Working with AWS: Initial exploration and configuration

A. Launch an Amazon EC2 instance

1. Log into RosettaHUB using the user name and password provided by RosettaHUB.
2. Click the "Go to AWS Console" button.

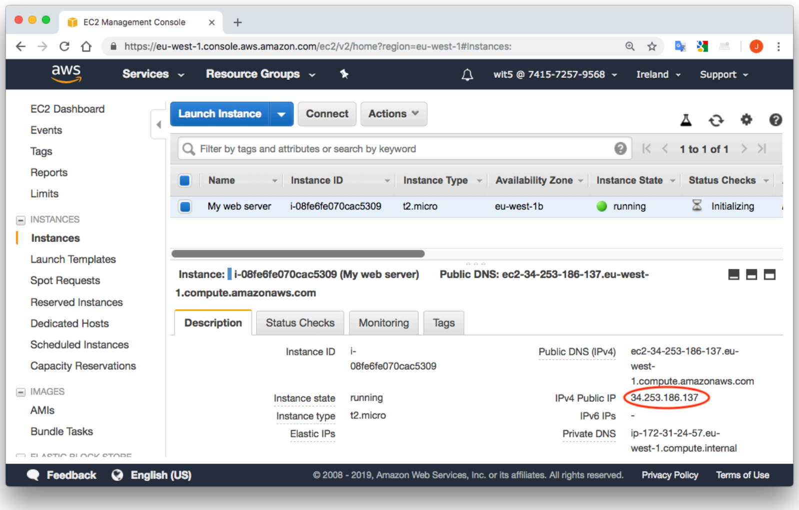
   Go To AWS Console

   The AWS management console should open in a new browser tab. You may need to enable pop-ups from rosettahub.com.
3. Verify that the region setting on the top right indicates *Ireland*. Click on **Services** near the top left and select **EC2** from the *Compute* category.
4. This will bring you to the EC2 Dashboard. From here, click on **Launch Instance**.

   Launch Instance
5. The *Step 1: Choose an Amazon Machine Image (AMI)* page displays a list of basic configurations, called *Amazon Machine Images (AMIs)*, that serve as templates for your instance. **Select** the first one listed, which runs *Amazon Linux 2*.
6. On the *Step 2: Choose an Instance Type* page, leave everything as it is (t2.micro) and click on **Next: Configure Instance Details** (not *Review and Launch* just yet).
7. On the *Step 3: Configure Instance Details* page, again leave everything as it is and click on **Next: Add Storage**.
8. On the *Step 4: Add Storage* page, again leave everything as it is and click on **Next: Add Tags**.
9. N.B. On the *Step 5: Add Tags* page, click **Add Tag** and enter "Name" as the *Key* and "My web server" as the *Value*. It is good practice to tag instances to help identify them. Then click on **Next: Configure Security Group.**
10. On the *Step 6: Configure Security Group* page, click **Add Rule** and enter the following:

HTTP	TCP	80	Anywhere	0.0.0.0/0, ::/0	Allow access to web server

11. Click **Review and Launch**.
12. On the *Step 7: Review Instance Launch* page, click **Launch**. For now, you can ignore the warning about security.
13. You will now see a dialogue: "Select an existing key pair or create a new key pair". Choose **Create a new key pair** from the drop-down list and enter a name for your key pair (e.g. *jbloggs_key*). Next click **Download Key Pair** and a file with a .pem extension will be downloaded to your local machine. You will need this file to connect to your instance. Click on **Launch Instances**.
14. Click the **View Instances** button at the bottom right of the *Launch Status* page. You should now be able to see the attributes of your newly-launched instance. Take note of its *public IP address*, which should become available after a few moments when the instance has started up and the *Instance State* changes to *running*.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
My web server	i-08fe6fe070cac5309	t2.micro	eu-west-1b	running	Initializing

Instance: i-08fe6fe070cac5309 (My web server) Public DNS: ec2-34-253-186-137.eu-west-1.compute.amazonaws.com

Instance ID	i-08fe6fe070cac5309	Public DNS (IPv4)	ec2-34-253-186-137.eu-west-1.compute.amazonaws.com
Instance state	running	IPv4 Public IP	34.253.186.137
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-24-57.eu-west-1.compute.internal

B. Connect to your EC2 instance and install a web server

15. Open your local Linux (virtual) machine and start a Terminal.
16. Locate the key file (with the .pem extension) that you downloaded when creating your EC2 instance. Copy[1] this to your Linux virtual machine and into your home directory.
17. In the terminal window, enter the following command (*N.B. on your **local** machine*):

```
chmod 400 key_pair.pem
ssh -i key_pair.pem ec2-user@public_ip_address
```

(replacing *key_pair.pem* and *public_ip_address* with your key name and instance IP address respectively)

18. You will get a security warning the first time you log into your instance to flag that you haven't logged into this machine before. This is normal; enter 'yes' to continue. You are now able to enter commands on your remote (cloud) instance.
19. Download and install the Apache web server with the command (*N.B. on **remote** machine via SSH*):

```
sudo yum install httpd -y
```

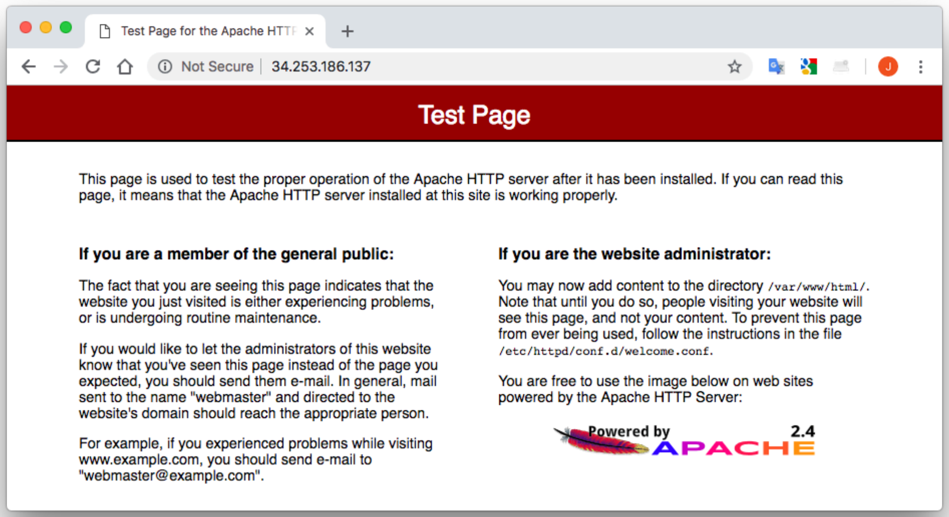
20. Enter the following command to have the system always start the web server following a reboot:

```
sudo systemctl enable httpd
```

21. Enter the following command to start the web server

```
sudo service httpd start
```

22. Test that the web server is working by pointing a web browser to
    http://public_ip_addr
    (replacing *public_ip_addr* with the address of your instance).
23. You should see a default home page for Apache on your instance. You can change this and/or add new web pages to your server by going to the directory **/var/www/html/** on the server.



Test Page

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:

[1] If using VMware, install *open-vm-tools* or *VMware Tools*, and then you can just drag and drop it with the mouse or copy/paste

C. Python, Boto3 and AWS CLI installation and set-up

This section assumes that you are running Ubuntu **locally – *this means all the commands here are to run on your local machine, not on an AWS EC2 instance*.**

24. First, verify that Python 3 is installed on your local machine (e.g. on Ubuntu VM). Just enter the "python3" command in a terminal window to launch an interactive interpreter:

```
[2]$ python3
[3]>>> print('Hello, World!')
>>> quit()
```

25. Boto is a Python interface to Amazon Web Services. You can find tutorials and the full API reference at https://boto3.readthedocs.io/en/latest/. You can use pip3 (package management system for Python 3) to install boto3.

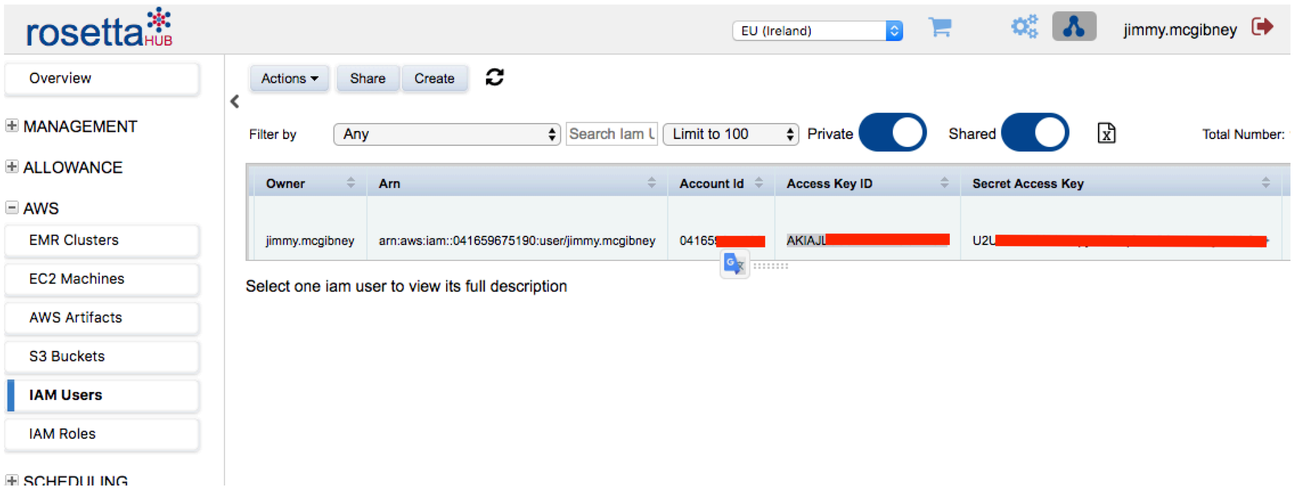
    On Ubuntu[4] enter:

```
$ sudo apt update
$ sudo apt install python3-pip
$ pip3 install boto3                # you can ignore warning about pip version
```

26. Next install the AWS Command Line Interface (CLI) tool (https://aws.amazon.com/cli/):

```
$pip3 install awscli                # on Ubuntu 18, it's:   sudo apt install awscli
```

Next, you'll need to configure your boto AWS credentials. To do this, you need your AWS Access Key ID and corresponding Secret Access Key. You can find these on the **RosettaHUB** console (not the AWS console). On the menu on the left, expand "AWS" and select "IAM Users".



Owner	Arn	Account Id	Access Key ID	Secret Access Key
jimmy.mcgibney	arn:aws:iam::041659675190:user/jimmy.mcgibney	04165[redacted]	AKIAJL[redacted]	U2U[redacted]

Select one iam user to view its full description

27. Now that we have a user and credentials, we can configure the scripting environment with the AWS CLI tool.

28. Back in the terminal, enter

```
$ aws configure
```

    You'll be prompted for the AWS access key ID, AWS secret access key, default region name, and default output format. Using the credentials from RosettaHUB, enter the access key ID and secret access key.

29. For the default region name, enter *eu-west-1*

[2] The $ sign here is shorthand for the Linux command prompt, which might be something like `jbloggs@ubuntu:~$` for you
[3] The >>> indicates the Python prompt. Python is an *interpreted* language, like bash in some respects.
[4] This will be a little different on other distributions or platforms.

30. Options for the default output format are text, JSON, and table. Enter “text” for now.

```
AWS Access Key ID [None]: AKIAJFUD72GXAN5SQRPA
AWS Secret Access Key [None]: LLL1tjMJpwNsCq23XRE3ZXLJhvYkjH2Df4UO9zzz
Default region name [None]: eu-west-1
Default output format [None]: text
```

31. Now that your environment is all configured, let’s run a quick test with the AWS CLI tool before moving on. In the shell, enter:

```
$ aws ec2 describe-instances
```

If you already have instances on your account, you’ll see the details of those instances. If not, you should see an empty response. If you see any errors, walk through the previous steps to see if anything was overlooked or entered incorrectly, particularly the access key ID and secret access key.

32. Now we can check if Boto3 is correctly using installed and you can used it to communicate with AWS from the Python3 interpreter

```
$ python3
>>> import boto3
>>> ec2 = boto3.resource('ec2')
>>> for inst in ec2.instances.all():
>>>     print (inst.id, inst.state)
i-00dc214270426a28e {'Name': 'running', 'Code': 80}
>>> quit()
```

Here you are getting an EC2 service resource object and using this to iterate across a collection of instance objects and print their IDs. You will only see a response (instance ID and state in this case) if you have some EC2 instances set up.