Exercise 1

Start up an instance on Amazon EC2 and get Apache web server running

Prior Knowledge
Unix Command Line Shell

Learning Objectives
Understand about EC2 instances
Start an instance using the web interface
Configure the AWS command line
Manage instances from a command line
Understand Security Groups

Software Requirements
(see separate document for installation of these)

- AWS CLI

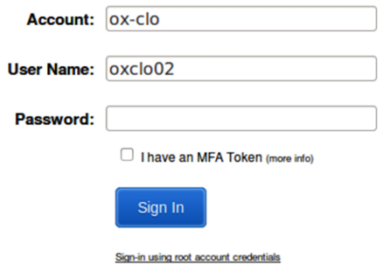
Part A: Starting an Instance from the Web Console.

1. You have been provided with an Ubuntu VM. Start that up. Please ask the TA or lecturer if you don't know how to do that.

2. The course is also providing time and resources on the Amazon AWS/EC2 cloud for the duration of the course. Please don't abuse this!

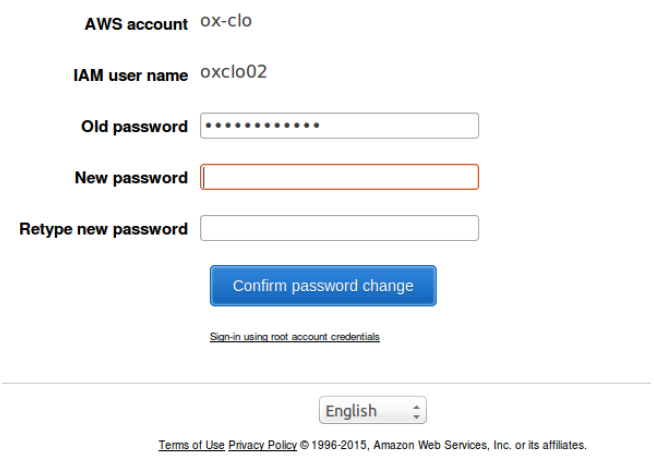
3. Although it is possible to do the following exercises on your normal "host" OS, please do not! Part of this exercise is to install a key and access key into the Ubuntu VM which is needed for later exercises, **so please do this inside the Ubuntu VM.**

4. Open up a browser window and navigate to
   https://ox-clo.signin.aws.amazon.com/console

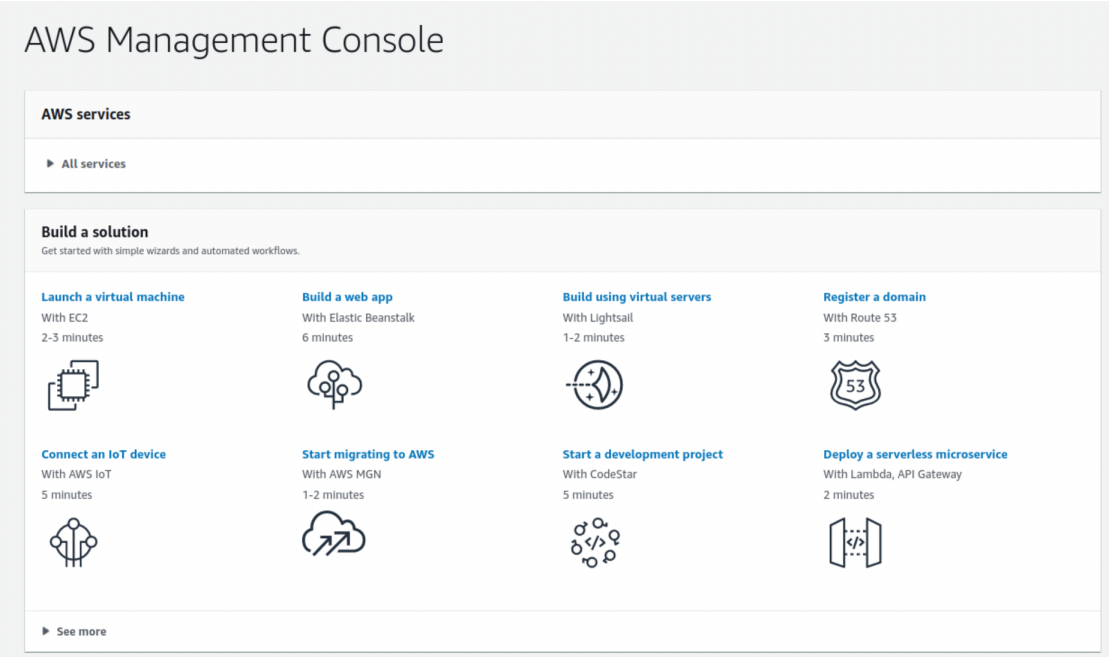


   Hint: make a bookmark for that URL

5. Use the userid and password **that you have been given**. You will need to create a new password:

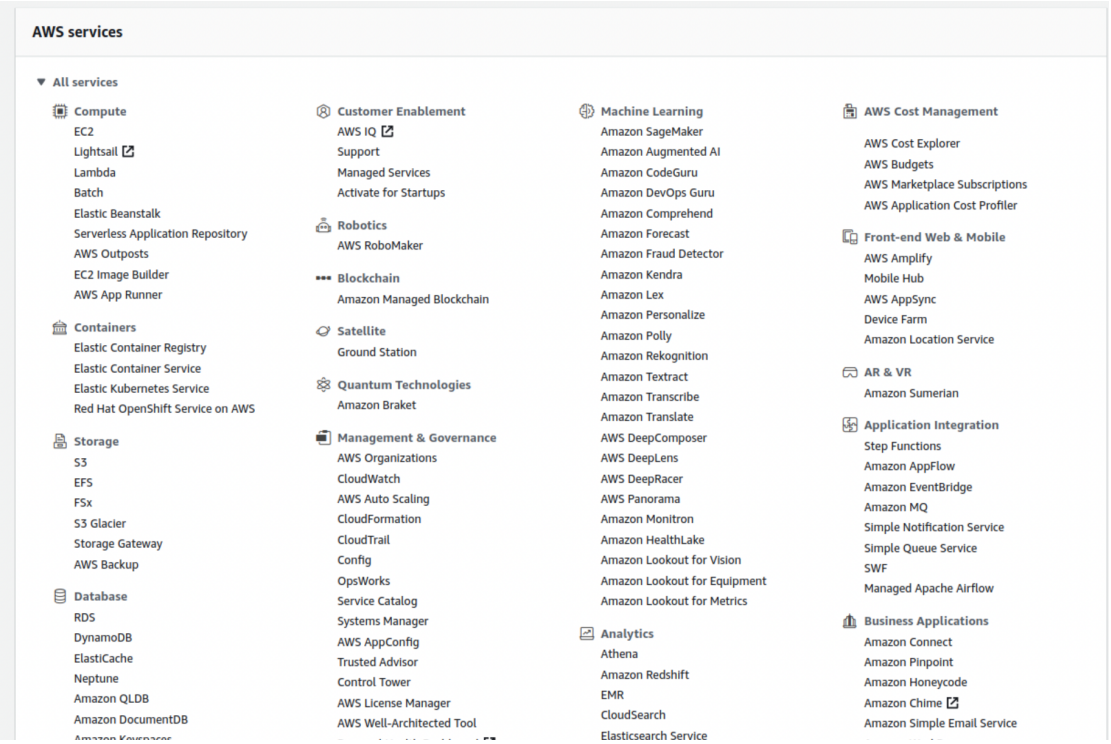


6. You should see a screen like this:



7. In the top right corner click on Oregon and change to **EU (Ireland) (unless it is already on Ireland!)**

8. Expand **All Services:**

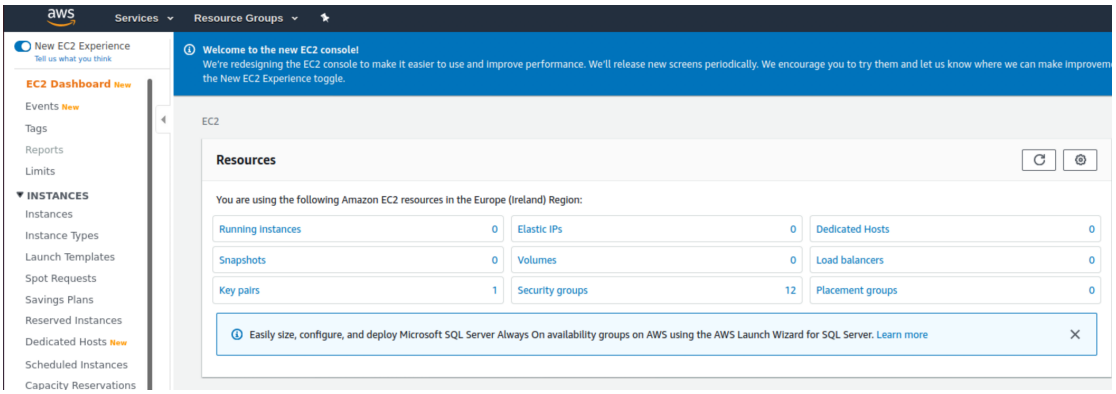


9. Now click on the link **EC2**

10. Please note:

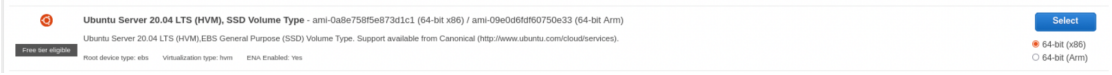
> *You will be working in a shared environment with other students on the course (unless you have chosen to use your own Amazon account). As a result, we will need to be very careful not to interfere with other students' instances, volumes, etc. Therefore please be careful to **tag and name** your resources clearly so that you can identify them. (Instructions on how to do that will follow!).*

11. As a result, the screen below will differ depending on who has done different parts of this exercise.



12. Click on the button: **Launch Instance**

13. Choose "**Ubuntu Server 20.04 LTS (HVM), SSD Volume Type**"



14. Choose the instance type **t2.micro**.

15. Click **Next: Configure Instance Details**

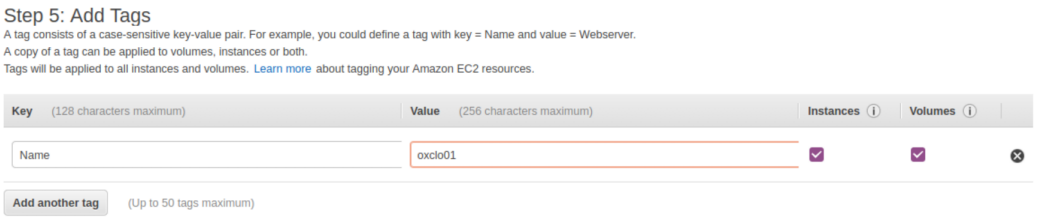
16. Click **Next: Add Storage**

17. Click **Next: Add Tags**

18. In the Tag Instance screen, give your instance a Name.

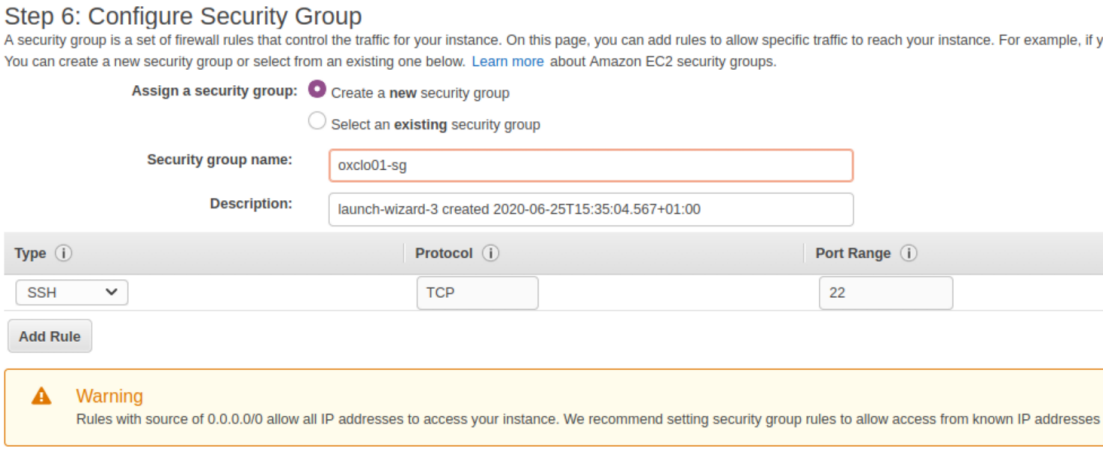
Add a tag.

Make the *Key* be **Name**
Make the *Value* the same as your numbered userid (e.g. oxclo01)



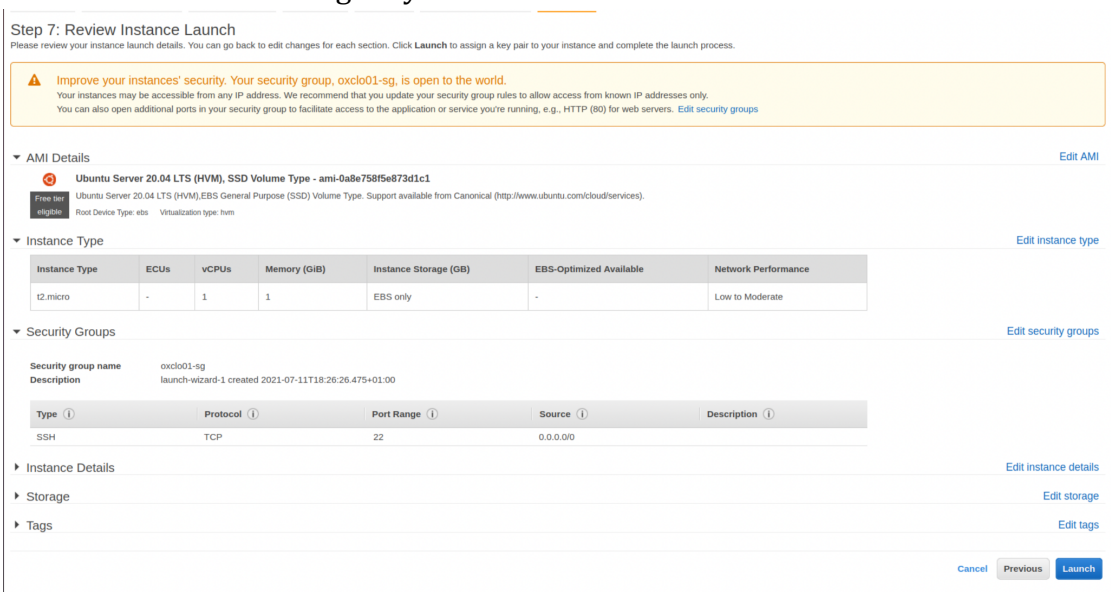
19. Now click: **Next: Configure Security Group**

20. Change the name of the security group to <your userid>-sg.



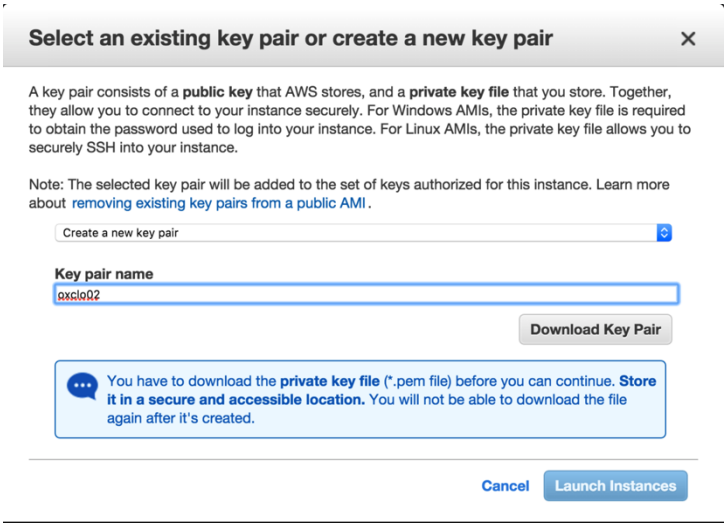
21. Click **Review and Launch**

You should see something very like this:



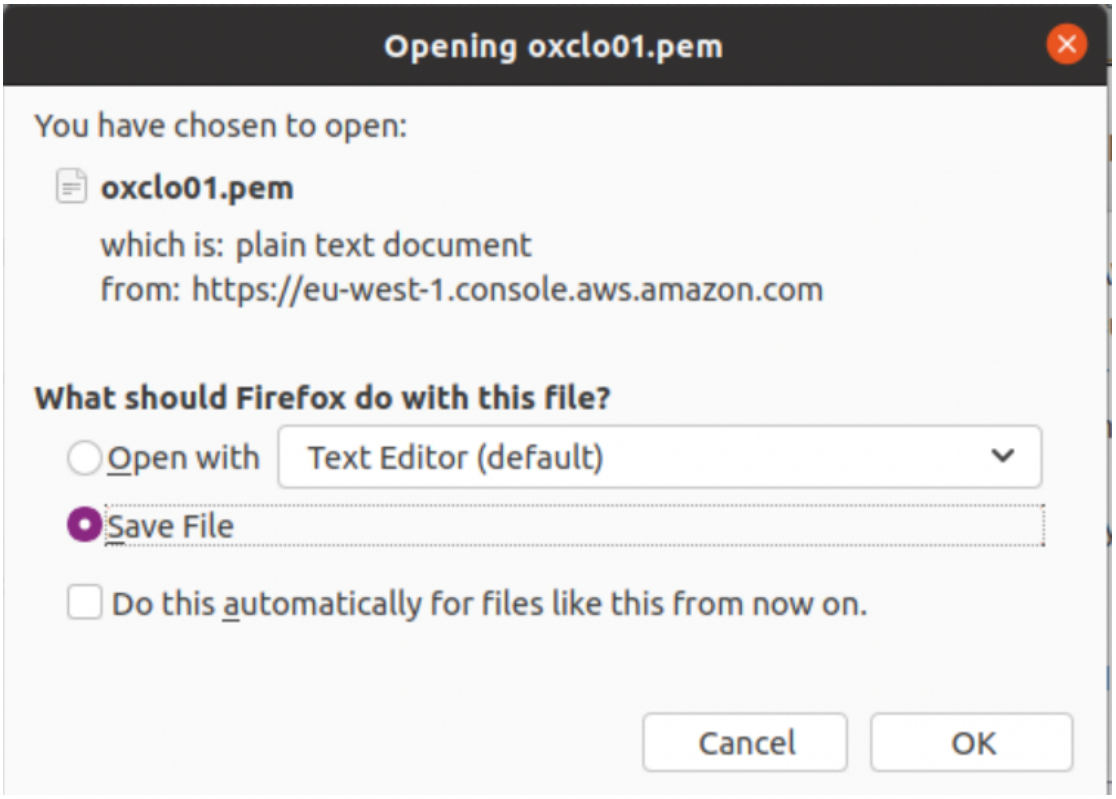
22. Click **Launch**

23. You will be prompted with a new window to decide on the correct key pair to secure this instance with. Since this is the first time you are using EC2, you need to create a key pair. Change the dropdown box to **Create a new key pair.**



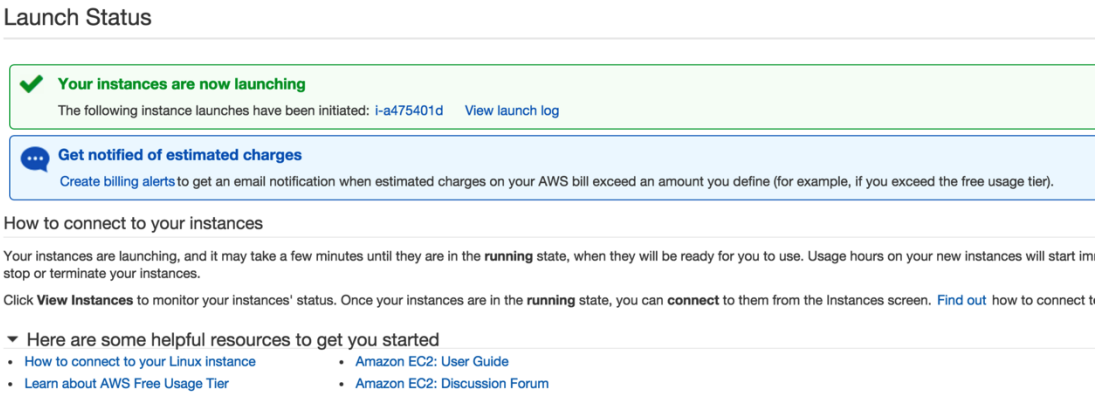
24. Change the name of the key pair to your numbered userid.

25. Click **Download Key Pair**.

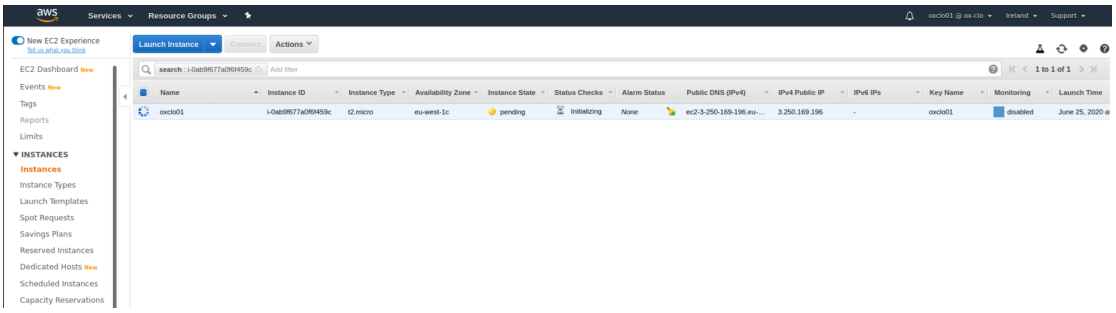


Save File. This will save a file to your ~/Downloads directory.

26. Click **Launch Instances**
You should see something like:



27. Click on the blue instance ID link (e.g. **i-a475401d** in the screenshot above)
    You will see a dashboard like:



28. Make sure you are running the Ubuntu VM, and start a fresh terminal window (Ctrl-Alt-T, or find Terminal in the side bar)

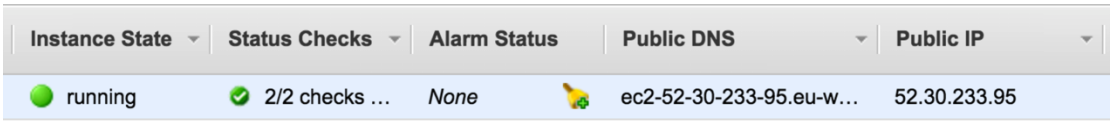
29. Check is there is already a `~/keys` directory.

    If not, then make a directory to store your private key:
    `mkdir ~/keys`

30. Copy your private key to the new directory:
    `cp ~/Downloads/oxclo*.pem ~/keys/`

31. Before you can use the key you need to change the permissions on it. Type:
    `chmod 400 ~/keys/oxclo*.pem`

32. Check to see if the status checks on your instance are now complete. Refresh the browser window:



33. Copy the Public IP Address from the browser window (e.g. 52.30.233.95 in my case)

34. Try to SSH into the machine. Replace your key file name and the IP address below!

    `ssh -i ~/keys/oxclo**nn**.pem ubuntu@**ww.xx.yy.zz**`

35. As this is the first time you are accessing this host, the key on the server side is not known. You should see something like:

```
The authenticity of host '52.30.233.95 (52.30.233.95)' can't be
established.
ECDSA key fingerprint is
SHA256:7GhOakN9Pj3vWAegV0uYhPVI9qqVEe9RlNM0wcutO1E.
Are you sure you want to continue connecting (yes/no)?
```

Type **yes** and hit Enter.

You will see something like:

```
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-1045-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun Jul 11 17:34:12 UTC 2021

  System load:  0.45              Processes:             103
  Usage of /:   16.4% of 7.69GB   Users logged in:       0
  Memory usage: 22%               IPv4 address for eth0: 172.31.26.217
  Swap usage:   0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

36. **Congratulations** – you have a cloud instance running.

PART B – Running a Web Server

37. In the SSH shell type:

```
sudo apt update
```

You will see a lot of log, e.g.:

```
Hit http://eu-west-1.ec2.archive.ubuntu.com trusty/universe
Translation-en
Ign http://eu-west-1.ec2.archive.ubuntu.com trusty/main
Translation-en_US
Ign http://eu-west-1.ec2.archive.ubuntu.com trusty/universe
Translation-en_US
Fetched 10.3 MB in 3s (2,713 kB/s)
Reading package lists... Done
```

38. Now type:

```
sudo apt install apache2
```

39. You will see:

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apache2-bin apache2-data libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine apache2-suexec-custom apache2-utils
  openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap ssl-cert
0 upgraded, 8 newly installed, 0 to remove and 130 not upgraded.
Need to get 1,285 kB of archives.
After this operation, 5,348 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

40. Hit Enter (same as Y). The log should look like:

```
Enabling conf serve-cgi-bin.
Enabling site 000-default.
 * Starting web server apache2
*
Setting up ssl-cert (1.0.33) ...
Processing triggers for libc-bin (2.19-0ubuntu6.6) ...
Processing triggers for ureadahead (0.100.0-16) ...
Processing triggers for ufw (0.34~rc-0ubuntu2) ...
```

41. Check locally if it is running:

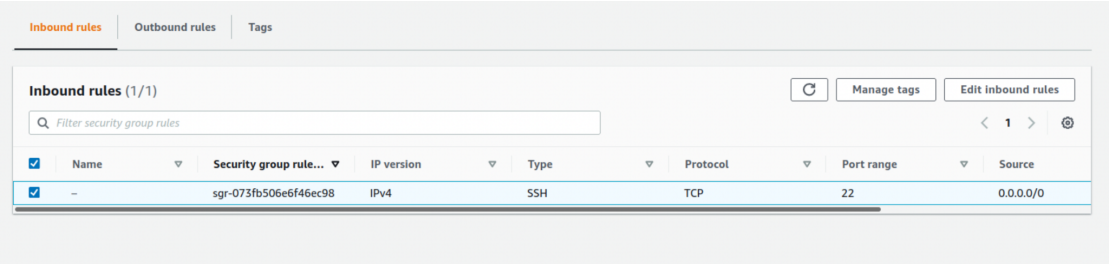
a. curl http://localhost

b. You should see a lot of HTML scroll by.

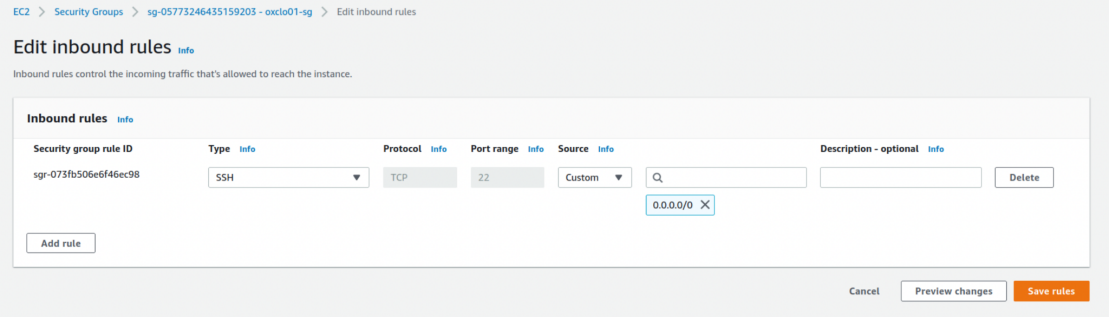
42. Now try browsing the server from your local machine. Find the Public IP address or Public DNS name from the EC2 console and use that in a browser window.

43. It will timeout because we have not enabled port 80 (www) to be accessed. Go back to the EC2 dashboard, and choose **Security Groups** from the left hand menu.

44. Find the group that you created that uses your userid as the Group Name, select it, and then choose **Edit Inbound rules**



45. You should see:



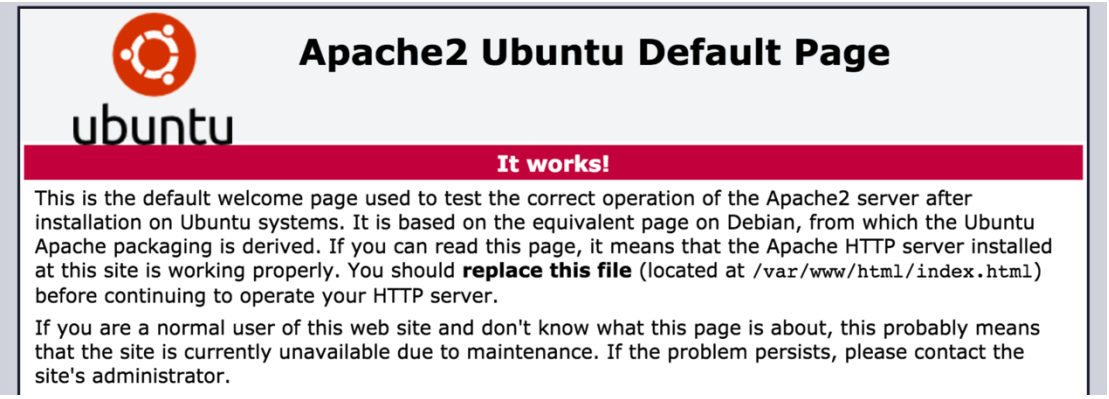
Click **Add Rule**

46. Click on the drop down box that says "Custom TCP Rule" and change it to **HTTP**.

47. Change the **Source** from Custom to **Anywhere-IP4**

48. Click **Save rules**.

49. Now try browsing to the webpage again. You should see:



50. Congratulations!

PART C – Using the AWS Command Line

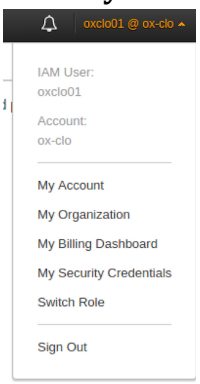
51. The AWS Command Line (AWS CLI) is available as part of the Python PIP installed code. PIP is a package manager for Python.

52. Now you can configure the AWS command line with your credentials

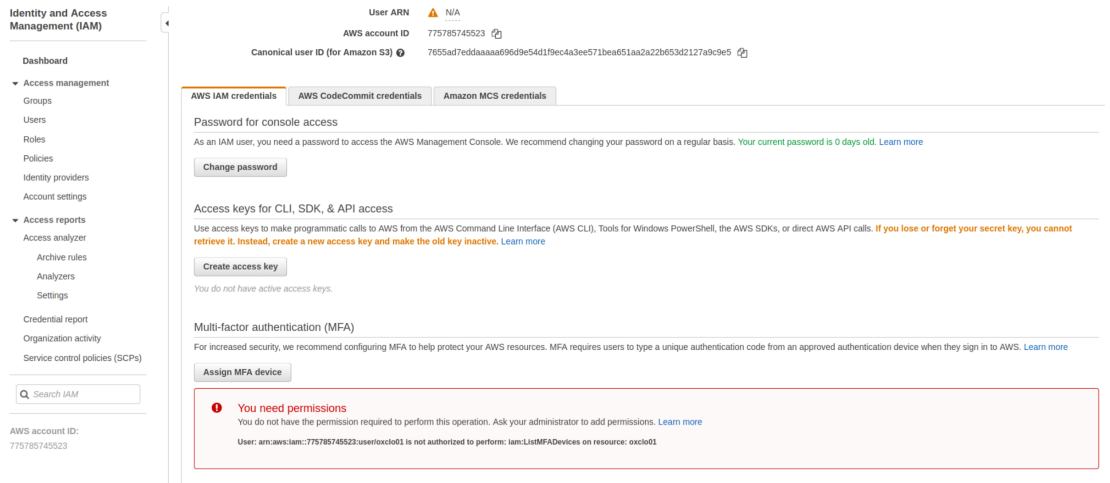
    a. First we need to create an Access Key and Secret Key for you. I could have printed one out for you, but that would be difficult to type in, so let's go create one in the AWS Console.

    b. Go to the AWS Console

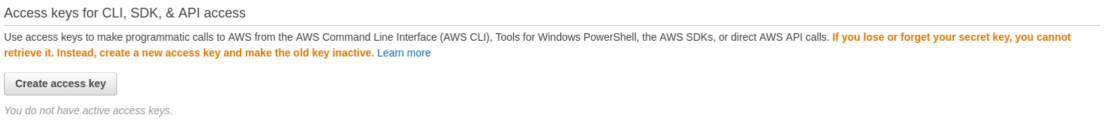
    c. In the top right corner, click on your username, then choose **My Security Credentials**:



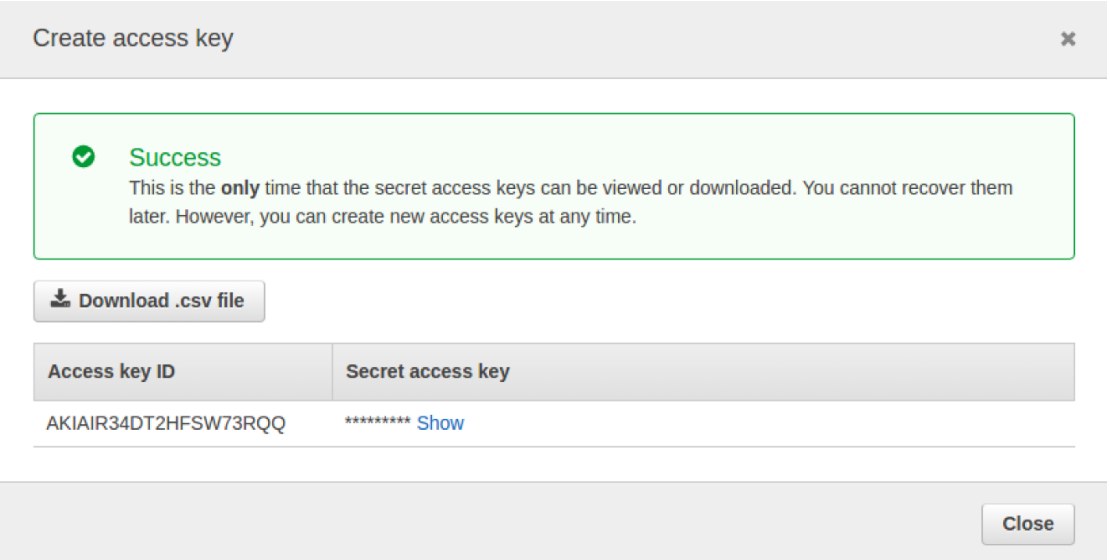
    d. You should see something like:



Click on **Create Access Key.**

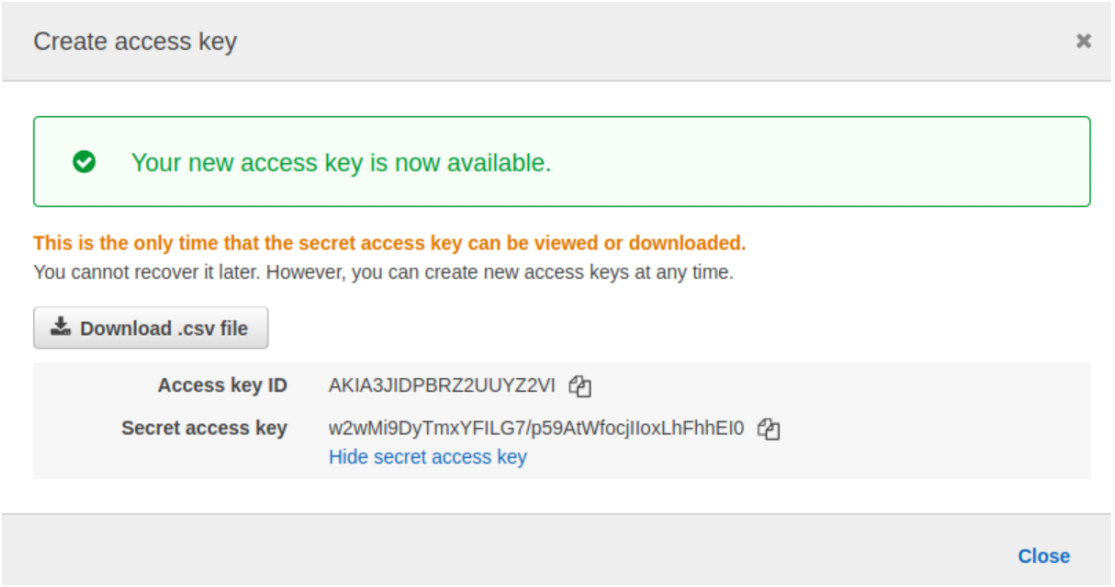


You should see



e. Click **Download .csv file** and then **Save**

f. You can also click Show and then copy and paste these two token identifiers into a new text file



g. You need to make a note of these credentials or download them, because the secret key will not be available again.

53. Now we can use these keys to configure the AWS CLI.
    In a fresh terminal window (**NOT THE SSH ONE)** type:

    ```
    aws configure
    ```

    a. When prompted
    AWS Access Key ID [None]:

    Type the Access Key ID from the text file or CSV (cut and paste)

    b. Do the same for the Secret Access Key.

    c. For the region choose Ireland: **eu-west-1**

    d. For the output format, type **json**

Hint: You now have three credentials for AWS:

- *Your userid/password*
- *An Access Key/Secret Key for controlling EC2/AWS through command line, third-party tools and apps, and any Web Service APIs*
- *An SSH Private Key pair for accessing the actual instances that you startup.*

54. Now let's use the CLI to terminate your instance.

55. From the console (we could get this from the CLI too, but its complex to describe) copy the instance id of your running instance.

56. Now use the AWS CLI to terminate:
    Replacing the instance ID with your own, type:

    ```
    aws ec2 terminate-instances --instance-ids i-0b735618d9e69b35b
    ```


57. You should see log like:

```
aws ec2 terminate-instances --instance-ids i-0fa3d4032833ea933
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-0fa3d4032833ea933",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

    Your SSH session to the server will die, and the web site will no longer be running.

58. Congratulations! You have completed all three parts of this Lab.