

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.

https://d17h27t6h515a5.cloudfront.net/topher/2016/December/58463173_create-an-aws-account/create-an-aws-account.png

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.

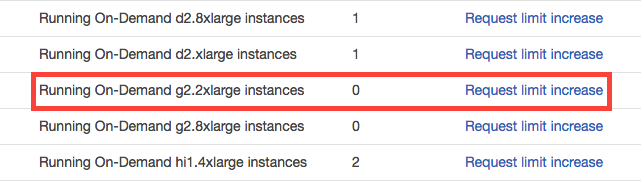
Once you finish signing up, wait a few minutes to receive your AWS account confirmation email. Then return to [aws.amazon.com](https://aws.amazon.com) and sign in.

**2. View Your Limit**

View your EC2 Service Limit report at: <https://console.aws.amazon.com/ec2/v2/home?#Limits>

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

Note: Not every AWS region supports GPU instances. If the region you've chosen does not support GPU instances, but you would like to use a GPU instance, then change your AWS region.



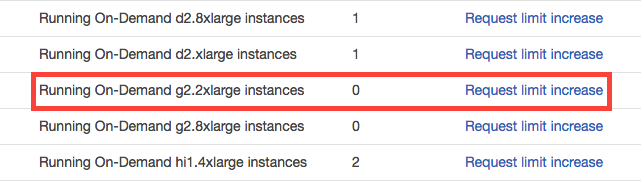
Amazon Web Services has a service called [Elastic Compute Cloud (EC2)](https://aws.amazon.com/ec2), which allows you to launch virtual servers (or “instances”), including instances with attached GPUs. The specific type of GPU instance you should launch for this tutorial is called “g2.2xlarge”.

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.

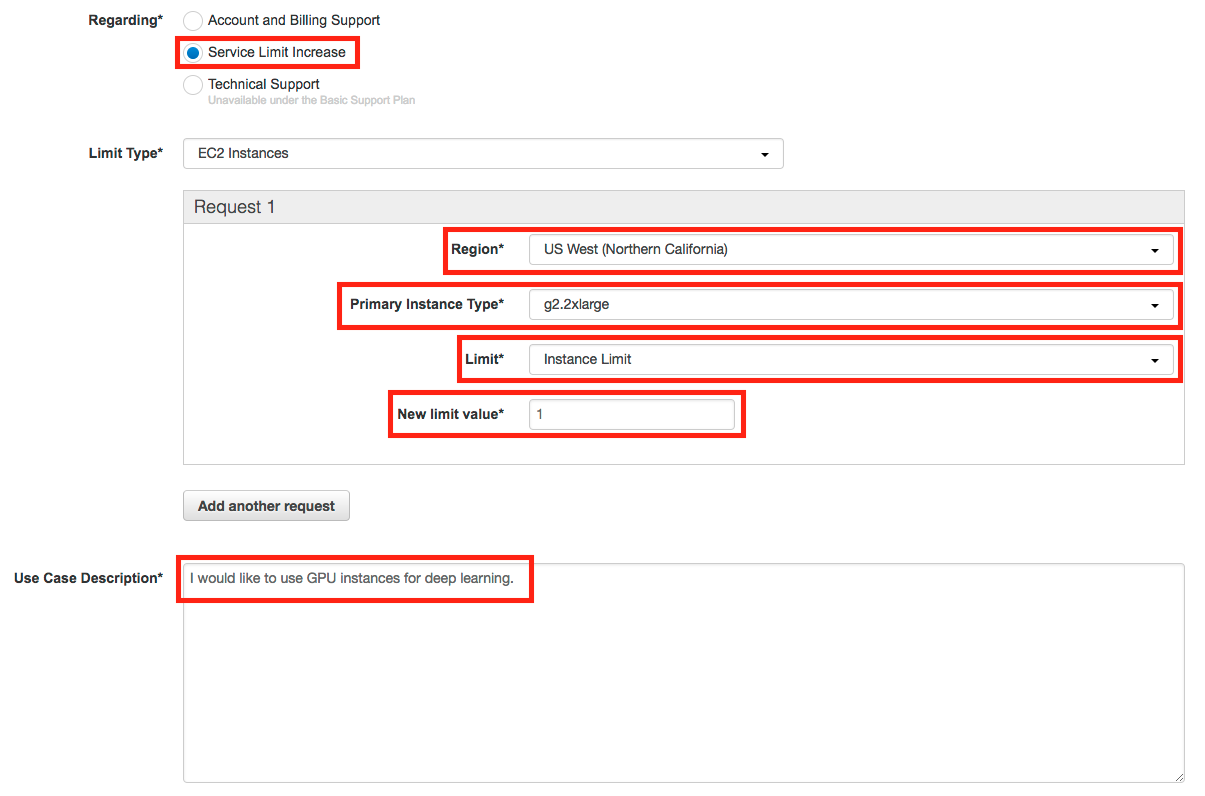


On the service request form, you will need to complete several fields.

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**

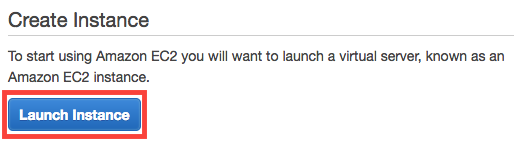
You must wait until AWS approves your Limit Increase Request. AWS typically approves these requests within 48 hours.

**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>

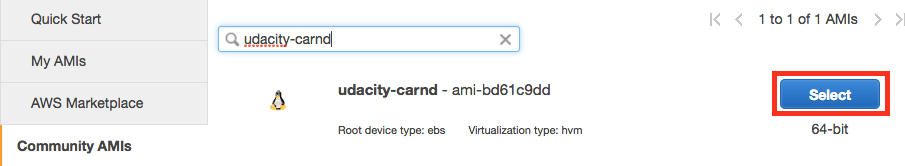
Click on the “Launch Instance” button.



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.

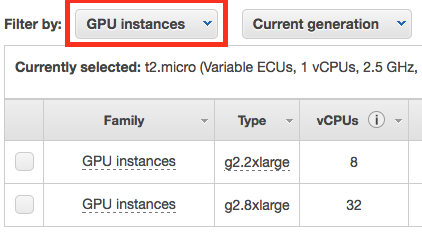


Click on the “Select” button.

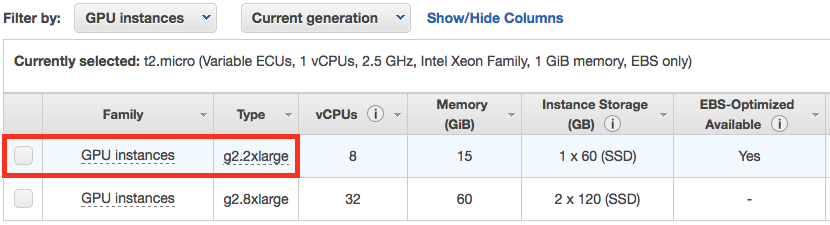
**6. Select the Instance Type**

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:

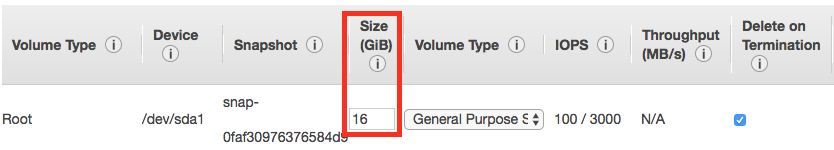


Finally, click on the “Review and Launch” button:

https://d17h27t6h515a5.cloudfront.net/topher/2016/December/584633cb_review-and-launch/review-and-launch.png

https://d17h27t6h515a5.cloudfront.net/topher/2016/December/584633f6_edit-storage/edit-storage.png

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



Click on the “Review and Launch” button again.

**7. Configure the Security Group**

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

Most of these configurations are already set up on the udacity-carnd AMI. However, you must also configure the security group correctly when you launch the instance.

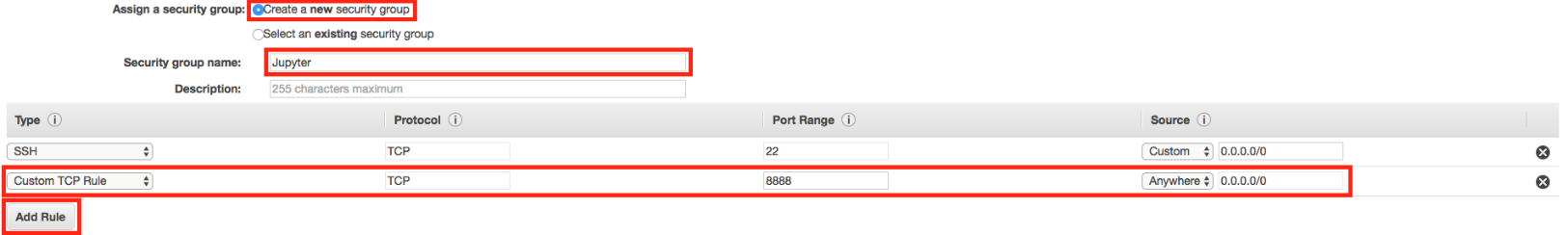
By default, AWS restricts access to most ports on an EC2 instance. In order to access the Jupyter notebook, you must configure the AWS Security Group to allow access to port 8888.

Click on "Edit security groups".

https://d17h27t6h515a5.cloudfront.net/topher/2016/December/58543293_edit-security-group/edit-security-group.png

On the "Configure Security Group" page:

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



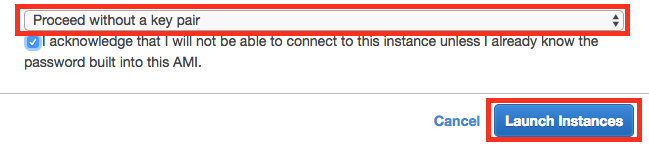
**8. Launch the Instance**

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

https://d17h27t6h515a5.cloudfront.net/topher/2016/December/58463444_launch/launch.png

**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!).

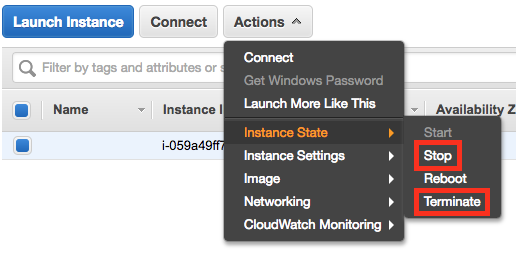
Next, click the “View Instances” button to go to the EC2 Management Console and watch your instance boot.

**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.

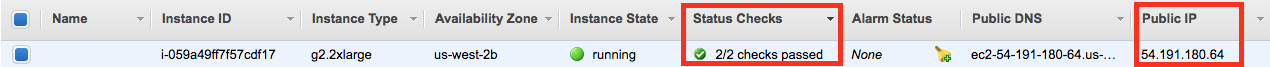


There is no way to limit AWS to only a certain budget and have it auto-shutdown when it hits that threshold. However, you can set [AWS Billing Alarms](http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/free-tier-alarms.html).

**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

If login issues regarding setting up a private key are encountered:

* make sure step 9 has been followed
* try switching clients, many students have been successful with git-bash and putty

**12. Launch a Jupyter Notebook**

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

Make sure everything is working properly by verifying that the instance can run the [LeNet-5 lab solution](https://github.com/udacity/CarND-LeNet-Lab/blob/master/LeNet-Lab-Solution.ipynb).

On the EC2 instance:

1. Clone the LeNet Lab repo: git clone https://github.com/udacity/CarND-LeNet-Lab.git
2. Enter the repo directory: cd CarND-LeNet-Lab
3. Activate the new environment: source activate carnd-term1
4. Run the notebook: jupyter notebook LeNet-Lab-Solution.ipynb

**Alternative Instructions**

The instruction for launching and connecting to Jupyter Notebook may not work for all users. If these instruction do not work for you, please try this (differences start at step 3):

1. Clone the LeNet Lab repo: git clone https://github.com/udacity/CarND-LeNet-Lab.git
2. Enter the repo directory: cd CarND-LeNet-Lab
3. Activate the new environment: source activate carnd-term1
4. Start Jupyter: jupyter notebook --ip=0.0.0.0 --no-browser
5. Look at the output in the window, and find the line that looks like the following:Copy/paste this URL into your browser when you connect for the first time to login with a token: http://0.0.0.0:8888/?token=3156e...
6. Copy and paste the complete URL into the address bar of a web browser (Firefox, Safari, Chrome, etc). Before navigating to the URL, replace 0.0.0.0 in the URL with the "IPv4 Public IP" address from the EC2 Dashboard. Press Enter.
7. Your browser should display a list of the folders in the repository. Select the target notebook and happy coding.

**13. Run the Jupyter Notebook**

From your local machine:

1. Access the Jupyter notebook index from your web browser by visiting: X.X.X.X:8888 (where X.X.X.X is the IP address of your EC2 instance)
2. Click on the "LeNet-Lab-Solution.ipynb" link to launch the LeNet Lab Solution notebook
3. Run each cell in the notebook

It took me 7.5 minutes to train LeNet-5 for ten epochs on my local CPU, but only 1 minute on an AWS GPU instance!

**Troubleshooting Missing Modules**

Some students have reported missing dependencies. These include tdqm and libgtk

* **tdqm** To install, execute conda install -c anaconda tqdm
* **libgtk** The command import cv2 may result in the following error. ImportError: libgtk-x11-2.0.so.0: cannot open shared object file: No such file or directory. To address make sure you are switched into the correct environment and try source activate carnd-term;conda install opencv. If that is unsuccessful please try apt-get update;apt-get install libgtk2.0-0 (may need to call as sudo) More information can be found [here](https://discussions.udacity.com/t/importerror-libgtk-x11-2-0-so-0/244471).