



An NVIDIA GPU.

In this concept, you will learn how to launch a GPU-enabled server on which to train your neural networks!

Earlier in this lesson, you set up an AWS account for deep learning by following Steps 1-4 below. If you haven't already, please complete these steps now.

1. Create an AWS Account

Visit 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.

AWS GPU Instances (Part 2)

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 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 p2.xlarge 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.

| | | |
|---|---|--|
| Running On-Demand p2.16xlarge instances | 0 | Request limit increase |
| Running On-Demand p2.8xlarge instances | 0 | Request limit increase |
| Running On-Demand p2.xlarge instances | 0 | Request limit increase |

Amazon Web Services has a service called [Elastic Compute Cloud \(EC2\)](#), 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".

By default, however, 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.

3. Submit a Limit Increase Request

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.

| | | |
|---|---|--|
| Running On-Demand p2.16xlarge instances | 0 | Request limit increase |
| Running On-Demand p2.8xlarge instances | 0 | Request limit increase |
| Running On-Demand p2.xlarge instances | 0 | Request limit increase |

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

For the "Region" field, select the region closest to you.

AWS GPU Instances (Part 2)

learning.”

Regarding* ☐ Account and Billing Support
☒ Service Limit Increase
☐ Technical Support
Unavailable under the Basic Support Plan

Limit Type* EC2 Instances

Request 1

Region* US West (Northern California)

Primary Instance Type* p2.xlarge

Limit* Instance Limit

New limit value* 1

Add another request

Use Case Description* 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.

AWS GPU Instances (Part 2)

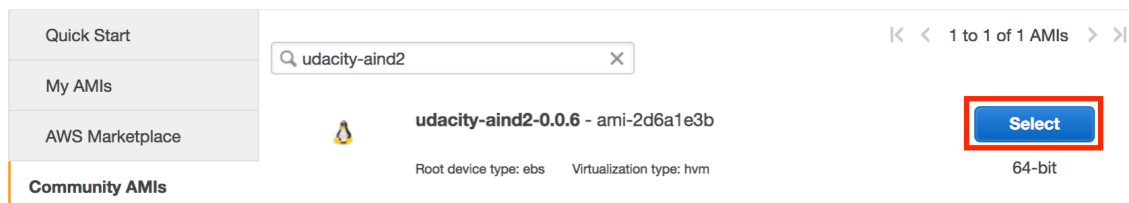
To start using Amazon EC2, you will want to launch a virtual server, known as an Amazon EC2 instance.

Launch Instance

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!

Click on "Community AMIs", and search for the "udacity-aind2" 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 compute":

AWS GPU Instances (Part 2)

Currently selected: p2.xlarge (11.75 ECUs, 4 vCPUs, 2.7 GHz,

| | Family | Type | |
|-------------------------------------|-------------|-------------|--|
| <input checked="" type="checkbox"/> | GPU compute | p2.xlarge | |
| <input type="checkbox"/> | GPU compute | p2.8xlarge | |
| <input type="checkbox"/> | GPU compute | p2.16xlarge | |

Select the p2.xlarge instance type:

Filter by: GPU compute Current generation Show/Hide Columns

Currently selected: p2.xlarge (11.75 ECUs, 4 vCPUs, 2.7 GHz, E5-2686v4, 61 GiB memory, EBS only)

| | Family | Type | vCPUs | Memory (GiB) | Instance Storage (GB) |
|-------------------------------------|-------------|-------------|-------|--------------|-----------------------|
| <input checked="" type="checkbox"/> | GPU compute | p2.xlarge | 4 | 61 | EBS only |
| <input type="checkbox"/> | GPU compute | p2.8xlarge | 32 | 488 | EBS only |
| <input type="checkbox"/> | GPU compute | p2.16xlarge | 64 | 732 | EBS only |

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



7. Configure the Security Group

Your instance is now configured and ready for launch, but running and accessing a Jupyter notebook from AWS requires special configurations.

Most of these configurations are already set up on the **udacity-ai** 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.

AWS GPU Instances (Part 2)

▼ Security Groups Edit security groups

Security group name: launch-wizard-9
Description: launch-wizard-9 created 2017-05-09T13:14:17.191-04:00

| Type ⓘ | Protocol ⓘ | Port Range ⓘ | Source ⓘ |
|----------------------------------|------------|--------------|----------|
| This security group has no rules | | | |

On the "Configure Security Group" page:

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

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name: Jupyter
Description: Jupyter

| Type ⓘ | Protocol ⓘ | Port Range ⓘ | Source ⓘ |
|------------|------------|--------------|--------------------------|
| SSH | TCP | 22 | Custom 0.0.0.0/0 |
| Custom TCP | TCP | 8888 | Anywhere 0.0.0.0/0, ::/0 |

Add Rule

8. Launch the Instance

Click on the "Launch" button to launch your GPU instance!

[Cancel](#) [Previous](#) [Launch](#)

9. Proceed Without a Key Pair

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

AWS GPU Instances (Part 2)

Cancel

Launch Instances

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!).

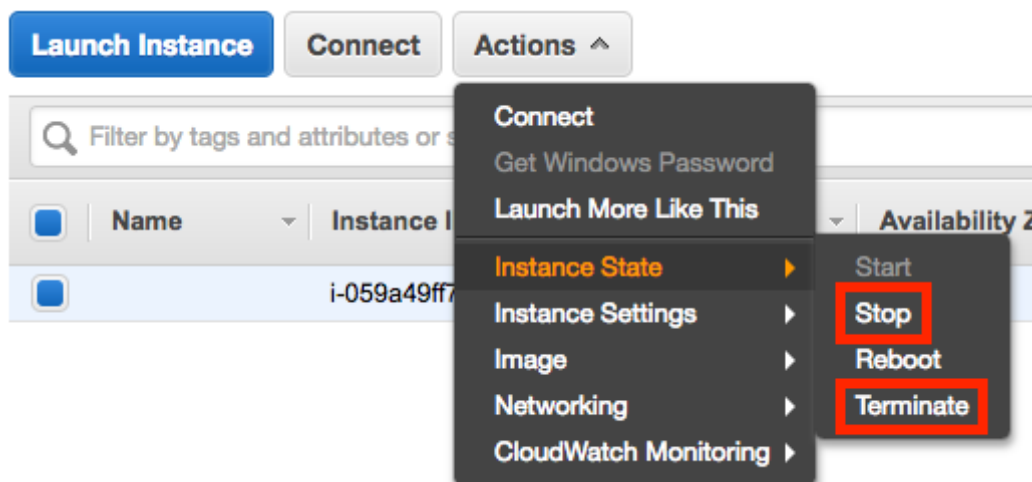
Next, click the “View Instances” button to go to the EC2 Management Console and watch your instance boot.

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](#).

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](#).

11. Log In

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

AWS GPU Instances (Part 2)

| Instance Type | Availability Zone | Instance State | Status Checks |
|---------------|-------------------|---|---|
| p2.xlarge | us-east-1c |  running |  2/2 checks passed |

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

From a terminal, SSH to that address as user "aind2":

```
ssh aind2@X.X.X.X
```

You might receive a message that asks "Are you sure you want to continue connecting (yes/no)?" Type **yes**, and hit Enter.

Authenticate with the password: **aind2**

12. Launch Jupyter

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

On the EC2 instance:

1. Clone the aind2-cnn repo: `git clone https://github.com/udacity/aind2-cnn.git`
2. Enter the repo directory: `cd aind2-cnn`
3. Activate the new environment: `source activate aind2`
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
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.

13. Run a Jupyter Notebook

Your browser should display a list of the folders in the [repository](#).

AWS GPU Instances (Part 2)

Select items to perform actions on them.

| <input type="checkbox"/> | <input type="text"/> | <input type="button" value="Name ↑"/> | <input type="button" value="Last Modified ↑"/> |
|--------------------------|--|---------------------------------------|--|
| <input type="checkbox"/> | 📁 cifar10-augmentation | | 13 minutes ago |
| <input type="checkbox"/> | 📁 cifar10-classification | | 13 minutes ago |
| <input type="checkbox"/> | 📁 conv-visualization | | 13 minutes ago |
| <input type="checkbox"/> | 📁 mnist-mlp | | 13 minutes ago |
| <input type="checkbox"/> | 📁 transfer-learning | | 13 minutes ago |
| <input type="checkbox"/> | 📄 README.md | | 13 minutes ago |
| <input type="checkbox"/> | 📄 requirements.txt | | 13 minutes ago |

You can now make sure everything is working properly by verifying that the instance can run a Jupyter notebook.

First click on **cifar10-classification**.

 jupyter

Logout

FilesRunningClusters

Select items to perform actions on them.

UploadNewRefresh

Name ↑Last Modified ↑

| | | | |
|--------------------------|-------------|------------------------|----------------|
| <input type="checkbox"/> | folder icon | cifar10-augmentation | 13 minutes ago |
| <input type="checkbox"/> | folder icon | cifar10-classification | 13 minutes ago |
| <input type="checkbox"/> | folder icon | conv-visualization | 13 minutes ago |
| <input type="checkbox"/> | folder icon | mnist-mlp | 13 minutes ago |
| <input type="checkbox"/> | folder icon | transfer-learning | 13 minutes ago |
| <input type="checkbox"/> | file icon | README.md | 13 minutes ago |
| <input type="checkbox"/> | file icon | requirements.txt | 13 minutes ago |

Then, click on **cifar10_mlp.ipynb**. Run all of the cells in the notebook to make sure everything is working properly.

NEXT