What is Amazon Web Services (AWS)

Cloud Based Computational Resource
What does AWS offer?

Many many things but here are the two main things to care about for DL...

EC2 - Compute Resources

Train the models

S3 - Data Storage

Store training data, models, etc
EC2 - What kinds of machines are available?

Different types and different subtypes (you can mix and match what you want)...

Here are the ones you may care about

<table>
<thead>
<tr>
<th>General Purpose:</th>
<th>Compute Optimized:</th>
<th>GPU Optimized:</th>
</tr>
</thead>
<tbody>
<tr>
<td>T2 - Webservices</td>
<td>C2 - Multiplayer Gaming Servers, scientific computing</td>
<td>P3/P2 - Machine Learning</td>
</tr>
<tr>
<td>M3/M4 - Databases, Fileservers, etc</td>
<td>C3/C4 - Ad serving machines, MMO servers, etc</td>
<td>G3 - Fluid dynamics, graphics rendering, etc</td>
</tr>
</tbody>
</table>

Machine sizes - nano, micro, medium, large, xlarge, 2xlarge, …, 16xlarge
EC2 - What kinds of machines are available?

Different types and different subtypes (you can mix and match what you want)...

- **T2.nano**
  - 1 vCPU (Xeon)
  - 0.5 GB RAM

- **M4.large**
  - 2 vCPU (Xeon)
  - 8 GB RAM

- **C4.8xlarge**
  - 36 vCPU (Xeon)
  - 60 GB RAM

- **P2.16xlarge**
  - 64 vCPU (Xeon)
  - 16 GPU (Nvidia K80)
  - 192 GB GPU-Memory
  - 732 GB RAM
EC2 - So what do we put on these machines?

Amazon Machine Instances (AMIs)

● Virtual images of existing machines
  ○ You can create an image of your machine
    ■ Transfer it to a different machine
    ■ Save it as a backup

● Use cases
  ○ Software packages that are incredibly difficult to install
  ○ Need to create multiple different machines with the exact same data for parameters servers
  ○ Load balancing - create a new machine with the same AMI to be used in a different region depending on load
Now you know what AWS is and what you can do with it
Let’s set up a basic machine

Create An Account:

https://aws.amazon.com

Click “Sign Up” in the top right and follow the instructions

(If you already have an account you can skip this step)
Let’s set up a basic machine
Let’s set up a basic machine

Click on “Services” in the top left

Then, under “Compute”, select “EC2”
Let’s set up a basic machine

Then Click on “Running Instances”

Make sure you are in the “US West (Oregon)” region
Let’s set up a basic machine

Here you see your current instances

(I have 1 that is stopped, you shouldn’t have any)

Here are the details of that instance
Let’s set up a basic machine

Click “Launch Instance”
Let’s set up a basic machine

ami-f1e73689
Let's set up a basic machine

Select the t2-micro because it is “free tier eligible”

Select Next
Let’s set up a basic machine

Just select next
Let’s set up a basic machine

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

<table>
<thead>
<tr>
<th>Volume Type</th>
<th>Device</th>
<th>Snapshot</th>
<th>Size (GiB)</th>
<th>Volume Type</th>
<th>IOPS</th>
<th>Throughput (MiB/s)</th>
<th>Delete on Termination</th>
<th>Encrypted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root</td>
<td>/dev/sda1</td>
<td>snap-0b0e16d57074d26d</td>
<td>8</td>
<td>General Purpose SSD (GP2)</td>
<td>100 / 3000</td>
<td>N/A</td>
<td></td>
<td>Not Encrypted</td>
</tr>
</tbody>
</table>

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

Make sure you choose 8 GB of SSD Storage Space

Select next
Let’s set up a basic machine

Step 5: Add Tags
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>Instances</th>
<th>Volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(127 characters maximum)</td>
<td>(255 characters maximum)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This resource currently has no tags

Choose the Add tag button or click to add a Name tag.
Make sure your IAM policy includes permissions to create tags.

Add Tag  (Up to 50 tags maximum)
Let’s set up a basic machine

Make sure you have an SSH rule set (This should be default) & a TCP rule for 8888 so you can connect your Ipython Notebook.

You can set HTTP or other rules here too if you want.

Select Review and Launch.
Let’s set up a basic machine

Step 7: Review Instance Launch
Please review your instance launch details. You can go back to edit changes for each section. Click Launch to assign a key pair to your instance and complete the launch process.

Take one last look to make sure you are happy with everything...

Select Launch
Let’s set up a basic machine

This window allows you to create a private key to access your machine… when you ssh you will need it.

Download your key and save it.

Make sure it’s in a place where it won’t get lost - this key is the only way to connect to this specific instance!
Now you have a basic AWS machine up and running
Now let’s connect to it

Your instance is now identified by its IP address. You can ssh to your instance by using this public IP address and your address key.

Let’s start by changing the instance type. For student accounts, t2.micro is free-tier eligible, but let’s still start on a t2.nano instance.

Remember your instance’s IP address changes every time you restart it.
Set instance type

Right-click on your instance under the instance tab, go to instance settings and change instance type to t2.nano
Set instance type

Right-click on your instance under the instance tab, go to instance settings and change instance type to t2.nano
Launch instance

Right-click and Start your instance.
Launch instance

Right-click and Start your instance.

This will start the process of allocating resources to your instance. Once this is completed, your instance will be running and you can connect to it.

Stopping this instance removes the compute associated with the current session.

Do NOT terminate! This will wipe the slate clean. Terminate ONLY when you are sure of ‘throwing away’ the data.
Connect to launched instance

Right-click and click connect

The ssh command shown includes the currently assigned IP address. This will change at each launch!

Usually we won’t have root privileges - use ubuntu@IP instead. (Depending on the AMI you would use “root”, “ec2-user”, etc)

Note - the ssh command will use the path to the key associated with this instance.
Now you have a running AWS machine and you can connect
Now let’s set up a Jupyter Notebook
Now let’s set up a Jupyter Notebook
Now let’s set up a Jupyter Notebook

```bash
mkdir notebooks
cd notebooks/
```
Now let’s set up a Jupyter Notebook.
Now let’s set up a Jupyter Notebook
Now let’s set up a Jupyter Notebook

https://pastebin.com/artPNii0
Now let’s set up a Jupyter Notebook

```
ubuntu@ip-172-31-20-189:~/.notebooks$ jupyter notebook
[I 18:45:08.932 NotebookApp] [nb_conda_kernels] enabled, 15 kernels found
[W 18:45:09.345 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[I 18:45:09.467 NotebookApp] [nb_anoncondacloud] enabled
[I 18:45:09.470 NotebookApp] [nb_conda] enabled
[I 18:45:09.506 NotebookApp] nbpresent HTML export ENABLED
[W 18:45:09.506 NotebookApp] nbpresent PDF export DISABLED: No module named 'nbbrowserpdf'
[I 18:45:09.723 NotebookApp] sparkmagic extension enabled!
[I 18:45:09.725 NotebookApp] Serving notebooks from local directory: /home/ubuntu/notebooks
[I 18:45:09.725 NotebookApp] 0 active kernels
[I 18:45:09.725 NotebookApp] The Jupyter Notebook is running at: http://[all ip addresses on your system]:8888/
[I 18:45:09.726 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```
Now let’s set up a Jupyter Notebook
Now you have a running AWS machine and you can connect and you can test code using Jupyter Notebooks!
Now let's shut it down so you don't get billed

Finish and save your progress, exit from session

From ec2-console on your browser, stop the instance

You can just start the instance and resume whenever you want.
Now you know how to use machines on EC2
Now lets redeem some AWS credits
Now lets redeem some AWS credits
Now lets redeem some AWS credits
Now you know how to use machines on EC2 and you can afford to use the expensive ones.
Parting thoughts

● Remember to shut down your machines
  ○ Just because you don’t have an open ssh connection doesn’t mean your machine is off
  ○ You will still get billed, and you will run out of credits

● Check to see if the code you need exists as an AMI
  ○ Don’t waste time trying to build crazy versions of libraries, or installing some obscure runtime
  ○ You can probably find what you need in an existing AMI
Parting thoughts

● If you are working in a group, consider making a group AWS account
  ○ You can share credits on the account
  ○ You don't have to wait for someone to turn on the machine if you want to work
  ○ Everyone learns how to use AWS

● You need permission to launch a GPU instance
  ○ You will need to create a support ticket to launch GPU Machines
  ○ If you attempt to launch on you will be guided through the process
  ○ You are typically only allowed to launch 1 GPU machine at a time

● Only launch an expensive instance when it is time to train, not develop
  ○ Launch a basic instance with everything you need to develop and test your code
  ○ When it comes time to train your system for real, then launch a decked out instance
  ○ Saves you money