

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

The screenshot shows the AWS sign-in interface for the 'ox-clo' account. It includes input fields for 'Account' (pre-filled with 'ox-clo'), 'User Name' (pre-filled with 'oxclo02'), and 'Password'. Below the password field is a checkbox labeled 'I have an MFA Token (more info)'. A blue 'Sign In' button is positioned below the checkbox. At the bottom, there is a small link that reads 'Sign in using root account credentials'.

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:

AWS account ox-clo

IAM user name oxclo02

Old password

New password

Retype new password

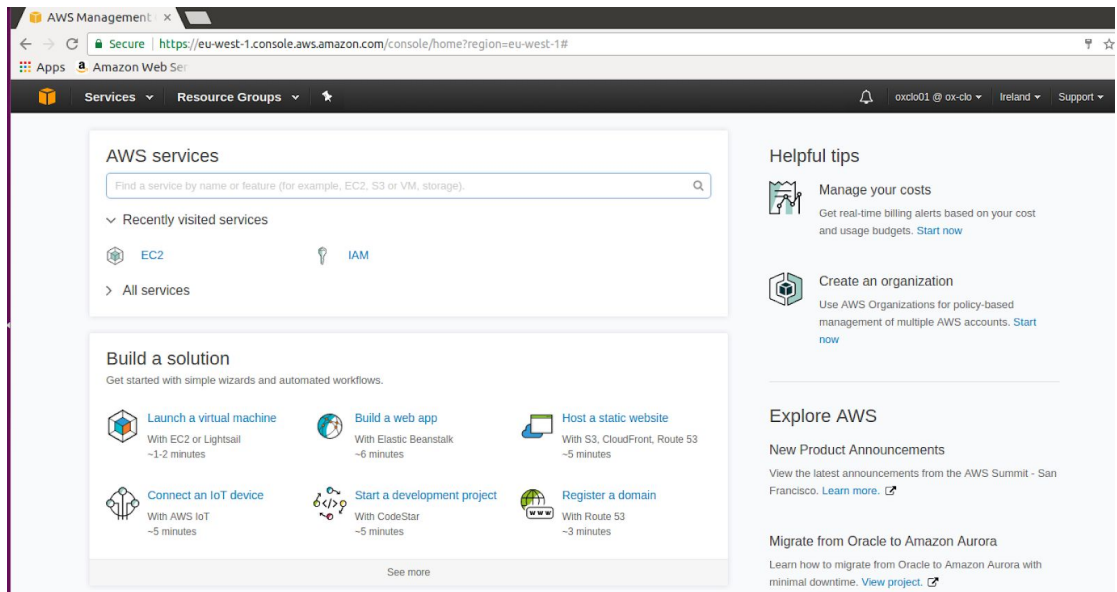
[Confirm password change](#)

[Sign-in using root account credentials](#)

English

[Terms of Use](#) [Privacy Policy](#) © 1996-2015, Amazon Web Services, Inc. or its affiliates.

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**:

▼ All services



Compute

- EC2
- Lightsail
- ECR
- ECS
- EKS
- Lambda
- Batch
- Elastic Beanstalk
- Serverless Application Repository



Storage

- S3
- EFS
- FSx
- S3 Glacier
- Storage Gateway
- AWS Backup



Machine Learning

- Amazon SageMaker
- Amazon Comprehend
- AWS DeepLens
- Amazon Lex
- Machine Learning
- Amazon Polly
- Rekognition
- Amazon Transcribe
- Amazon Translate
- Amazon Personalize
- Amazon Forecast
- Amazon Textract
- AWS DeepRacer



Analytics

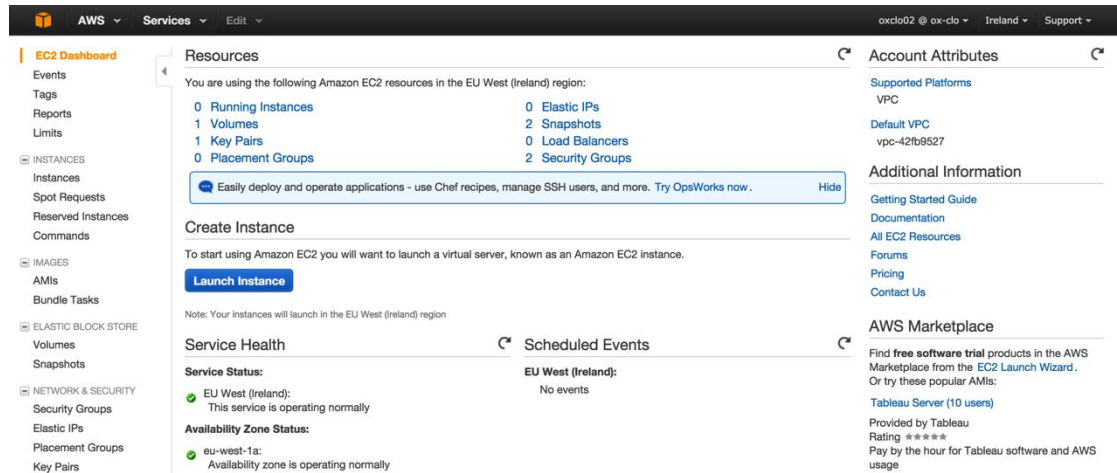
- Athena
- EMR

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

10. Please note:

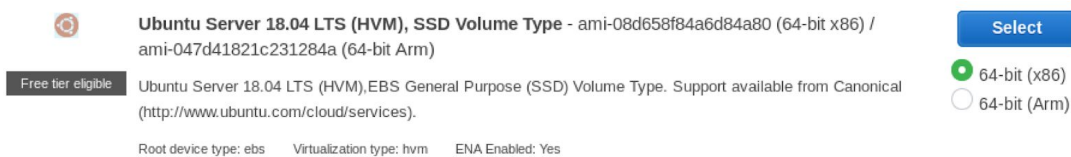
*You will be working in a shared environment with other students on the course (unless you have chosen to use your own Amazon account). As a result, we will need to be very careful not to interfere with other students' instances, volumes, etc. Therefore please be careful to **tag and name** your resources 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 blue button: **Launch Instance**

13. Choose “**Ubuntu Server 18.04 LTS (HVM), SSD Volume Type**”



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

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

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.

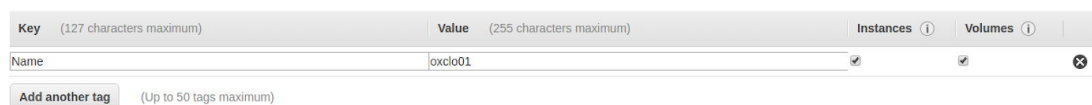
Make the *Key* be **Name**

Make the *Value* the same as your userid.

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.



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

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

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group
☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Add Rule

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.

21. Click **Review and Launch**

You should see something very like this:

AMI Details [Edit AMI](#)

 **Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-47a23a30**
 Ubuntu Server 14.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
 Root Device Type: ebs Virtualization type: hvm

Free tier eligible

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

Security group name: oxclo02
 Description: launch-wizard-1 created 2015-11-16T09:27:30.852+00:00

Type	Protocol	Port Range	Source
SSH	TCP	22	0.0.0.0/0

Instance Details [Edit instance details](#)

Storage [Edit storage](#)

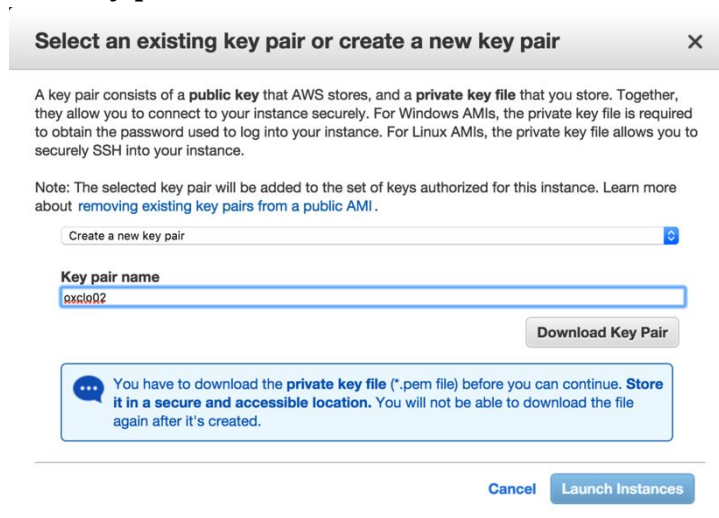
Tags [Edit tags](#)

Key	Value
Name	oxclo02

[Cancel](#) [Previous](#) [Launch](#)

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**.



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair name
pxclo02

Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel Launch Instances

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

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

26. Click **Launch Instances**

You should see something like:

Launch Status

✓ Your instances are now launching

The following instance launches have been initiated: i-a475401d [View launch log](#)

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately when you launch them.

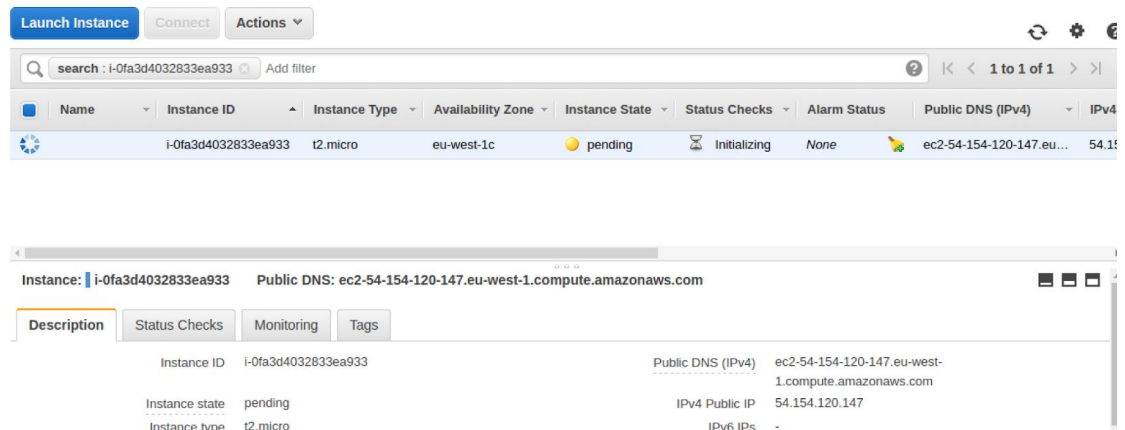
Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. [Find out](#) how to connect to your instances.

Here are some helpful resources to get you started

- [How to connect to your Linux instance](#)
- [Amazon EC2: User Guide](#)
- [Learn about AWS Free Usage Tier](#)
- [Amazon EC2: Discussion Forum](#)

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 if 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:

Instance State	Status Checks	Alarm Status	Public DNS	Public IP
running	2/2 checks ...	None	ec2-52-30-233-95.eu-w...	52.30.233.95

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/oxclonn.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
SHA256:7Gh0akN9Pj3vWAegV0uYhPVI9qqVEe9RlNM0wcut01E.
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

@ packages can be updated.
@ 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.
```

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:

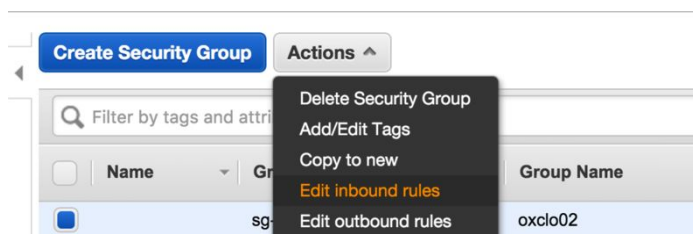
a. curl <http://localhost>

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 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 **Actions -> Edit Inbound rules**

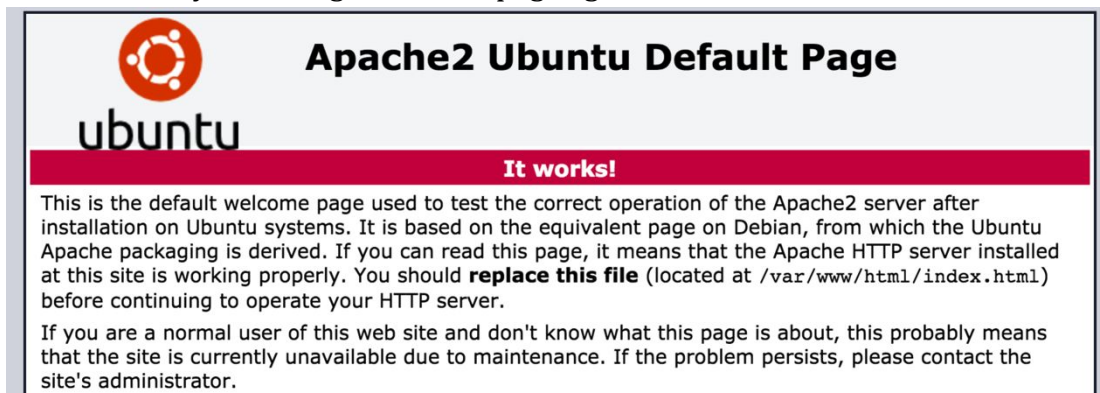


45. Click **Add Rule**

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

47. Click **Save**.

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



49. Congratulations!

PART C – Using the AWS Command Line

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

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

a. Type:

```
sudo pip install awscli
```

you should see log ending like:

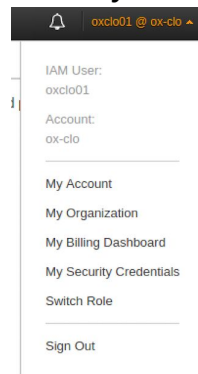
```
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in
/usr/local/lib/python2.7/dist-packages (from
botocore==1.5.80->awscli) (0.9.3)
Requirement already satisfied: pyasn1>=0.1.3 in
/usr/local/lib/python2.7/dist-packages (from
rsa<=3.5.0,>=3.1.2->awscli) (0.2.3)
Requirement already satisfied: six>=1.5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

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

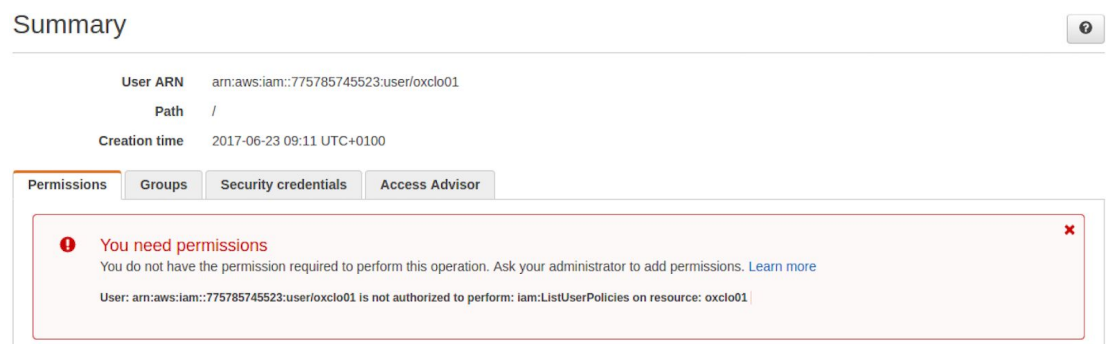
```
changing mode of /usr/local/bin/rst2s5.py to 755
changing mode of /usr/local/bin/rst2xetex.py to 755
changing mode of /usr/local/bin/rst2man.py to 755
changing mode of /usr/local/bin/rst2html.py to 755
Successfully installed awscli docutils boto3 rsa
jmespath python-dateutil pyasn1
Cleaning up...
```

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

- 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.
- Go to the AWS Console
- In the top right corner, click on your username, then choose **My Security Credentials**:



- In the left hand menu choose **Users**
- Click on your own userid
- You should see something like



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

Create access key

✓ Success

This is the **only** time that the secret access keys can be viewed or downloaded. You cannot recover them later. However, you can create new access keys at any time.

Download .csv file

Access key ID	Secret access key
AKIAIR34DT2HFSW73RQQ	***** Show

Close

- g. Click **Download .csv file** and then **Save**
- h. You can also click Show and then copy and paste these two token identifiers into a new text file

Create access key

✓ Success

This is the **only** time that the secret access keys can be viewed or downloaded. You cannot recover them later. However, you can create new access keys at any time.

Download .csv file

Access key ID	Secret access key
AKIAITJOR7ZIBCVCD5PA	GreasluQ9j3PzFD6uJ+HNACnfZimwOUyRo92RIP Hide

Close

- i. 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. 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.*

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.