COURSEWARE

AWS EC2

Contents

Overview

An EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure.

AWS is a comprehensive, evolving cloud computing platform; EC2 is a service that allows business subscribers to run application programs in the computing environment. The EC2 can serve as a practically unlimited set of virtual machines.

Instances are created from Amazon Machine Images (AMI).
The machine images are like templates that are configured with an operating system and other software, which determine the user's operating environment.
Users can select an AMI provided by AWS, the user community, or through the AWS Marketplace.
Users can also create their own AMIs and share them.

Number of Instances

You define the number of instances that you want to create based on the the AMI and tier you previously selected.

Network

You define the **Virtual Private Cloud** which you want your EC2 instance(s) to sit within. This could mean that you could create instances that are not available to the rest of the world.

Subnet

You can define the specific location that your EC2 instance(s) are going to sit in within the network.

Auto-assign IP Address

By default this should be set to **Use subnet setting (Enable)**, if not, then select this option. This will automatically assign an IP address for internet access. Without a public IP you wouldn't be able to reach the VM from your computer.

Disclaimer

AWS will occasionally update the Web Console UI, so the images shown will go out of date at some point. However,
the main functionality will still be there, so you will simply need to follow the instructions and not just rely
on the screenshots.

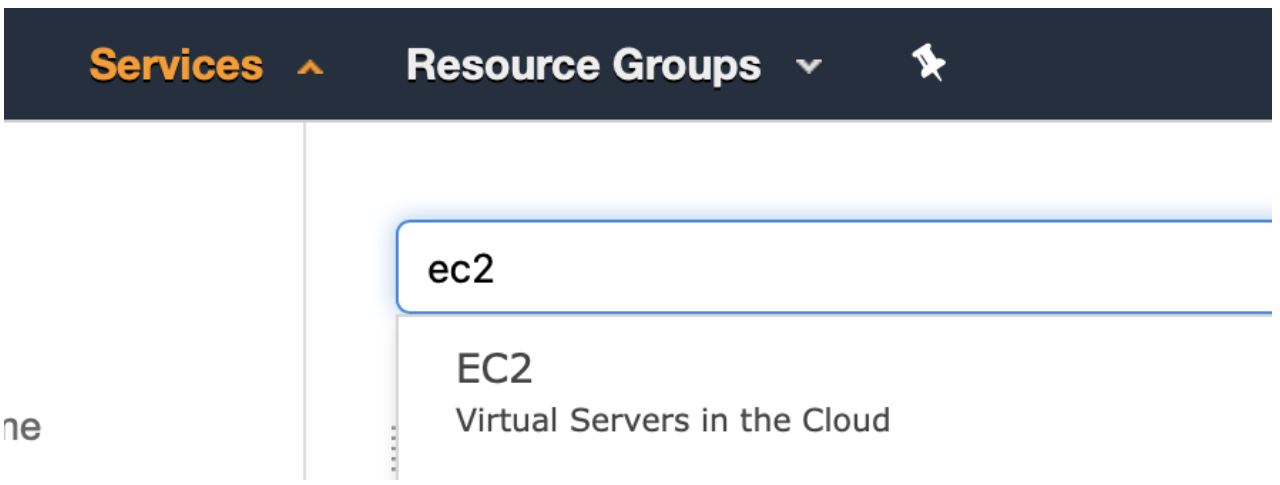
Tutorial

This part of the tutorial will go over how to deploy your EC2 instance using the AWS Web Console. The tutorial will also be taking place in the **eu-west-1** region of AWS, which is **Ireland**.

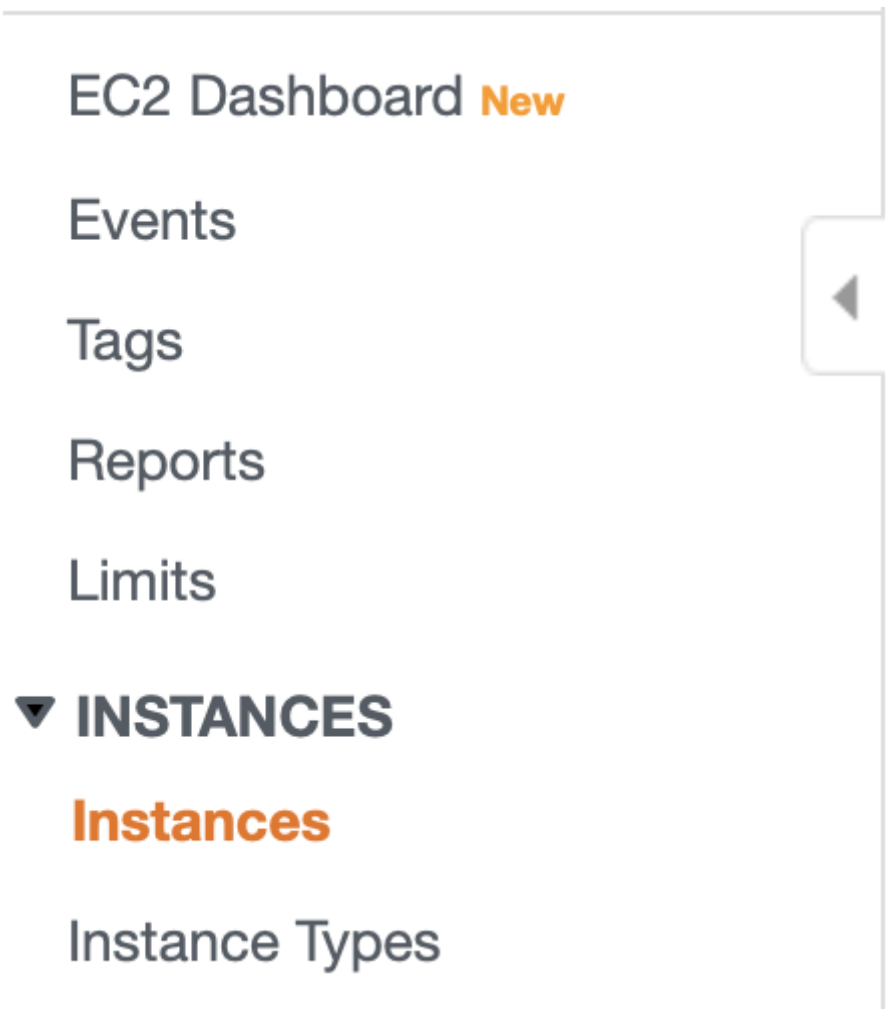
If you are running a *Windows* machine, then you can download GitBash here

AWS Web Console

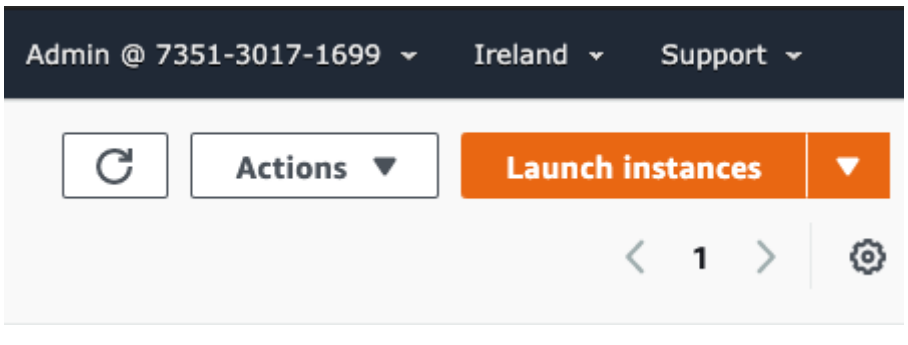
1. Navigate to the AWS Console and sign in here
2. Search for EC2 under the services dropdown menu, and click on EC2, shown in the screenshot



3. On the left panel, navigate to **Instances**



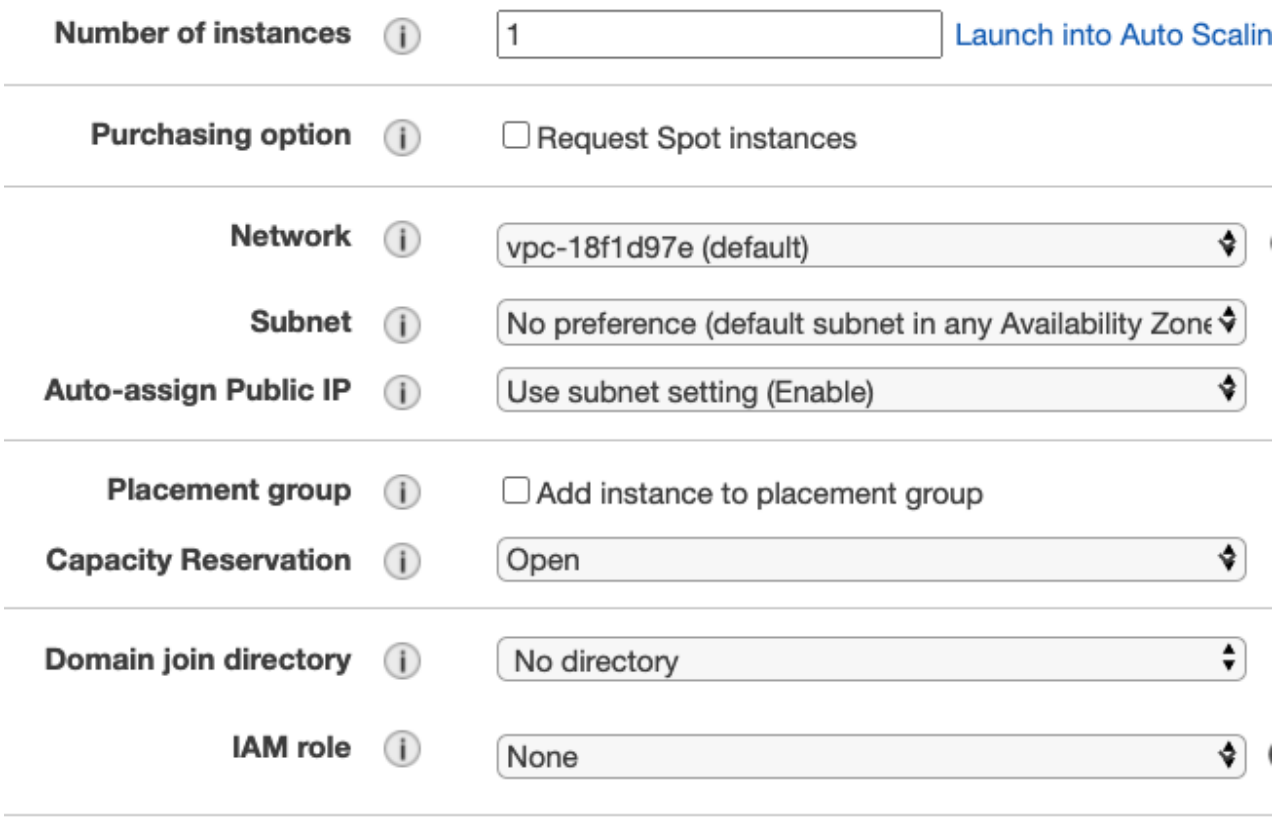
4. Click on the orange **Launch Instances** button on the right-hand side



5. You will be shown a large number of AMIs. Scroll down and select the **Ubuntu 18.04 LTS** AMI

6. You will be directed to a list of tiers which you can select. By default, **t2.micro** should be selected, if not, select **t2.micro** and continue to configuring details.

7. In the next window, you will be configuring your EC2 instance, how many instances you want, network
configurations and so on. This is shown in the screenshot below:

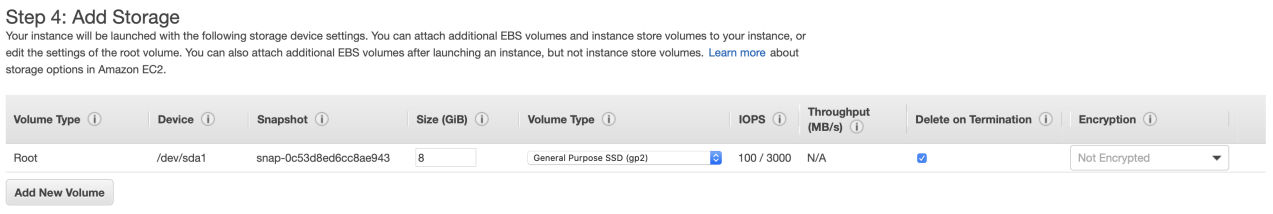


Leave everything as default.

8. In the next window, you will define how much additional storage you want to attach to your EC2 instance. The
default storage will be used to install the OS. This will be the minimal storage the OS needs to run.

The second storage will be attached to your EC2 instance but you will need to mount the additional volume to a
directory in your OS.

The screenshot shown below is what you should be expecting to see. Note, the snapshot id shown is going to be
different for whenever you create a new instance as well as unique for each user.



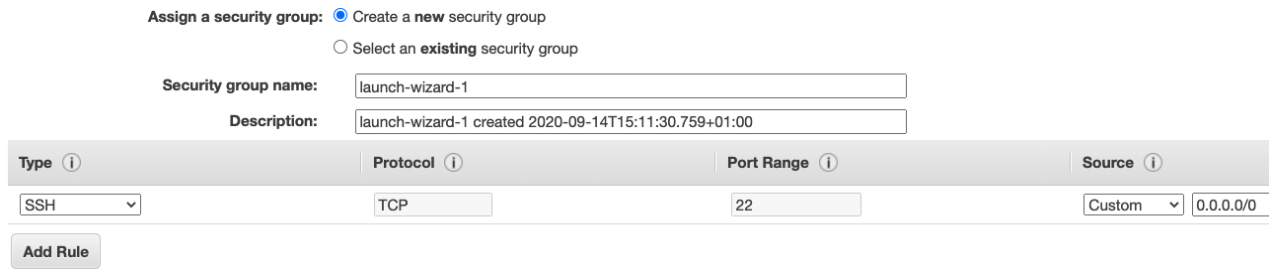
If you use a **Windows** server then this will be a 30GB storage for your OS to run.

Click **Next: Add Tags** and continue

9. You can make a key-value pair tags for your instance(s), so you can identify what your instance is being used for or for any other advanced architecture. Leave it empty and continue onto **Next: Configure Security Group**

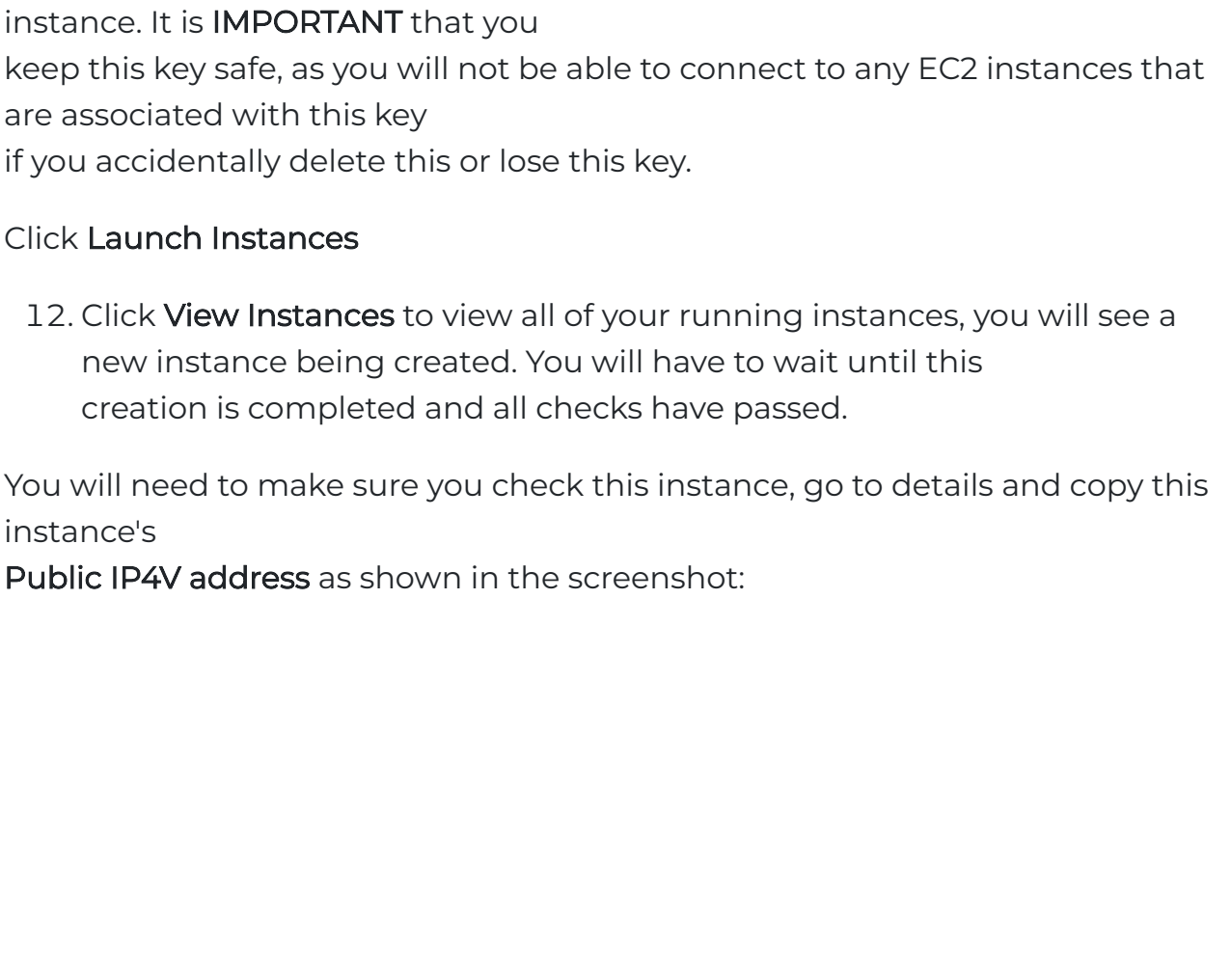
10. Security group is similar to firewall rules. As you will define the ports you want to open and the source IP Address you want that port to be open to. Most of the time you will only configure the inbound rules and leave the outbound rules to its default setting.

By default, the inbound rule has port 22 open to the whole world, meaning anyone in the world can SSH into the EC2 instance provided they have the credentials. Leave it as the default setting, you will be expected to ensure good security as you progress in this course.



Continue to **Review and Launch**

11. Once you're happy with all the configuration, click **Launch**. You will be prompted to select an existing key
or create a new key. Select **Create a new key pair**, and provide a name for your key pair.

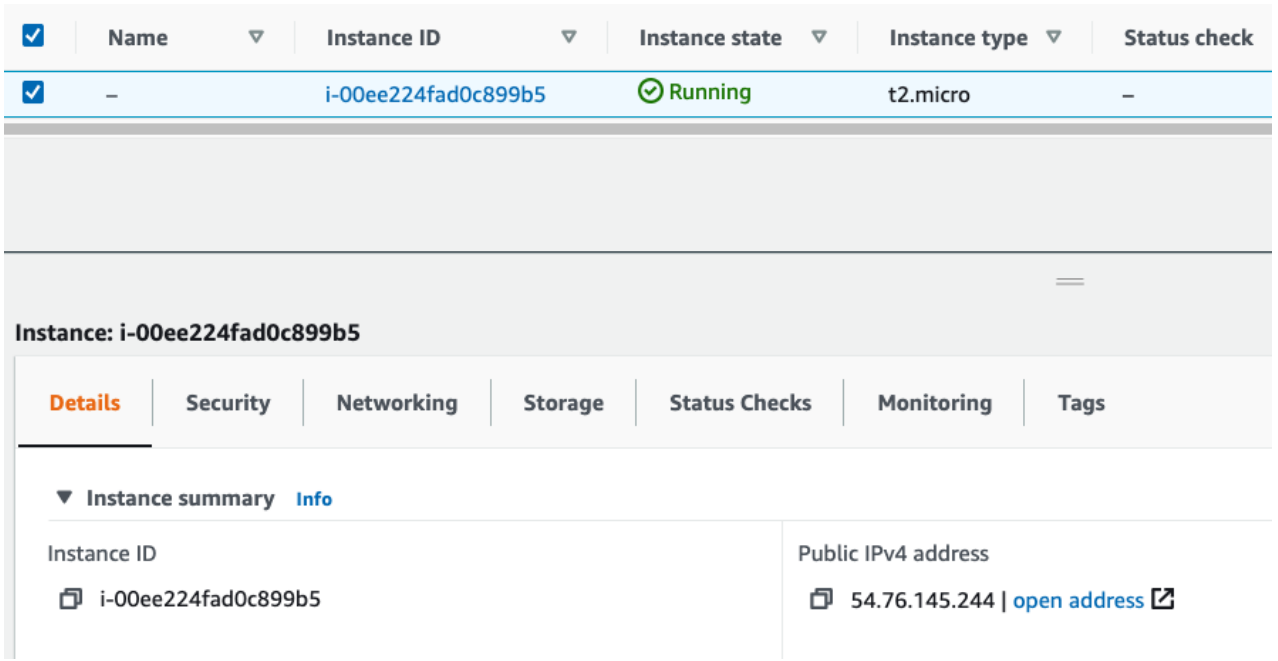


You will then need to download the key or you won't be able to launch your instance. It is **IMPORTANT** that you
keep this key safe, as you will not be able to connect to any EC2 instances that are associated with this key
if you accidentally delete this or lose this key.

Click **Launch Instances**

12. Click **View Instances** to view all of your running instances, you will see a new instance being created. You will have to wait until this
creation is completed and all checks have passed.

You will need to make sure you check this instance, go to details and copy this instance's
Public IP4V address as shown in the screenshot:

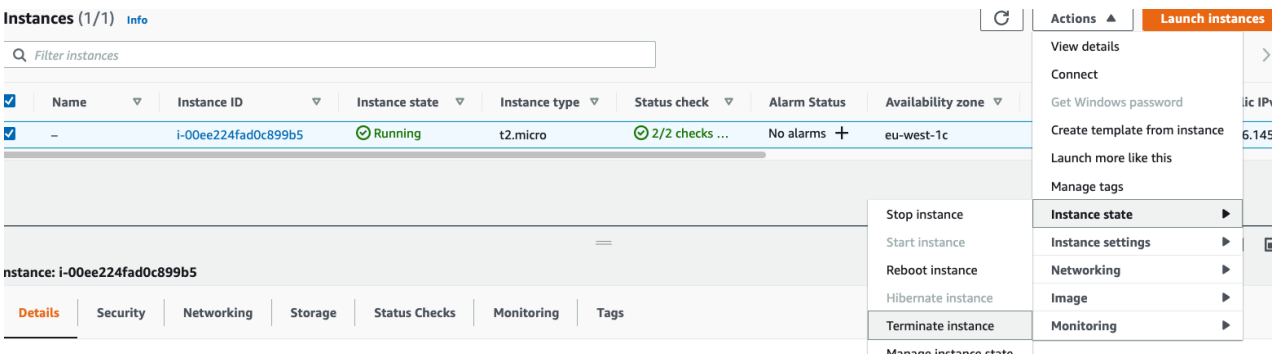


13. Lets SSH into the EC2 Instance. Navigate to where your key has been downloaded to, usually in the "Download" folder, open up *GitBash* or the *Terminal*, and execute the following command:

```
chmod 400 <Your pem key>
ssh -i <Your pem key> ubuntu@<EC2 instance IP Address>
```

14. To ensure you are not using up all your free-tier usage limit, you will need to **terminate** your EC2 instance.
    1. Select the instance you want to terminate
    2. Click on the **Action** button
    3. Go to **Instance State** then **Terminate instance**

Note: the terminated instance will appear in the list of instances for some time before it gets removed from your list, this is normal behavior



AWS CLI

In order to use the AWS CLI, you will require to make sure to have configured your machine with AWS specifically
setting the region to **eu-west-1**.

1. We will be running an EC2 instance through AWS CLI. You will need to make sure you have created at least
   a key-pair to associate with the instance, if not, then I would suggest looking at the module that
   goes over creating key-pairs.

Running an Instance

Running an instance requires quite a few options:

- **Image ID:**
  This is the base image that the machine will use, and it usually comes with an operating system install, such as Ubuntu.
- **Count:**
  The amount of instances to run from this command
- **Instance Type:**
  The size of the machine, how many CPUs and how much RAM
- **Key Name:**
  The Key Pair to install onto the machine. We need this to be able to securely connect to the instance with SSH.

- Subnet ID:
  Which subnet to run the instance in. The instance will be given a private IP address, from the subnet that you put it in, and security group rules from the VPC will be applied to the instance (if no `subnet-id` is specified, the `default` will be used).

```
# aws ec2 run-instances --image-id [IMAGE_ID] --count [AMOUNT_OF_INSTANCES] --instance-type [MACHINE_SIZE] --key-name [KEY_PAIR_NAME] --subnet-id [SUBNET_ID]
aws ec2 run-instances --image-id ami-04137ed1a354f54c4 --count 1 --instance-type t2.micro --key-name(key-pair-name)
```

The above command will launch an EC2 instance of Type **t2.micro** in **Ireland**.

2. We can view all of our instances, as if we were logged in our AWS Web Console and viewing the
   EC2 Dashboard.

View Running Instances

To check what existing instances there are, we can use the `describe-instances` command:

```
aws ec2 describe-instances
```

When running the above command, please make sure you have copied the instance-id, for the instance you want to
delete. We will be using this instance-id for the next step.

3. We can now delete our instance.

Terminate a Running Instance

We can terminate instances by providing their IDs to the `terminate-instances` command:

```
# aws ec2 terminate-instances --instance-ids [INSTANCE_IDS]
aws ec2 terminate-instances --instance-ids (your instance id)
```

When running this command, it will output the status of your instance if it succeeded in terminating
or not.

Exercises

There are no exercises for this module.