

Amazon Web Service (AWS) Elastic Compute Cloud (EC2) ([video link](#))

An EC2 Instance is a virtual server for running applications on the Amazon Web Services infrastructure. AWS is a cloud computing platform, and an EC2 is a service that allows subscribers to run application programs in the computing environment.

This section of Recitation 0 will review the following;

- Creating and connecting an AWS account with the AWS Command Line Interface
- Getting started with a demo on an EC2 Instance

Part 1/2:

Creating and Connecting an AWS account with the AWS Command Line Interface

1. Set up an AWS account ([create account](#))
 - a. Provide your contact information
 - b. Provide your credit card information
 - i. FYI: Students are only charged in the case that they use up all of the coupons for payment provided through the class.
 - c. Choose the Free / Basic account
2. Connect your AWS account with the Command Line Interface
 - a. Install CLI [version 2](#) for either Linux, macOS, or Windows ([formal instructions](#))
 - i. Confirm by the CLI version installed

```
$ which aws
/usr/local/bin/aws
$ aws --version
aws-cli/2.0.23 Python/3.7.4 Darwin/18.7.0 botocore/2.0.0
```

- b. Verify Authenticity **if download for linux** ([formal instructions](#))
 - i. Download and install the gpg command ([GnuPG website](#))
 - ii. Create a text file and paste the following to create the public key file:

-----BEGIN PGP PUBLIC KEY BLOCK-----

```
mQINBF2Cr7UBEADJZHcgusOJI7ENSyumXh85z0TRV0xJorM2B/JL0kHOyigQluUG
ZMLhENaG0bYatdrKP+3H91lvK050pXwnO/R7fB/FSTouki4cilx5OuLlnJZlxSzx
PqGI0mkxlmLNbGWoi6Lto0LYxqHN2iQtzlwTVmq9733zd3XfcXrZ3+LbIHAgEt5G
TfNxEKJ8soPLyWmwDH6HWCnjZ/alQRBtIQ05uVeEoYxSh6wOai7ss/KveoSNBbYz
gbdzoql2Y8cgH2nbfgp3DSasaLZEdCSsisK1u05CinE7k2qZ7KgKAUlcT/cR/grk
C6VwsnDU0OUCideXcQ8WeHutqvqZH1JgKDbznolzeQHJD238GEu+eKhRHcz8/jeG
94zkcqJoz3KbZGYMiTh277Fvj9zzvZsbMBCedV1BTg3Tqgv4X4bdkhf5cH+7NtWO
lrFj6UwAsGukBTA0xC0l/dnSmZhJ7Z1KmEWilro/gOrjtOxqRQutllqG22TaqoPG
fYVN+en3ZwbT97kcgZDwqbuykNt64oZWc4XKCa3mprEGC3lbJTBfqqIXmZ7l9yWg
EEUJYOib2XrSuPWml39beWdKM8kzr1OjniOm6+lpTRCBfo0wa9F8YZRhHPAkWkKX
XDeOGpWRj4ohOx0d2GWkyV5xyN14p2tQOCdOODmz80yUTgRpPVQUtOEhXQARAQAB
tCFBV1MgQ0xJIFRIYW0gPGF3cy1jbGIAyW1hem9uLmNvbT6JAIQEEwEIAD4WIQT7
Xbd/1cEYUaURraimMQrMRnJHXAUCXYKvtQlbAwUJB4TOAAULCQgHAgYVCgkICwIE
FglDAQleAQIXgAAKCRcmMQrMRnJHXXIXEACHLUIkg80uPUKgjE3jejvQSA1aWuAM
zyy6fdpdlRUz6M6nmsUhoExjVlvibEJpzK5mhuSZ4lb0vJ2ZUPgCv4zs2nBd7BGJ
MxKiWgBReGvTdqZ0SzyYH4PYCJSE732x/Fw9hfnh1dMTXNcrQXzwOmmFNNegG0Ox
```

```

au+VnpcR5Kz3smiTrlwZbRudo1ijhCYPQ7t5CMp9kjC6bObvy1hSlg2xNbMAN/Do
ikebAI36uA6Y/Uczjj3GxZW4ZWeFirMidKbtqvUz2y0UFszobjiBSqZZHCreC34B
hw9bFNpuWC/0SrXgohdsc6vK50pDGdV5kM2qo9tMQ/izsAwTh/d/GzZv8H4IV9eO
tEis+EpR497PaxKKh9tJf0N6Q1YLRHof5xePZtOIS3gfvH5hXA3HJ9yIxb8T0H
QYmVr3alUes20i6mel3fuV36VFupwfrTKaL7VXnsrK2fq5cRvyJLNzXucg0WAjPF
RrAGLzY7nP1xeg1a0aeP+pdsqqlIPJom8OCWc1+6DWbg0jsC74WoesAggBlitODMB
rsal1y/q+bPzpsnWjzHV8+1/EtZmSc8ZUGSJOPkFC7hObnfkl18h+1QtKTjZme4d
H17gsBJr+opwJw/Zio2LMjQBOqlm3K1A4zFT7wBC7He6KPQea1p2XAMgtvATtNe
YLZATHZKTJyiqA==
=vYOk
-----END PGP PUBLIC KEY BLOCK-----

```

- iii. Import the AWS CLI public key, substituting in the file name of the public key created in step ii.

```

$ gpg --import public-key-file-name
gpg: /home/username/.gnupg/trustdb.gpg: trustdb created
gpg: key A6310ACC4672475C: public key "AWS CLI Team <aws-cli@amazon.com>" imported
gpg: Total number processed: 1
gpg:             imported: 1

```

- iv. Download the AWS CLI signature for the package you downloaded, which has the extension “.sig”

```

$ curl -o awscliv2.sig https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip.sig

```

- v. Verify the signature by passing both the “.sig” and “.zip” file names as parameters

```

$ gpg --verify awscliv2.sig awscliv2.zip

```

- vi. Check to ensure that the output looks like this:

```

gpg: Signature made Mon Nov  4 19:00:01 2019 PST
gpg:             using RSA key FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC 4672 475C
gpg: Good signature from "AWS CLI Team <aws-cli@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:             There is no indication that the signature belongs to the owner.
Primary key fingerprint: FB5D B77F D5C1 18B8 0511 ADA8 A631 0ACC 4672 475C

```

- c. Configure with Admin Privileges ([formal instructions](#))
 - i. [Sign in to the IAM console](#) as the “root user” with your AWS account.
 - ii. Go to “My Account” and scroll down to “IAM User and Role Access to Billing Information”
 1. Click edit, check the box to “Activate IAM Access”, update, and return to the IAM dashboard.
 - iii. In the navigation pane, chose “Users”, and then “Add User”
 1. Name your user “Administrator”.
 2. Check the box to give “AWS Management Console access”.
 3. Select “Custom password”, type your new password in the text box, then click “Next: Permissions”.
 4. On the Permissions page, do the following:
 - a. Choose “Add user to group”.

- b. Choose "Create group".
 - c. In the Create group dialog box, for Group name type "Administrators".
 - d. Select the check box for the "AdministratorAccess" policy.
 - e. Choose "Create group".
 - f. Back on the page with the list of groups, select the check box for your new group. Choose Refresh if you don't see the new group in the list.
 - g. Choose "Next: Tags".
 5. Choose "Next: Review", choose "Create user", and then return to the IAM dashboard.
- iv. In the navigation pane, choose "Users"
 1. Choose the name of the user you just created, then choose the "Security credentials" tab.
 2. In the "Access keys" section, choose "Create access key".
 3. To download the key pair, choose "Download .csv" file, then choose "Close". Store the keys in a **secure location** (**You will not have access to the secret access key again after this dialog box closes**).
- v. Configure AWS CLI
 1. Open the terminal, and execute the following,

```
$ aws configure
```

2. As instructed in red, fill in the information

```
AWS Access Key ID [None]: copy from the download key file
AWS Secret Access Key [None]: copy from the download key file
Default region name [None]: us-west-2 (change it based on your
Location - for options, go to "Global" dropdown list in AWS page)
Default output format [None]: json
```

3. To view the active instances, run the following

```
$ aws ec2 describe-instances
```

Part 2/2:

Getting Started with a demo on an EC2 Instance

1. Launch an Instance
 - a. Go to Services, then choose EC2

- b. If not chosen, select your current region (if not there, the nearest to your location) from the “Global” tab. For instance, “US East (Ohio)” for those in Pittsburgh PA.
 - c. Select “launch instance”, type in the search box “deep learning AMI”, select “Deep Learning AMI (Ubuntu 18.04) Version 32.0”.
 - d. On the “Choose an Instance Type” page, select “t2.micro” (Non-GPU instance), choose “Next: Configure Instance Details”, then choose “Next: Add Storage”.
 - e. On the “Storage” page, keep or modify the predefined storage size, select “Next: Add Tag”, choose “Next: Configure Security Group”, then choose “Review and Launch”.
 - f. On the “Review Launch Instance” page, select “create a new key pair”, give it a name, download the “key pair” file (**Later, this key pair will be used to remotely access your instance from your machine**), then choose “launch instances” (The same key pair can be selected when launching future instances).
 - g. When launching successfully, choose “View Instance”, the instance status should be “pending” then changes to “running”.
2. Terminate and/or stop an Instance
 - a. On the “Launch Instance” page, choose “Actions”, then select “Instance State”.
 - b. To stop the instance, select “Stop”.
 - c. To shutdown the instance, select “Terminate” (all files stored in the instance will be irretrievable).
3. Switch from GPU to non-GPU
 - a. On the “Launch Instance” page, choose “Actions”, select “Instance Settings”, then choose “Change Instance Type”.
 - b. From the “Instance Type” dropdown list, select “g4dn.xlarge” for GPU instance, then choose apply.
 - c. Note: in order to use GPU instances, a permission request needs to be made.
4. Getting GPU Permissions
 - a. You Will Need Permission to get an Instance with a GPU.
 - i. Go to this [link](#).
 - ii. Click “Create Case”.
 - iii. Ensure “Service Limit Increase” is checked.
 - iv. Ensure “EC2 Instances” is selected under the “Case Classification” module.
 - v. In the “Requests” module, select the region, instance type and the new limit value that you’d like. Ask for at least 4 in the “New limit value” section.
 - vi. Under the case description, explain that you are taking a class in deep learning at a university and that you are requesting GPUs such that you can complete the homework assignments. You will usually be granted access, but if for some reason you are denied, please let us know, and you will need to open a new case.
5. Connecting and controlling an EC2 Instance
 - a. Start, Stop, or Terminate an instance using AWS CLI ([formal instructions](#))
 - i. To start an instance, execute the following command after replacing the italicized with your own values. (List of *image IDs*, which indicated next to

the image name, can be found when you go to EC2 dashboard and click on Launch Instance, Step 1), (List of *instance types* can be found when you go to EC2 dashboard and click on Launch Instance, Step 2), (*key-name* is the name of the key pair you created earlier), and (List of *security group IDs* can be found when you go to EC2 dashboard and click on security groups).

```
$ aws ec2 run-instances --image-id ami-xxxxxxx --count 1
--instance-type t2.micro --key-name MyKeyPair --security-group-ids
sg-903004f8
```

- ii. To stop and terminate an instance, execute the following commands respectively after replacing the italicized with your own values. (*Instance IDs* can be found by executing the describe command "**aws ec2 describe-instances**").

```
$ aws ec2 stop-instances --instance-ids i-5203422c
$ aws ec2 terminate-instances --instance-ids i-5203422c
```

- b. Accessing remotely the EC2 instance
 - i. To remotely access the created instance, execute the following ssh command. (The *key pair* file needs to be in the same directory when this ssh command is executed) and (*User* is usually **ubuntu** if the machine image is **ubuntu**, and *PublicDnsName* is the public DNS(IPv4) found by executing the describe command "**aws ec2 describe-instances**" or when you go to EC2 dashboard and click on running instances).

```
$ sudo ssh -i MyKeyPair.pem user@PublicDnsName
```

❖ Adding coupons to AWS

1. Go to AWS Homepage for your account.
2. Search 'Billing' in the search bar. This will take you to the '[Billing and Cost Management Dashboard page](#)'. This is where you can track your costs incurred.
3. In the left column, under 'Billing', go to 'Credits' or [link](#).

4. Click on 'Redeem credit' and copy+paste the coupon code provided to you under 'Promotion code'. Click 'Redeem credit'. This will apply the coupon to your account.

❖ **Tips for HW0 [AWS setup part]**

1. Set-up your development environment – with Conda, PyTorch, and CUDA.
2. Once you have activated the respective environment, type 'nvidia-smi' on the local terminal. The output should look similar to the expected output given in HW0P2 notebook.