you studied in the [**“Convolutional Network in TensorFlow”**](https://classroom.udacity.com/nanodegrees/nd013/parts/fbf77062-5703-404e-b60c-95b78b2f3f9e/modules/6df7ae49-c61c-4bb2-a23e-6527e69209ec/lessons/601ae704-1035-4287-8b11-e2c2716217ad/concepts/afe0660b-a035-499b-9441-737d601e19df) segment by running:

wget https://gist.githubusercontent.com/dsilver829/e7345a4f221b3aec5f17c02afb0af9a3/raw/8b1adbe796a735c76d87470f1f38b1b3ca45be56/cnn.py

Then train the network by running:

python3 cnn.py

This network took 1.5 minutes to train for 5 epochs on my GPU instance. The same parameters took 25 minutes on my local CPU!

