

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.
2. The course is also providing time and resources on the Amazon AWS/EC2 cloud for the duration of the course. You may prefer to use your own Amazon AWS account instead.
 - a. If you wish to do so, there are instructions here:
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/get-set-up-for-amazon-ec2.html>

New users to Amazon get 750 hours of free usage for small (t2.micro) instances.
3. If you wish to use the provided account, continue here.
4. Open up a browser window and navigate to
<https://ox-clo.signin.aws.amazon.com/console>



© Paul Fremantle 2015. Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See <http://creativecommons.org/licenses/by-sa/3.0/>

Account: ox-clo

User Name: oxclo02

Password:

I have an MFA Token (more info)

Sign In

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

Hint: make a bookmark for that URL!



© Paul Fremantle 2015. Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See <http://creativecommons.org/licenses/by-sa/3.0/>

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:

The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with tabs for AWS, Services, and Edit. Below the navigation bar, the main content area is titled "Amazon Web Services". It features a grid of service icons and names. On the left, under "Compute", are EC2, EC2 Container Service, Elastic Beanstalk, and Lambda. Under "Storage & Content Delivery" are S3, CloudFront, and various storage services like Elastic File System, Glacier, Import/Export Snowball, and Storage Gateway. Under "Database" are RDS, DynamoDB, ElastiCache, and Redshift. Under "Networking" are VPC and Direct Connect. In the center, there are sections for Developer Tools (CodeCommit, CodeDeploy, CodePipeline), Management Tools (CloudWatch, CloudFormation, CloudTrail, Config, OpsWorks, Service Catalog, Trusted Advisor), Security & Identity (Identity & Access Management, Directory Service, Inspector, WAF), Analytics (EMR, Data Pipeline), and Internet of Things (AWS IoT). To the right, there are sections for Resource Groups, Additional Resources (Getting Started, AWS Console Mobile App, AWS Marketplace, AWS re:Invent Announcements), and Service Health. A sidebar on the left lists various AWS services like Compute, Storage, Database, Networking, and more. The status bar at the bottom indicates "oxclo02 @ ox-clo Oregon" and the time "19:05".

7. In the top right corner click on Oregon and change to one of EU (Ireland), EU (Frankfurt) or US East (N. Virginia).



8. Now click on the top left EC2



9. 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.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Spot Requests, Reserved Instances, Commands, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Network & Security, Security Groups, Elastic IPs, Placement Groups, and Key Pairs. The main area has sections for Resources (listing 0 Running Instances, 0 Elastic IPs, 1 Volumes, 2 Snapshots, 1 Key Pairs, 0 Load Balancers, 0 Placement Groups, 2 Security Groups), Create Instance (with a 'Launch Instance' button), Service Health (EU West (Ireland) status: operating normally), and Scheduled Events (No events). On the right, there's an Account Attributes section (Supported Platforms: VPC, Default VPC: vpc-42fb9527) and an Additional Information section (Getting Started Guide, Documentation, All EC2 Resources, Forums, Pricing, Contact Us). A sidebar on the right also lists AWS Marketplace products like Tableau Server (10 users).

10. Click on the blue button: Launch Instance

11. Choose “Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-47a23a30”



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

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

Next: Configure Instance Details

14. Click **Next: Add Storage**



15. Click **Next: Tag Instance**

16. In the Tag Instance screen, give your instance a name that is the same as your userid:

Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(127 characters maximum)	Value	(255 characters maximum)
Name	oxclo02		

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

18. 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, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group.

Assign a security group: Create a new security group

Select an existing security group

Security group name:

oxclo02

Description:

launch-wizard-1 created 2015-11-16T09:27:30.852+00:00

Type	Protocol	Port Range
SSH	TCP	22

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.



© Paul Fremantle 2015. Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See <http://creativecommons.org/licenses/by-sa/3.0/>

19. Click Review and Launch

You should see something very like this:

AMI Details

Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-47a23a30
Free tier eligible Root Device Type: ebs Virtualization type: hvm

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

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

Tags

Key	Value
Name	oxclo02

Buttons: Edit AMI, Edit instance type, Edit security groups, Edit instance details, Edit storage, Edit tags, Cancel, Previous, Launch.

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**.

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

oxclo02

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

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**

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

25. 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: EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES (which is expanded), Instances, Spot Requests, and Reserved Instances. The main area has tabs: Launch Instance, Connect, Actions. Below is a search bar with 'search : i-a475401d' and an 'Add filter' button. A table lists instances with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, and Status Checks. One row is shown: oxclo02, i-a475401d, t2.micro, eu-west-1b, running, Initializing.

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

27. 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:

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 is a table with one row: running, 2/2 checks ..., None, ec2-52-30-233-95.eu-w..., 52.30.233.95.

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/oxclonnn.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  
SHA256:7GhOakN9Pj3vWAegV0uYhPVI9qqVEe9RlNM0wcut01E.  
Are you sure you want to continue connecting (yes/no)?
```

Type **yes** and hit Enter.

You will see something like:

```
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.13.0-48-generic x86_64)  
  
 * Documentation: https://help.ubuntu.com/  
  
 System information as of Mon Nov 16 09:50:28 UTC 2015  
  
 System load: 0.32           Memory usage: 5%   Processes:      82  
 Usage of /: 9.8% of 7.74GB   Swap usage:  0%   Users logged in: 0  
  
 Graph this data and manage this system at:  
   https://landscape.canonical.com/  
  
 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.  
  
 ubuntu@ip-172-31-23-34:~$
```

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

PART B – Running a Web Server

35. In the SSH shell type:

```
sudo apt-get 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
```



© Paul Fremantle 2015. Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See <http://creativecommons.org/licenses/by-sa/3.0/>

36. Now type:

```
sudo apt-get 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 pop up.

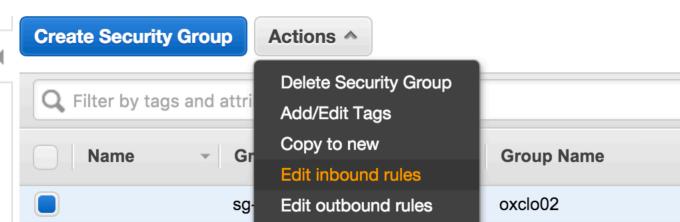
40. Now try browsing the server from your local machine. Find the Public IP address or Public DNS name and use that in a browser window.

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.



© Paul Fremantle 2015. Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See <http://creativecommons.org/licenses/by-sa/3.0/>

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



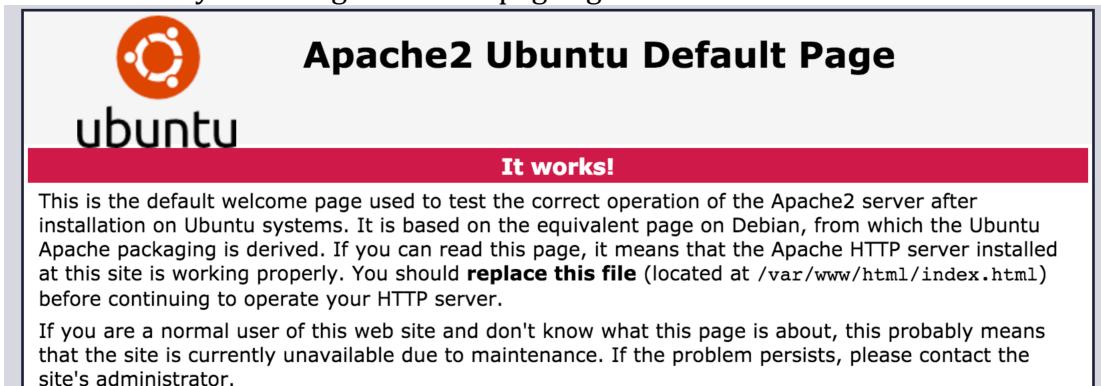
43. Click **Add Rule**

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

45. Add another rule to allow HTTPS as well.

46. Click **Save**.

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



48. Congratulations!

PART C – Using the AWS Command Line

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

50. 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

b. You will probably be prompted for oxclo's password. It is "oxclo"



c. You should see log ending like:

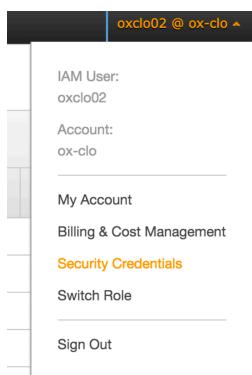
```
changing mode of /usr/local/bin/rst2s5.py to 755
changing mode of /usr/local/bin/rst2xetex.py to 755
changing mode of /usr/local/bin/rst2man.py to 755
changing mode of /usr/local/bin/rst2html.py to 755
Successfully installed awscli docutils botocore rsa
jmespath python-dateutil pyasn1
Cleaning up...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

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

52. 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.

53. Go to the AWS Console

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

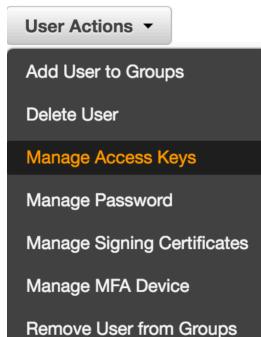


55. In the left hand menu choose **Users**

56. Ignore the lines that say things like:

We encountered the following errors while processing your request:
User: arn:aws:iam::775785745523:user/oxclo02 is not authorized to perform: iam>ListGroupsForUser on resource: djcomlab

a. Select your own userid, then click **User Actions -> Manage Access Keys**



- b. You will either see:

Manage Access Keys

Use access keys to make secure REST or Query protocol requests to any AWS service API.

This user does not currently have any access keys.

Note: For your protection, you should never share your secret keys with anyone. In addition, industry best practice recommends frequent key rotation.

› [Learn more about Access Keys](#)

[Cancel](#)

[Create Access Key](#)

Or

Manage Access Keys

Use access keys to make secure REST or Query protocol requests to any AWS service API.

Access Key ID	Created	Last Used	Last Used Service	Last Used Region	Status
AKIAJKBBQLH3ACPPXIJQ	2015-11-16 12:27 UTC	N/A	N/A	N/A	Active (Make Inactive Delete)

Note: For your protection, you should never share your secret keys with anyone. In addition, industry best practice recommends frequent key rotation.

› [Learn more about Access Keys](#)

[Cancel](#)

[Create Access Key](#)

- c. If you see the second screen then Delete the Access Key, and then go back and you will see the first screen.

- d. Click **Create Access Key**. You will see:

Manage Access Keys

Your access key has been created successfully.

This is the last time these User security credentials will be available for download.

You can manage and recreate these credentials any time.

► [Show User Security Credentials](#)

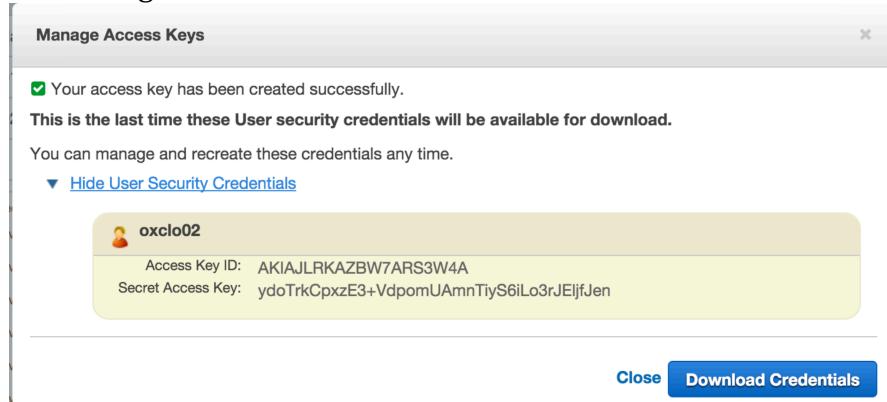
[Close](#)

[Download Credentials](#)

- e. Click Download Credentials.



- f. Also click on Show User Security Credentials. You will see something like this:



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

58. 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 browser screen (cut and paste)
- b. Do the same for the Secret Access Key.
- c. For the region choose whichever region you chose earlier, using these codes:
 - i. Ireland: **eu-west-1**
 - ii. Frankfurt: **eu-central-1**
 - iii. N. Virginia: **us-east-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.

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



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



© Paul Fremantle 2015. Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See <http://creativecommons.org/licenses/by-sa/3.0/>

61. Now use the AWS CLI to terminate:

Replacing the instance ID with your own, type:

```
aws ec2 terminate-instances --instance-ids i-a475401d
```

62. You should see log like:

```
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-a475401d",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

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

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



© Paul Fremantle 2015. Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See <http://creativecommons.org/licenses/by-sa/3.0/>