

# 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>

Account: ox-clo

User Name: oxclo02

Password:

I have an MFA Token (more info)

[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:

AWS Management Console

AWS services

All services

---

**Build a solution**  
Get started with simple wizards and automated workflows.

<b>Launch a virtual machine</b> With EC2 2-3 minutes 	<b>Build a web app</b> With Elastic Beanstalk 6 minutes 	<b>Build using virtual servers</b> With Lightsail 1-2 minutes 	<b>Register a domain</b> With Route 53 3 minutes 
<b>Connect an IoT device</b> With AWS IoT 5 minutes 	<b>Start migrating to AWS</b> With AWS MGN 1-2 minutes 	<b>Start a development project</b> With CodeStar 5 minutes 	<b>Deploy a serverless microservice</b> With Lambda, API Gateway 2 minutes 

▶ See more

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

## 8. Expand All Services:

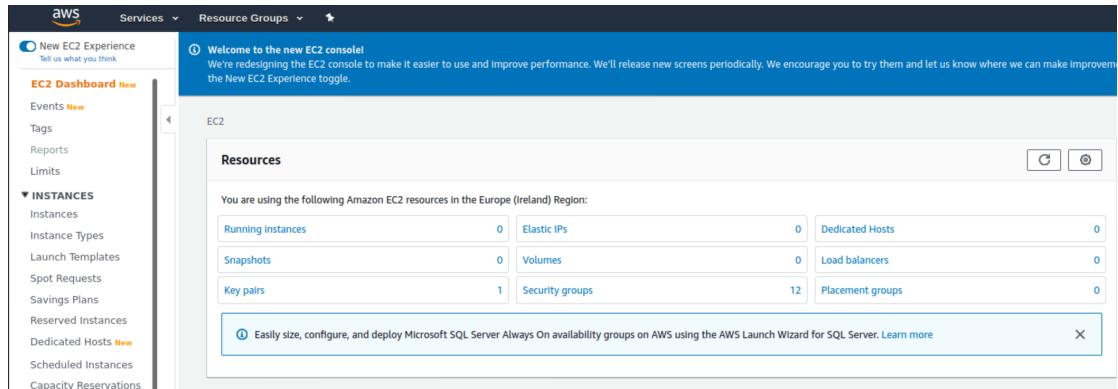
The screenshot shows the AWS Management Console with the 'AWS services' page. The left sidebar has a tree view with 'All services' expanded, showing categories like Compute, Containers, Storage, Database, and others. Each category has a list of specific services. For example, 'Compute' includes EC2, Lightsail, Lambda, Batch, Elastic Beanstalk, Serverless Application Repository, AWS Outposts, EC2 Image Builder, and AWS App Runner. The main area lists services under each category, such as SageMaker, Support, Managed Services, Activate for Startups, RoboMaker, Blockchain, Amazon Managed Blockchain, Ground Station, Quantum Technologies, Amazon Braket, AWS Organizations, CloudWatch, AWS Auto Scaling, CloudFormation, CloudTrail, Config, OpsWorks, Service Catalog, Systems Manager, AWS AppConfig, Trusted Advisor, Control Tower, AWS License Manager, AWS Well-Architected Tool, Machine Learning, Amazon SageMaker, Amazon Augmented AI, Amazon CodeGuru, Amazon DevOps Guru, Amazon Comprehend, Amazon Forecast, Amazon Fraud Detector, Amazon Kendra, Amazon Lex, Amazon Personalize, Amazon Polly, Amazon Rekognition, Amazon Textract, Amazon Transcribe, Amazon Translate, AWS DeepComposer, AWS DeepLens, AWS DeepRacer, AWS Panorama, Amazon Monitron, Amazon HealthLake, Amazon Lookout for Vision, Amazon Lookout for Equipment, Amazon Lookout for Metrics, Athena, Amazon Redshift, EMR, CloudSearch, and Elasticsearch Service. There are also sections for Customer Enablement, Robotics, Satellite, Analytics, and various cost management, front-end web/mobile, AR/VR, application integration, and business applications 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**

**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)**

#### 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.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes
Name		oxclo01		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

**Add another tag** (Up to 50 tags maximum)

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

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

### 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 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: launch-wizard-3 created 2020-06-25T15:35:04.567+01:00

Type	Protocol	Port Range
SSH	TCP	22

Add Rule

**⚠ Warning**  
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses

## 21. Click Review and Launch

You should see something very like this:

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

**⚠ Improve your instances' security. Your security group, oxclo01-sg, is open to the world.**  
Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.  
You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

**AMI Details** [Edit AMI](#)

<input checked="" type="radio"/> Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-0a8e758f5e873d1c1
Free tier eligible
Root Device Type: ebs Virtualization type: hvm

**Instance Type** [Edit instance type](#)

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

**Security Groups** [Edit security groups](#)

Security group name: oxclo01-sg
Description: launch-wizard-1 created 2021-07-11T18:26:26.475+01:00
Type: SSH Protocol: TCP Port Range: 22 Source: 0.0.0.0/0

**Instance Details** [Edit instance details](#)

**Storage** [Edit storage](#)

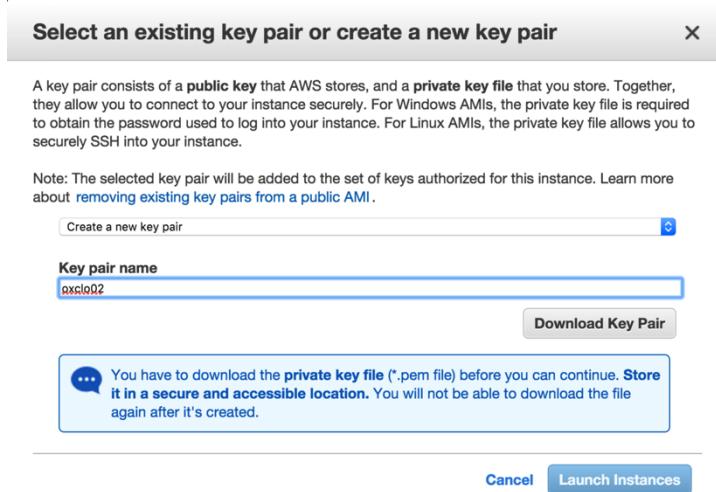
**Tags** [Edit tags](#)

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



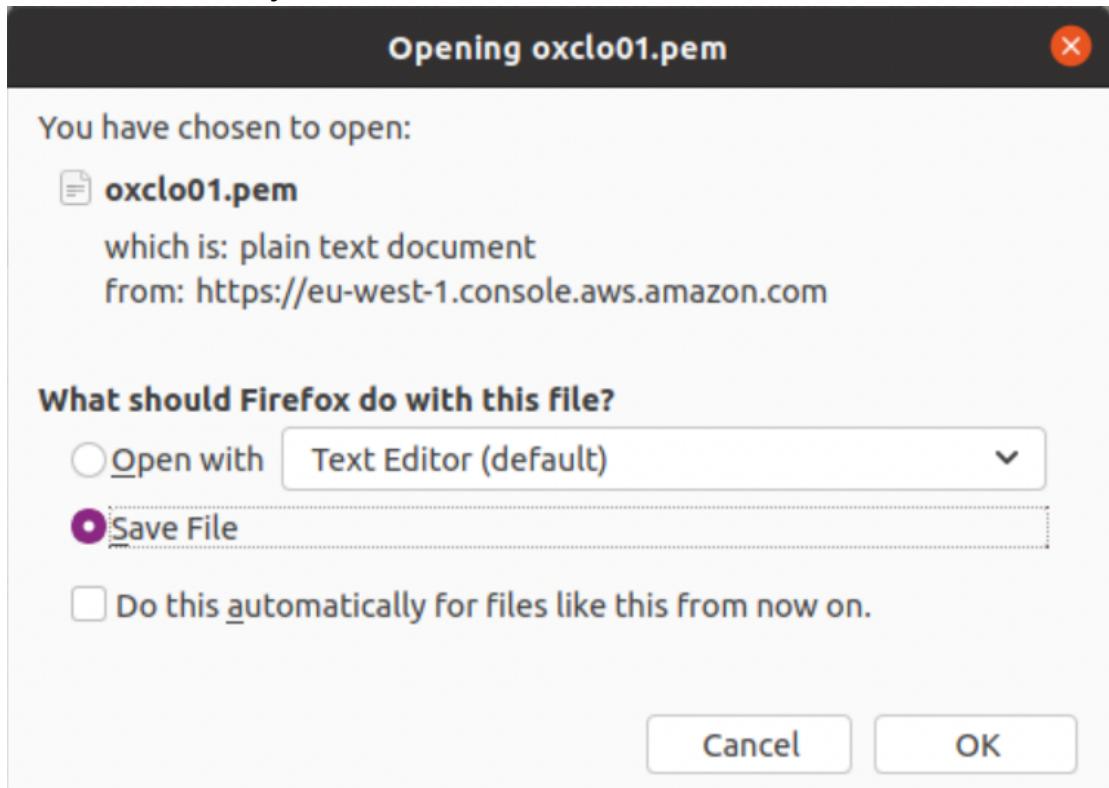
## 22. Click **Launch**

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**.



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:

## 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 after they are in the running state, so you can stop or terminate your instances.

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](#)
- [Learn about AWS Free Usage Tier](#)
- [Amazon EC2: User Guide](#)
- [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:

The screenshot shows the AWS EC2 Instances dashboard. On the left, there's a sidebar with links for EC2 Dashboard, Events, Tags, Reports, Limits, and various EC2 services like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, and Capacity Reservations. The main area has tabs for 'Launch Instance', 'Connect', and 'Actions'. A search bar at the top says 'search: i-0ab9677a06459c'. Below it is a table with one row for 'oxclo01'. The columns include Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), IPv4 Public IP, IPv6 IPs, Key Name, Monitoring, and Launch Time. The instance details are: Name: oxclo01, Instance ID: i-0ab9677a06459c, Instance Type: t2.micro, Availability Zone: eu-west-1c, Instance State: pending, Status Checks: Initializing, Alarm Status: None, Public DNS (IPv4): ec2-3-250-189-196.eu..., IPv4 Public IP: 3.250.169.196, Key Name: oxclo01, Monitoring: disabled, Launch Time: June 25, 2020 01:11:11.

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:

The screenshot shows the AWS CloudWatch Metrics dashboard. At the top, there are dropdown menus for 'Instance State', 'Status Checks', 'Alarm Status', 'Public DNS', and 'Public IP'. Below these are two rows of status indicators. The first row shows 'running' with a green dot and '2/2 checks ...' with a green checkmark. The second row shows 'None' with a yellow question mark icon and 'ec2-52-30-233-95.eu-w...' followed by '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/oxclo**.pem ubuntu@www.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
SHA256:7GhOakN9Pj3vWAegV0uYhPVI9qqVEe9RlNM0wcut01E.
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**

The screenshot shows the AWS Management Console interface for managing security groups. The 'Inbound rules' tab is selected. A single rule is listed:

Name	Security group rule...	IP version	Type	Protocol	Port range	Source
-	sgr-073fb506e6f46ec98	IPv4	SSH	TCP	22	0.0.0.0/0

45. You should see:

The screenshot shows the 'Edit inbound rules' dialog for a specific security group. It displays a single rule:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-073fb506e6f46ec98	SSH	TCP	22	Custom	0.0.0.0/0

Buttons at the bottom include 'Add rule', 'Cancel', 'Preview changes', and 'Save rules'.

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:

The screenshot shows the Apache2 Ubuntu Default Page. The page features the Ubuntu logo and the text "Apache2 Ubuntu Default Page". A red banner at the top right says "It works!". Below the banner, there is descriptive text about the default welcome page and instructions for replacing the index file. At the bottom, there is a note about the site being unavailable due to maintenance.

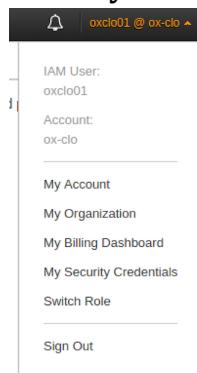
50. Congratulations!

## 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:

A screenshot of the AWS IAM user configuration page for the user 'oxclo01'. The left sidebar shows navigation options like 'Identity and Access Management (IAM)', 'Dashboard', 'Access management', 'Groups', 'Users', 'Roles', 'Policies', 'Identity providers', 'Account settings', 'Access reports', 'Access analyzer', 'Archive rules', 'Analyzers', 'Settings', 'Credential report', 'Organization activity', and 'Service control policies (SCPs)'. The main content area shows the user's ARN (User ARN: N/A), AWS account ID (775785745523), and Canonical user ID (for Amazon S3: 7655ad7edd...). It also shows the 'AWS IAM credentials' tab selected, which displays a 'Password for console access' section with a 'Change password' button, an 'Access keys for CLI, SDK, & API access' section with a 'Create access key' button, and a 'Multi-factor authentication (MFA)' section with an 'Assign MFA device' button. A red box highlights the 'Create access key' button. A message at the bottom states: 'You do not have active access keys.'

Click on **Create Access Key**.

A screenshot of a confirmation dialog box. It contains the text: 'You do not have active access keys.' and 'You need permissions'. It also includes the message: 'You do not have the permission required to perform this operation. Ask your administrator to add permissions.' and the error message: 'User: am:aws:iam::775785745523:user:oxclo01 is not authorized to perform: iam>ListMFADevices on resource: oxclo01'.

You should 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

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

## Create access key

 Your new access key is now available.

**This is the only time that the secret access key can be viewed or downloaded.**

You cannot recover it later. However, you can create new access keys at any time.

 Download .csv file

Access key ID AKIA3JIDPBRZ2UUYZ2VI 

Secret access key w2wMi9DyTmxYFILG7/p59AtWfocjIoxLhFhhEi0   
[Hide secret access key](#)

 Close

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.