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)

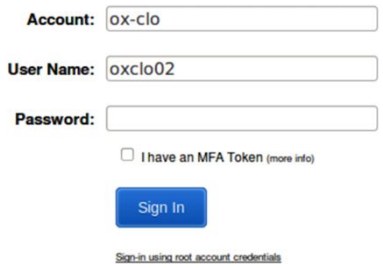
- AWS CLI

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.

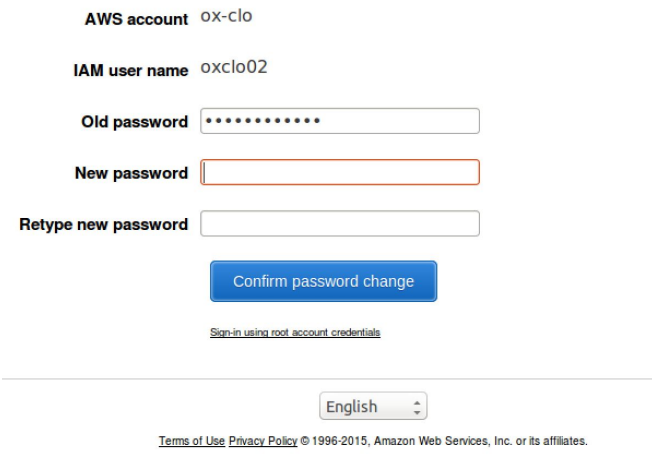
2. The course is also providing time and resources on the Amazon AWS/EC2 cloud for the duration of the course.

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

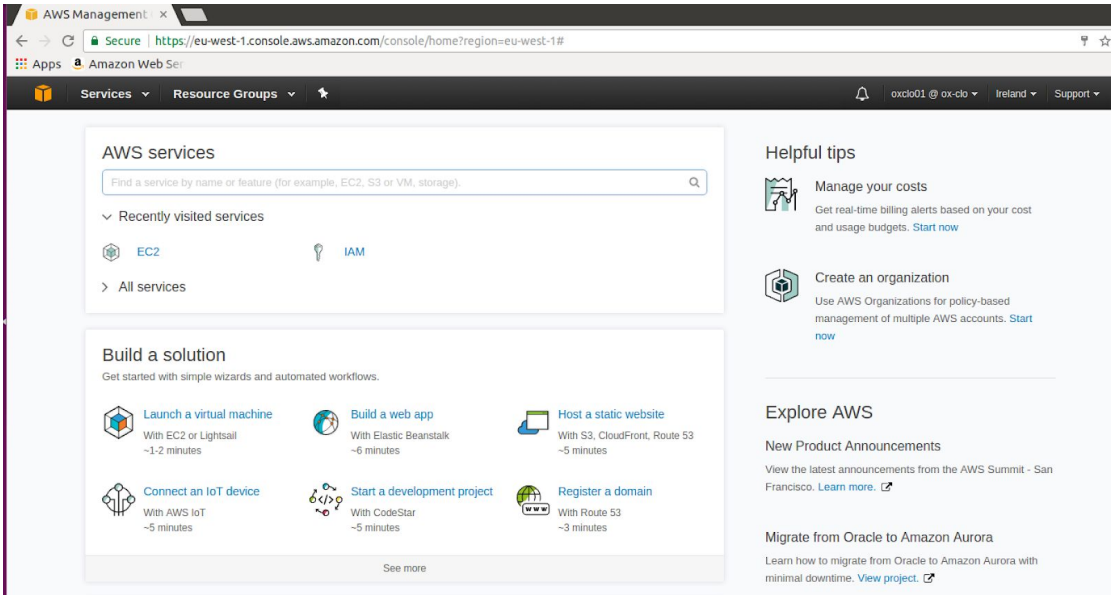


Hint: make a bookmark for that URL

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

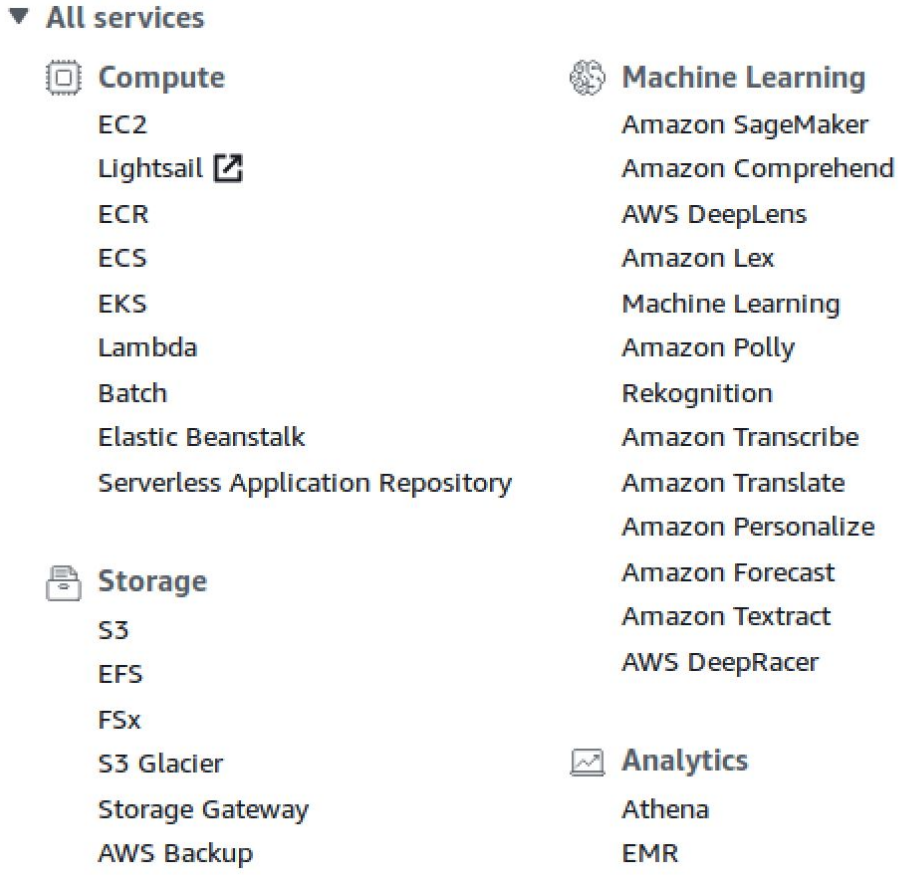


5. You should see a screen like this:



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

7. Expand **All Services:**

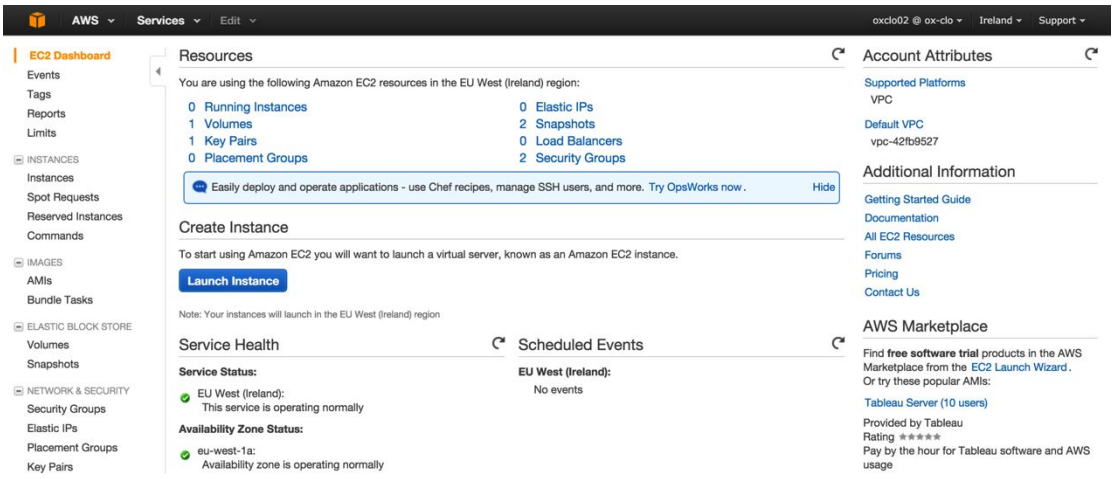


8. Now click on the link EC2

9. Please note:

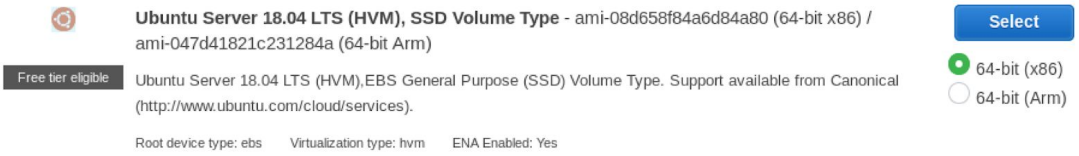
> *You will be working in a shared environment with other students on the course (unless you have chosen to use your own Amazon account). As a result, we will need to be very careful not to interfere with other students' instances, volumes, etc. Therefore please be careful to **tag and name** your resources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

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



10. Click on the blue button: Launch Instance

11. Choose "**Ubuntu Server 18.04 LTS (HVM), SSD Volume Type**"



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

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

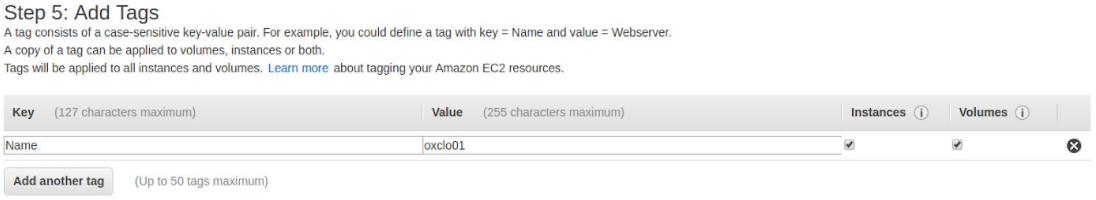
14. Click **Next: Add Storage**

15.

Click **Next: Add Tags**

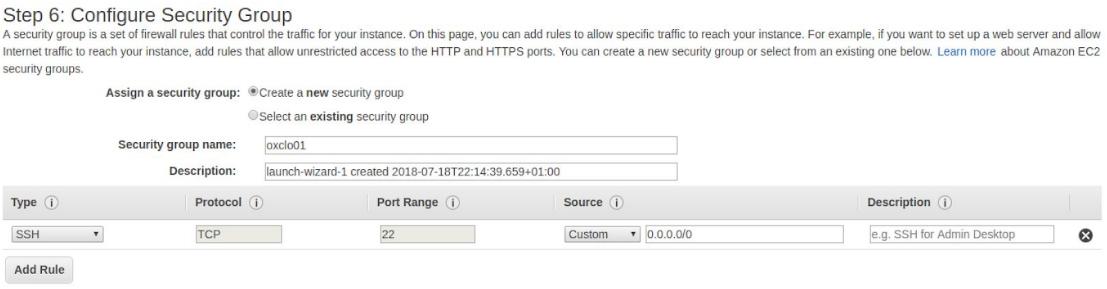
16. In the Tag Instance screen, give your instance a Name.

Make the *Key* be **Name**
Make the *Value* the same as your userid.



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

18. Change the name of the security group to your userid.



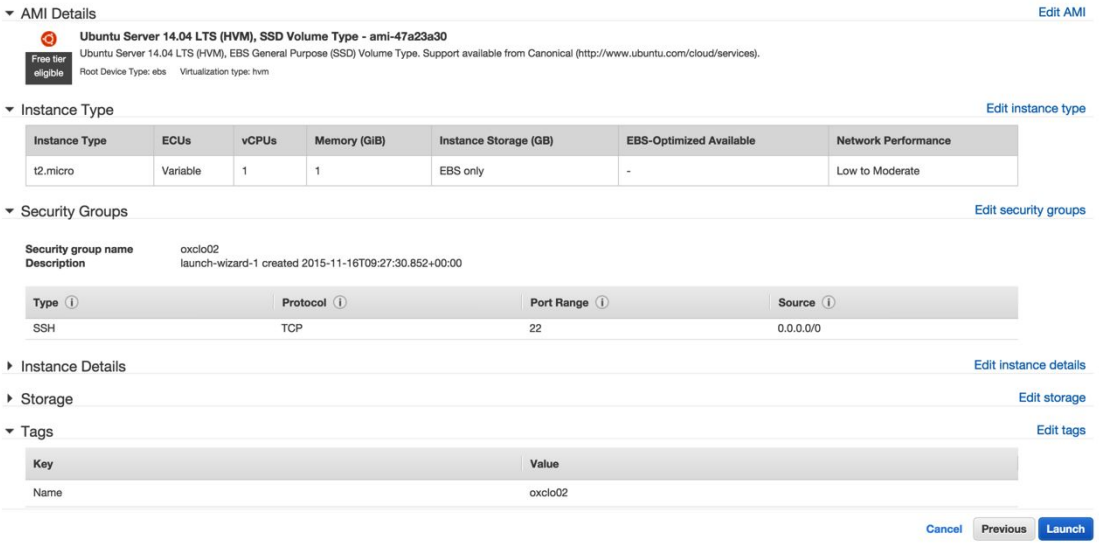
19.

Hint: There is a security warning about the security rule. The default rule allows Secure Shell (SSH) access from any IP address. If you know your company or personal internet connection comes from a specific IP address you can improve security by restricting to that.

Note this is NOT the IP address you get by looking at the local machine's configuration, but the publicly visible IP address that the Amazon cloud sees from you. You can see what your IP is by typing "what's my IP" into Google.

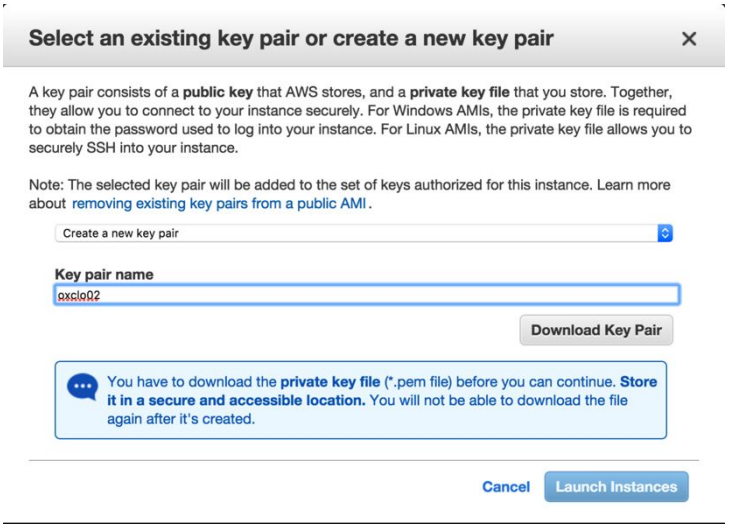
However, I am not sure if the Oxford network sends messages from different IPs or the same and therefore we will leave this as-is despite the warning.

Click **Review and Launch**
You should see something very like this:



20. Click **Launch**

21. 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.**

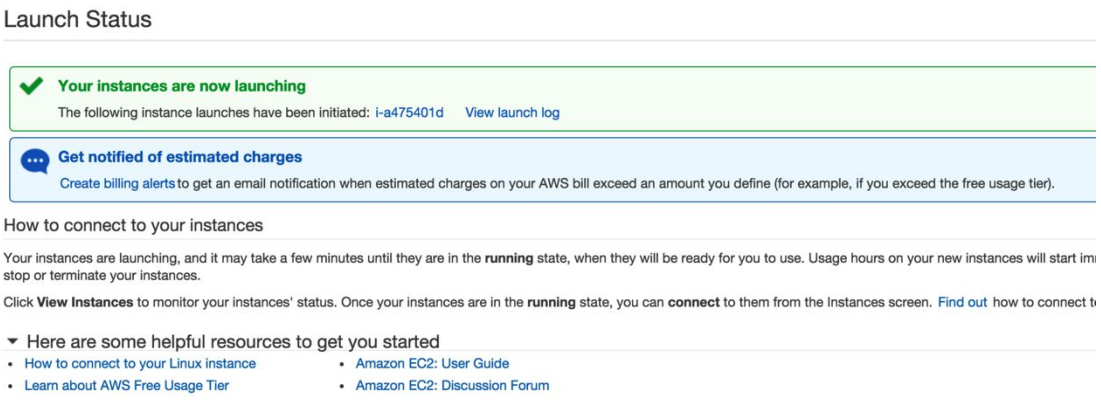


22. Change the name of the key pair to your userid.

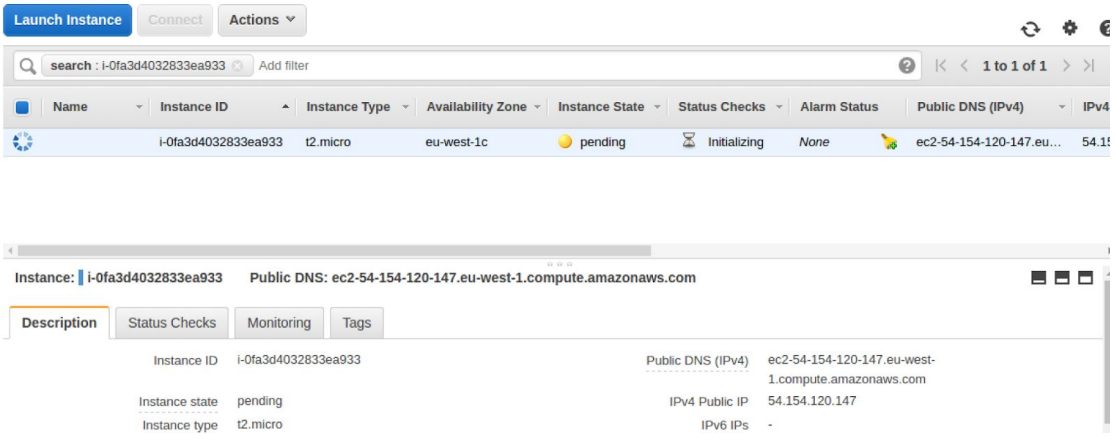
23. Click **Download Key Pair**. This will save a file to your ~/Downloads directory.

24.

Click **Launch Instancesc**
You should see something like:



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



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

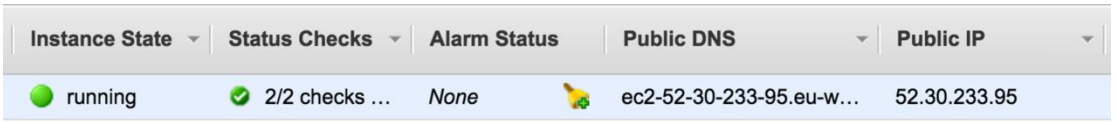
27. Check is there is already a ~/keys directory.

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

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

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

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



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

32. 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**

33. 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 16.04.2 LTS (GNU/Linux 4.4.0-1020-aws x86_64)

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

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.


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.
```

34.

Congratulations – you have a cloud instance running.

PART B – Running a Web Server

35. In the SSH shell type:
    sudo apt update

```
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
```

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

36. Now type:
sudo apt install apache2

37. 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]
```

38. 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) ...
```

39. Check locally if it is running:

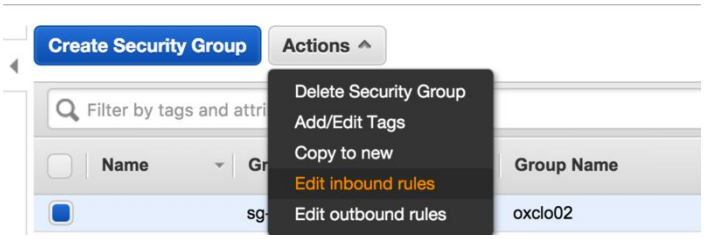
a. curl http://localhost

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

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

41. 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.

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

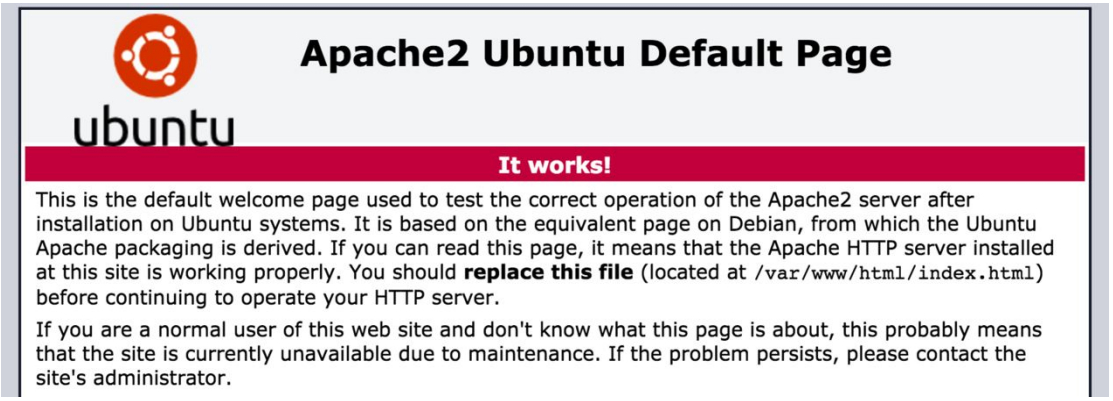


43. Click **Add Rule**

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

45. Click **Save**.

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



47. Congratulations!

PART C – Using the AWS Command Line

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

49. In a fresh Ubuntu Terminal Window (*make sure you are not doing this on your cloud server by mistake!)*

a. Type:

```
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in
/usr/local/lib/python2.7/dist-packages (from
botocore==1.5.80->awscli) (0.9.3)
Requirement already satisfied: pyasn1>=0.1.3 in
/usr/local/lib/python2.7/dist-packages (from
rsa<=3.5.0,>=3.1.2->awscli) (0.2.3)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python2.7/dist-packages (from
python-dateutil<3.0.0,>=2.1->botocore==1.5.80->awscli)
(1.10.0)
```

sudo pip install awscli

You should see log ending like:

This is because the it is already installed. Otherwise you will see:

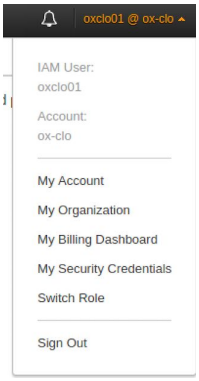
```
changing mode of /usr/local/bin/rst2s5.py to 755
  changing mode of /usr/local/bin/rst2xetex.py to 755
  changing mode of /usr/local/bin/rst2man.py to 755
  changing mode of /usr/local/bin/rst2html.py to 755
Successfully installed awscli docutils botocore rsa jmespath python-dateutil pyasn1
Cleaning up...
```

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

51. 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.

52. Go to the AWS Console

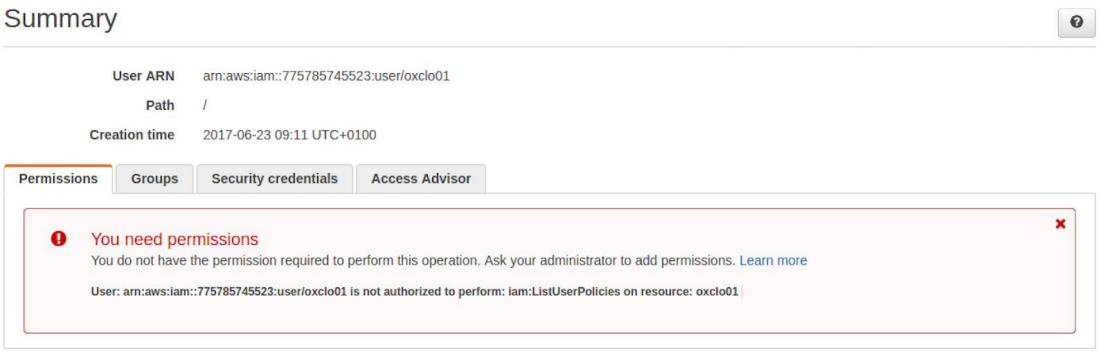
53. In the top right corner, click on your username, then choose **My Security Credentials**:



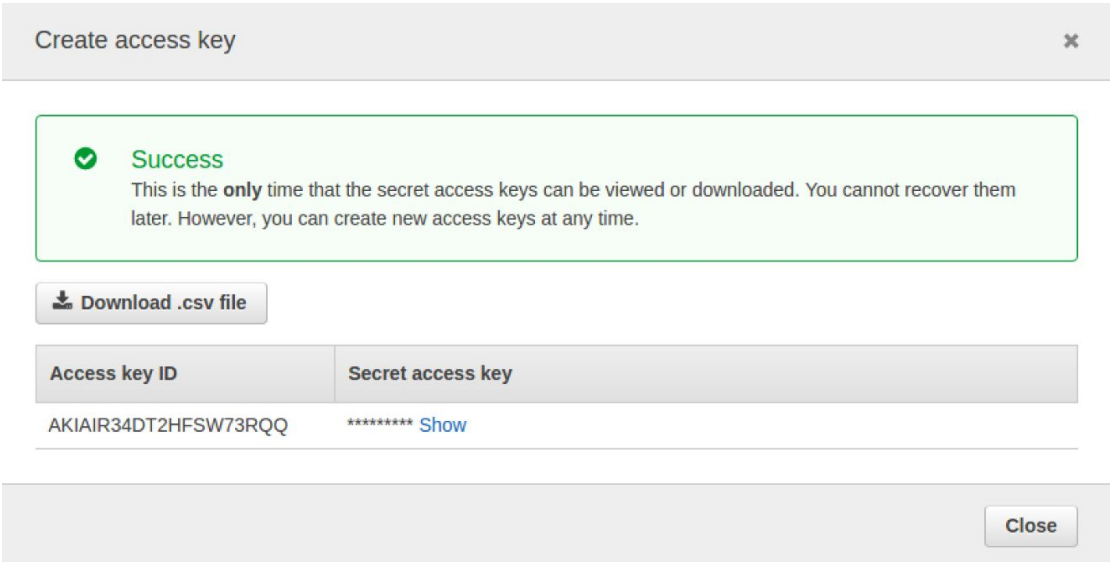
54. In the left hand menu choose **Users**

55. Click on your own userid

56. You should see something like

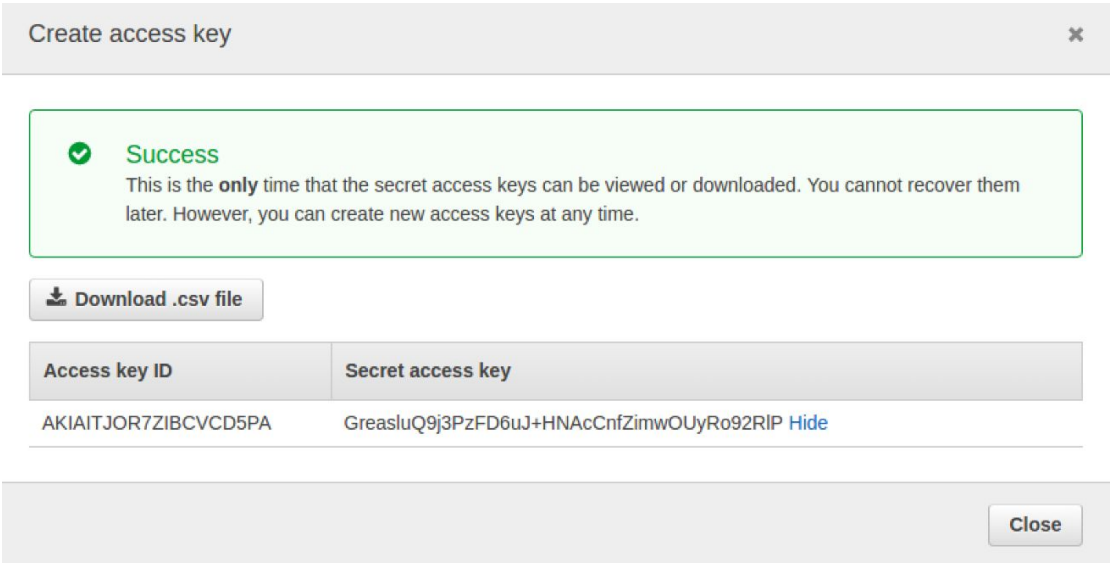


Scroll until you find: **Create Access Key.** Click on it. You will see:



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

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



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

60. Now we can use these keys to configure the AWS CLI. Back in the terminal window where you installed the AWS CLI, 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.*

61.

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

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

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

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

64. 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"
            }
        }
    ]
}
```

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

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