Pre-Course Exercise 2

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

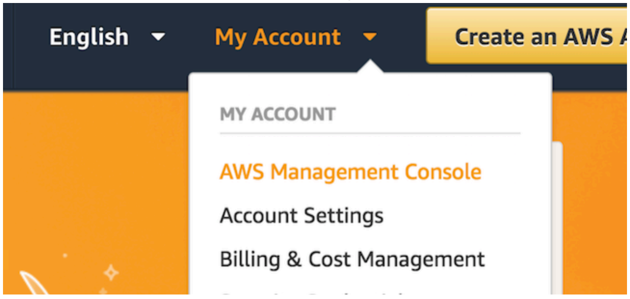
Software Requirements

- AWS CLI (to be installed during the exercise)

Part A: Starting an Instance from the Web Console.

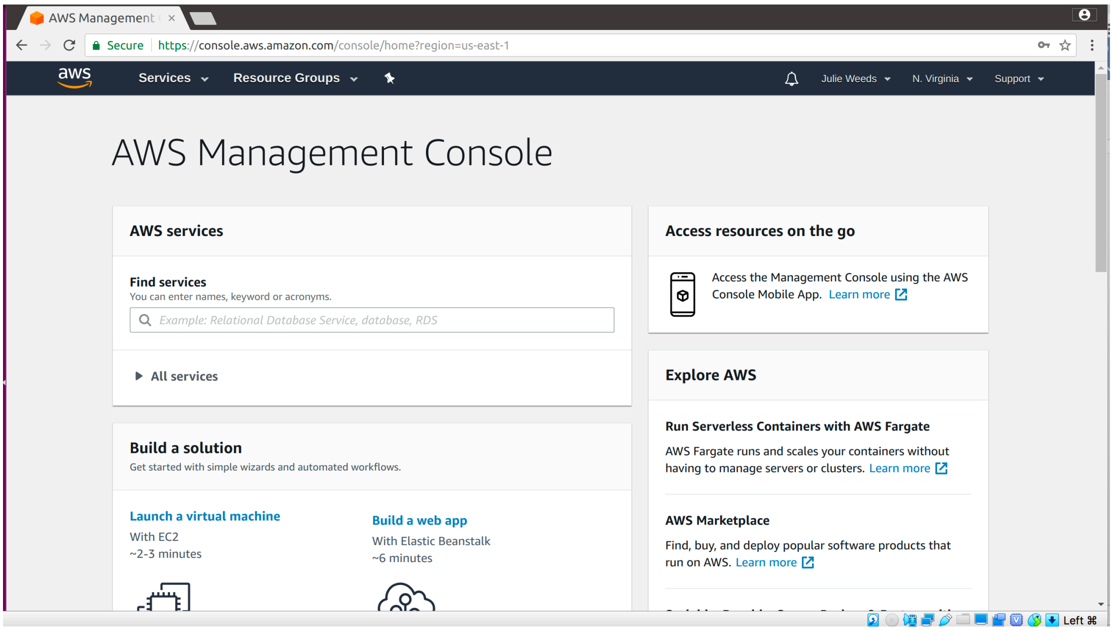
1. Open up a browser window and navigate to https://aws.amazon.com/

2. Click on the menu item My Account-> AWS Management Console



3. Log in with your credentials

4. You should see a screen like this:



AWS Management Console

AWS services

Find services
You can enter names, keyword or acronyms.
Example: Relational Database Service, database, RDS

▸ All services

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

Launch a virtual machine
With EC2
~2-3 minutes

Build a web app
With Elastic Beanstalk
~6 minutes

Access resources on the go

Access the Management Console using the AWS Console Mobile App. Learn more

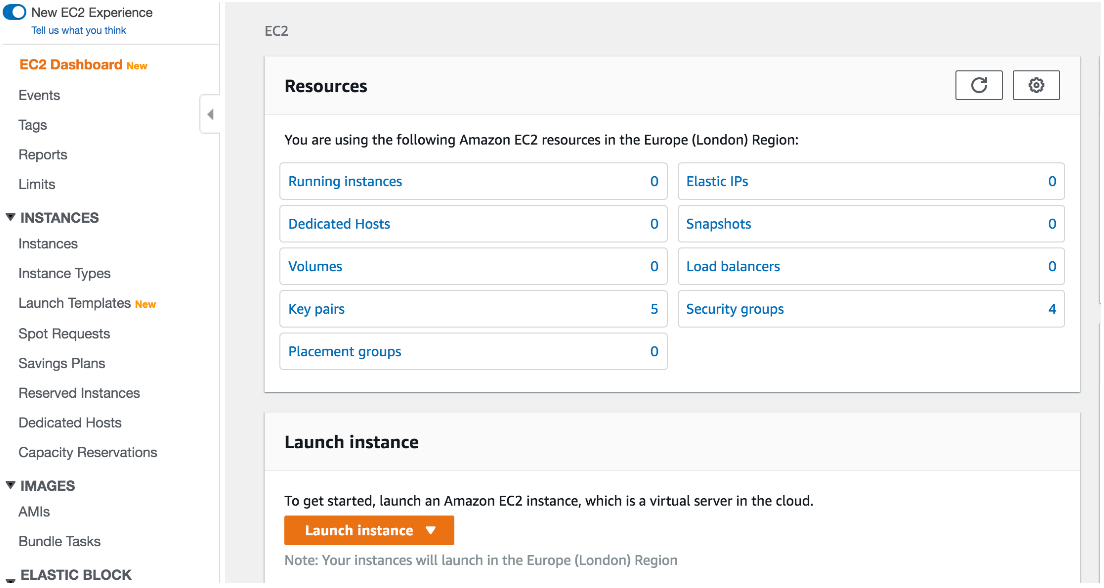
Explore AWS

Run Serverless Containers with AWS Fargate
AWS Fargate runs and scales your containers without having to manage servers or clusters. Learn more

AWS Marketplace
Find, buy, and deploy popular software products that run on AWS. Learn more

5. In the top right corner click on N. Virginia and change to **EU (London) (unless it is already on London!)**

6. Expand **All Services** and click on the link **EC2**



Resources

You are using the following Amazon EC2 resources in the Europe (London) Region:

Running instances	0	Elastic IPs	0
Dedicated Hosts	0	Snapshots	0
Volumes	0	Load balancers	0
Key pairs	5	Security groups	4
Placement groups	0		

Launch instance

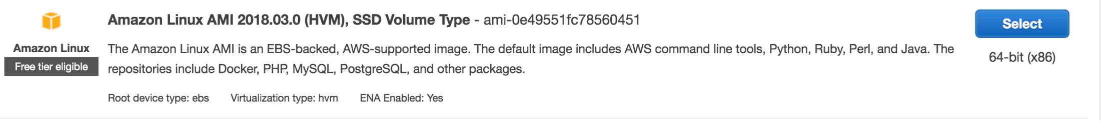
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance ▾

Note: Your instances will launch in the Europe (London) Region

7. Click on the orange button: Launch Instance

8. Choose "**Amazon Linux AMI 2018.03.0 (HVM) SSD Volume Type**"



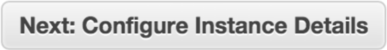
Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0e49551fc78560451

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes

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

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

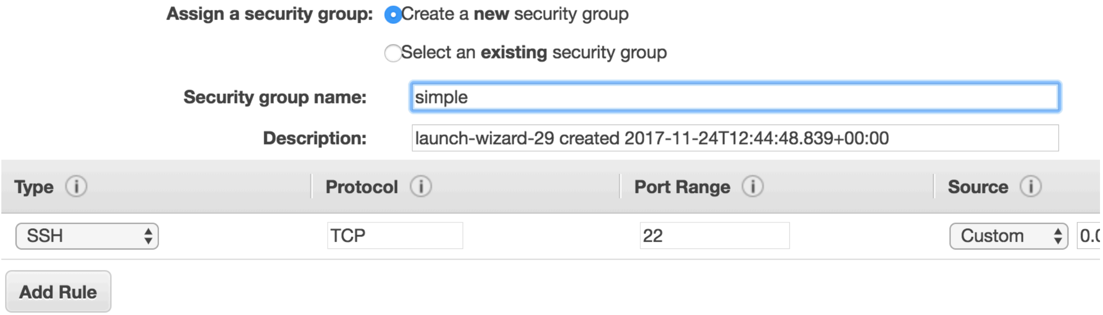


11. Click **Next: Add Storage**

12. Click **Next: Add Tags**

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

14. Change the name of the security group to **simple**



Assign a security group: ● Create a **new** security group

○ Select an **existing** security group

Security group name: simple

Description: launch-wizard-29 created 2017-11-24T12:44:48.839+00:00

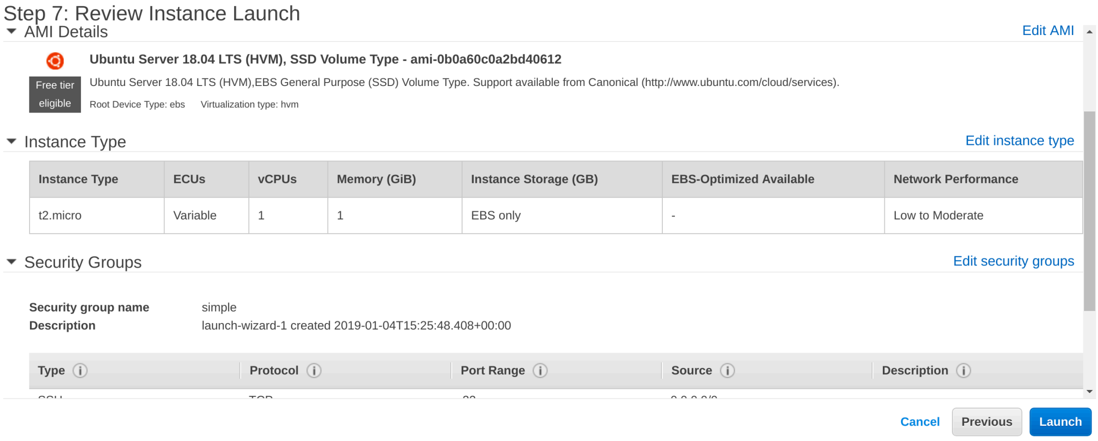
Type	Protocol	Port Range	Source
SSH	TCP	22	Custom 0.0

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 current network sends messages from different IPs or the same and therefore we will leave this as-is despite the warning.

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



Step 7: Review Instance Launch

▾ AMI Details

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0b0a60c0a2bd40612

Ubuntu Server 18.04 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

▾ 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

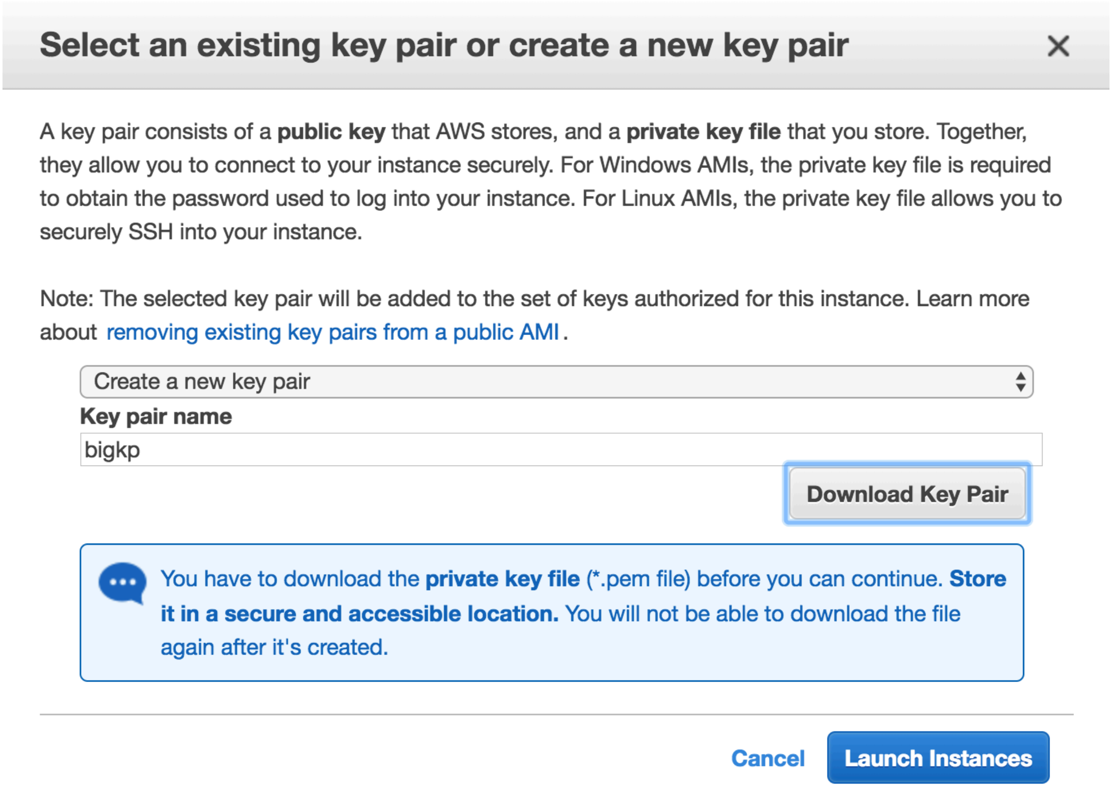
▾ Security Groups

Security group name simple
Description launch-wizard-1 created 2019-01-04T15:25:48.408+00:00

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
SSH	TCP	22	0.0.0.0/0	

16. Click **Launch**

17. 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

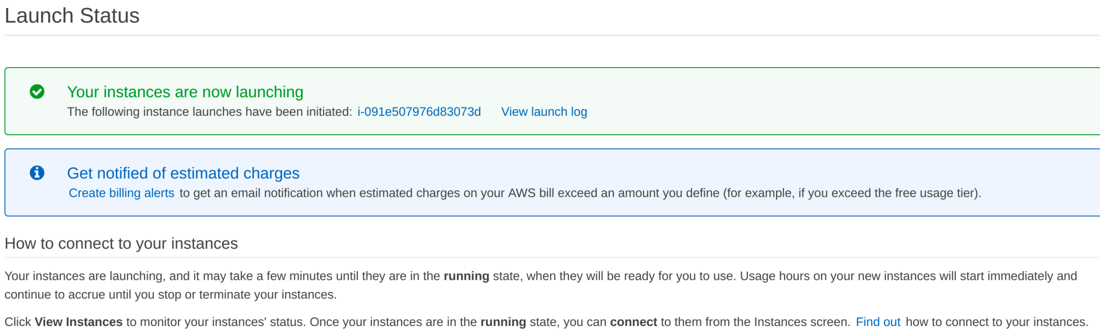
Key pair name
bigkp

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.

18. Use **bigkp** as the name of the keypair.

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

20. Click **Launch Instances**
    You should see something like:



Launch Status

Your instances are now launching
The following instance launches have been initiated: i-091e507976d83073d View launch log

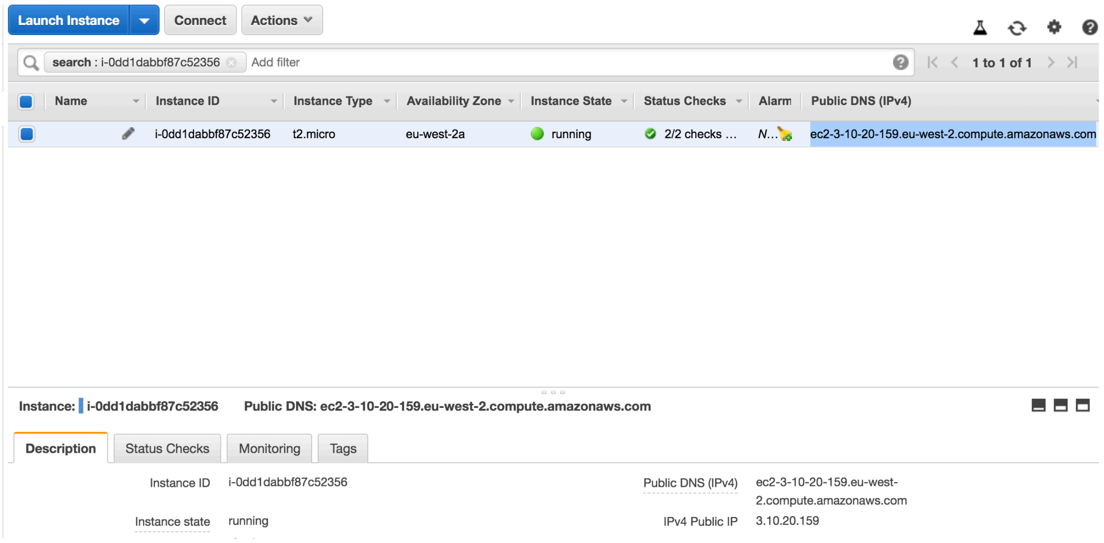
Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

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 and continue to accrue until you 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.

21. Click on the blue instance ID link (e.g. **i-091e507976d8307d** in the screenshot above)
    You will see a dashboard like:



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm	Public DNS (IPv4)
	i-0dd1dabbf87c52356	t2.micro	eu-west-2a	running	2/2 checks ...	N...	ec2-3-10-20-159.eu-west-2.compute.amazonaws.com

Instance: i-0dd1dabbf87c52356 Public DNS: ec2-3-10-20-159.eu-west-2.compute.amazonaws.com

Instance ID i-0dd1dabbf87c52356
Public DNS (IPv4) ec2-3-10-20-159.eu-west-2.compute.amazonaws.com
Instance state running
IPv4 Public IP 3.10.20.159

22. On your laptop, start a fresh terminal window (Ctrl-Alt-T, or find Terminal graphically)

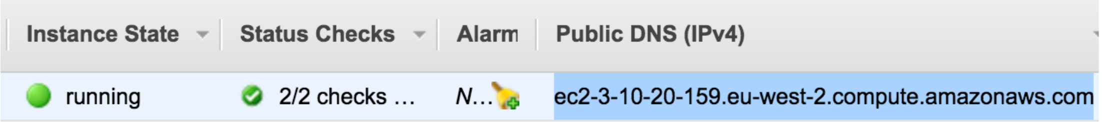
23. Check is there is already a ~/keys directory?

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

24. Copy your private key to the new directory:
    ```
    cp ~/Downloads/bigkp.pem ~/keys/
    ```

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

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



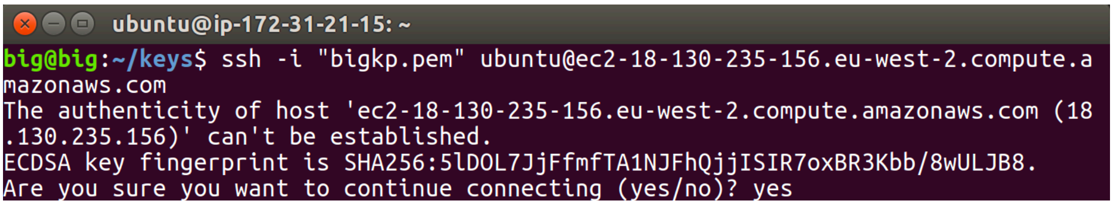
Instance State	Status Checks	Alarm	Public DNS (IPv4)
● running	✓ 2/2 checks …	N…	ec2-3-10-20-159.eu-west-2.compute.amazonaws.com

27. Copy the DNS server Address from the browser window (e.g. **ec2-3-10-20-159.eu-west-2.compute.amazonaws.com** in my case)

28. Try to SSH into the machine. Replace your key file name and the server address below!

```
ssh -i "~/keys/bigkp.pem" ec2-user@ec2-3-10-20-159.eu-west-
2.compute.amazonaws.com
```

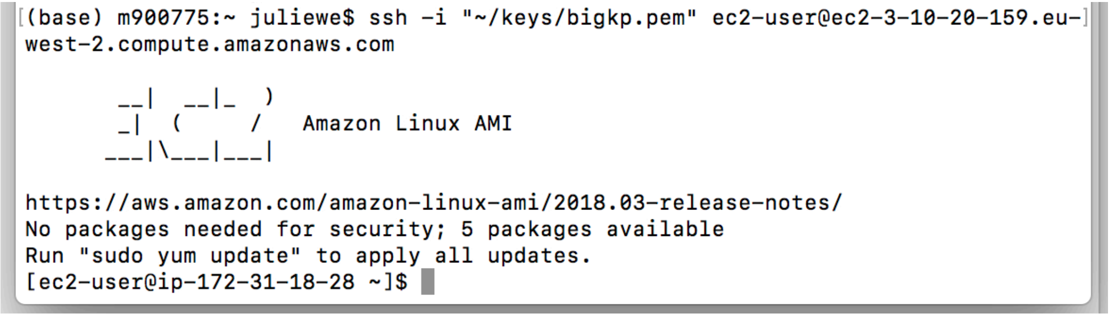
29. As this is the first time you are accessing this host, the key on the server side is not known. You should see something like:



```
big@big:~/keys$ ssh -i "bigkp.pem" ubuntu@ec2-18-130-235-156.eu-west-2.compute.a
mazonaws.com
The authenticity of host 'ec2-18-130-235-156.eu-west-2.compute.amazonaws.com (18
.130.235.156)' can't be established.
ECDSA key fingerprint is SHA256:5lDOL7JjFfmfTA1NJFhQjjISIR7oxBR3Kbb/8wULJB8.
Are you sure you want to continue connecting (yes/no)? yes
```

30. Type **yes** and hit Enter.

You will see something like:

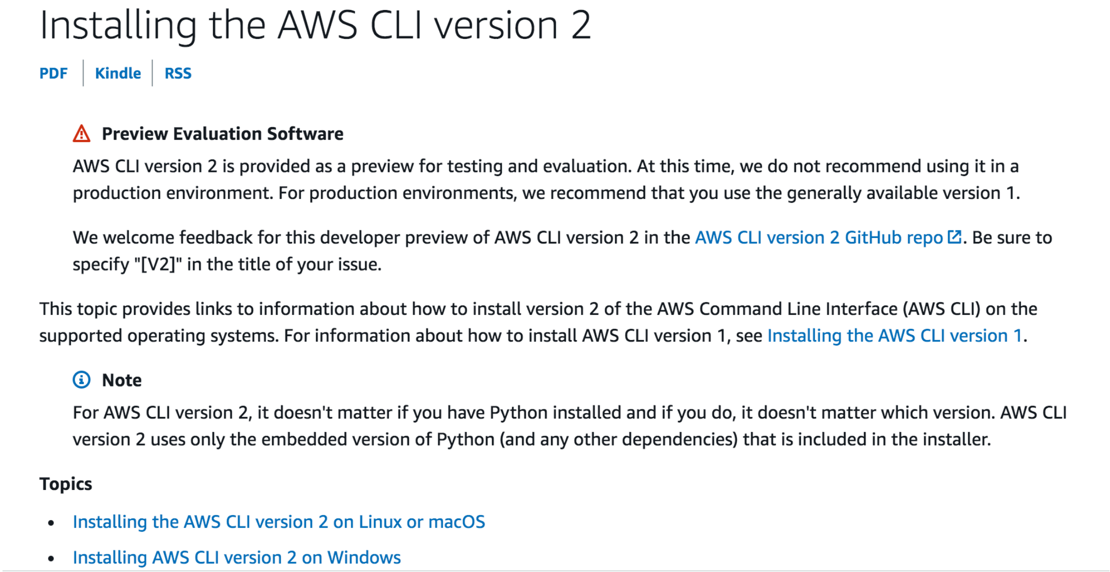


```
[(base) m900775:~ juliewe$ ssh -i "~/keys/bigkp.pem" ec2-user@ec2-3-10-20-159.eu-]
west-2.compute.amazonaws.com

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
No packages needed for security; 5 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-18-28 ~]$ 
```

31. Congratulations – you have a cloud instance running.

PART B – Using the AWS Command Line to terminate the instance

Follow the instructions on AWS to install the AWS Command line interface (version 2) appropriate for your operating system: https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html



Installing the AWS CLI version 2

PDF | Kindle | RSS

Preview Evaluation Software

AWS CLI version 2 is provided as a preview for testing and evaluation. At this time, we do not recommend using it in a production environment. For production environments, we recommend that you use the generally available version 1.

We welcome feedback for this developer preview of AWS CLI version 2 in the AWS CLI version 2 GitHub repo. Be sure to specify "[V2]" in the title of your issue.

This topic provides links to information about how to install version 2 of the AWS Command Line Interface (AWS CLI) on the supported operating systems. For information about how to install AWS CLI version 1, see Installing the AWS CLI version 1.

Note

For AWS CLI version 2, it doesn't matter if you have Python installed and if you do, it doesn't matter which version. AWS CLI version 2 uses only the embedded version of Python (and any other dependencies) that is included in the installer.

Topics

- Installing the AWS CLI version 2 on Linux or macOS
- Installing AWS CLI version 2 on Windows

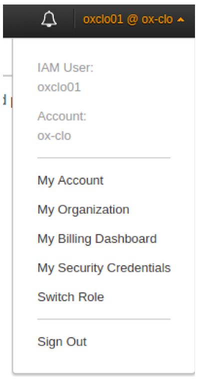
27. Open a fresh Terminal Window (*make sure you are not doing this on your cloud server by mistake!)*

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

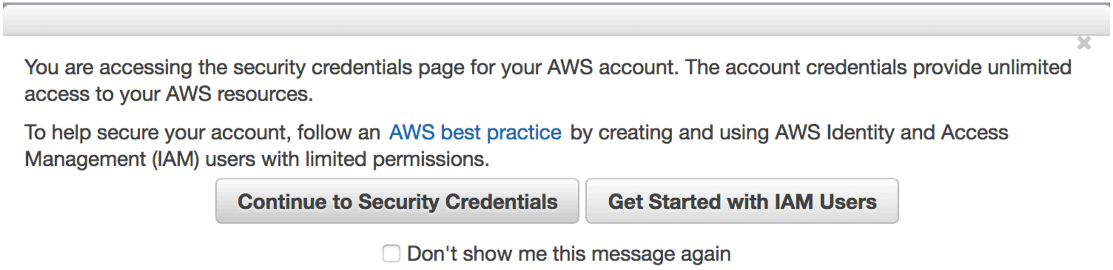
29. First you need to create an Access Key and Secret Key.

30. Go to the AWS Console

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



32. You will be warned as follows.
    Choose **Continue to Security Credentials.**

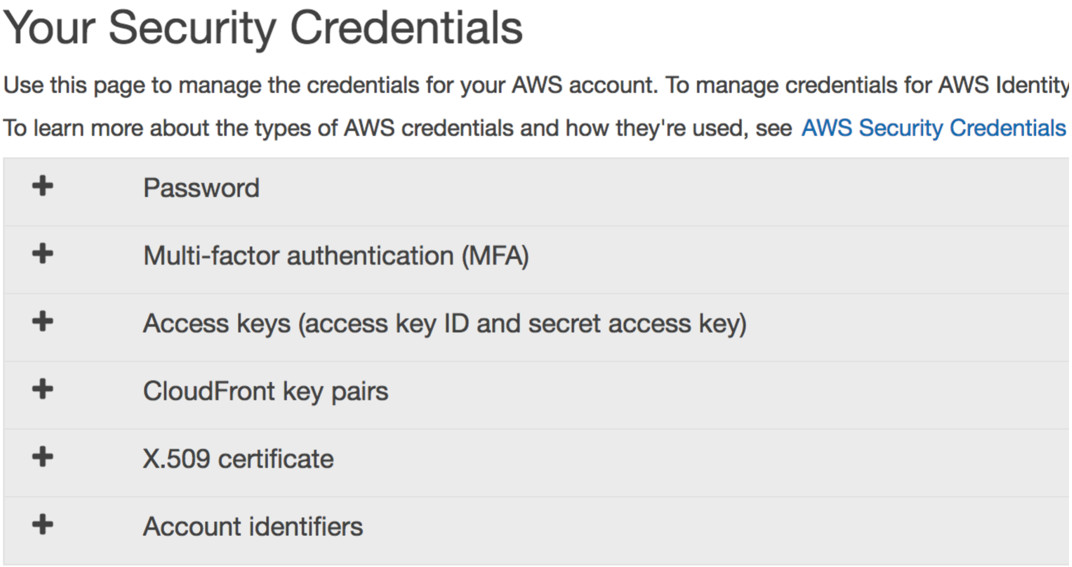


You are accessing the security credentials page for your AWS account. The account credentials provide unlimited access to your AWS resources.

To help secure your account, follow an AWS best practice by creating and using AWS Identity and Access Management (IAM) users with limited permissions.

Continue to Security Credentials
Get Started with IAM Users
□ Don't show me this message again

33. You should see:



Your Security Credentials

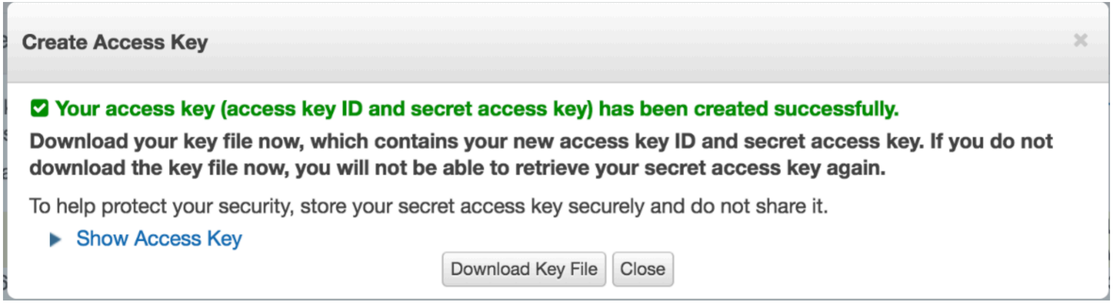
Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials

+ Password
+ Multi-factor authentication (MFA)
+ Access keys (access key ID and secret access key)
+ CloudFront key pairs
+ X.509 certificate
+ Account identifiers

34. Expand **AccessKeys**

35. Click **Create New Access Key.** You will see:



Create Access Key

☑ Your access key (access key ID and secret access key) has been created successfully.

Download your key file now, which contains your new access key ID and secret access key. If you do not download the key file now, you will not be able to retrieve your secret access key again.

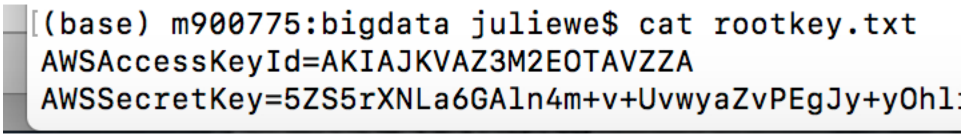
To help protect your security, store your secret access key securely and do not share it.

36. Click **Download Key File**
    It should download a file called **rootkey.csv**

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

38. In your terminal window, navigate to the directory where you have stored rootkey.txt and display its contents

```
cat rootkey.txt
```



```
(base) m900775:bigdata juliewe$ cat rootkey.txt
AWSAccessKeyId=AKIAJKVAZ3M2EOTAVZZA
AWSSecretKey=5ZS5rXNLa6GAln4m+v+UvwyaZvPEgJy+yOhl
```

39. Now we can use these keys to configure the AWS CLI. In a terminal window type:

```
aws2 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.

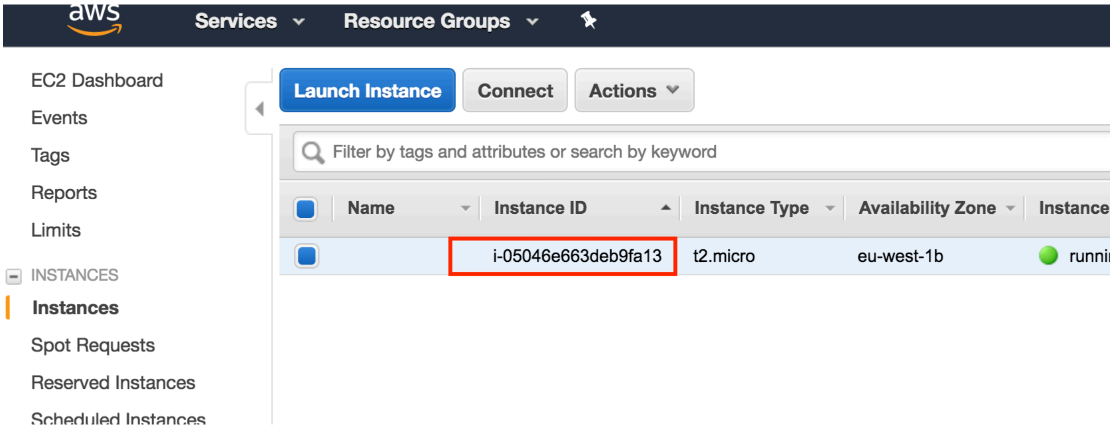
    c. For the region choose London: **eu-west-2**

    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.*

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

41. From the AWS Web-based console, go back to the EC2 page, and then choose Running Instances. Find your running EC2 instance and find the id of your running instance:



Name	Instance ID	Instance Type	Availability Zone	Instance
	i-05046e663deb9fa13	t2.micro	eu-west-1b	● runni

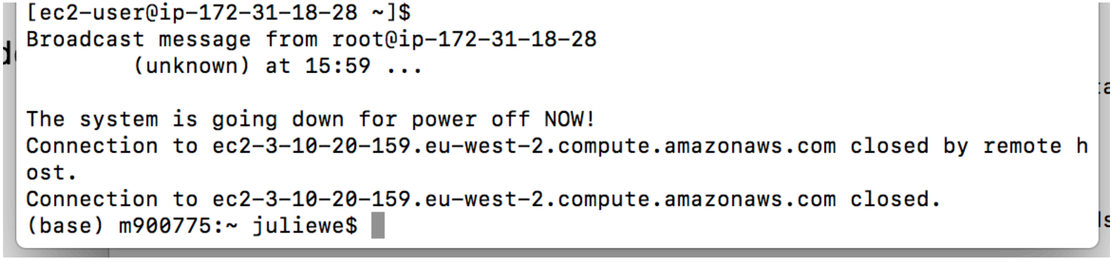
42. Now use the AWS CLI to terminate:
    Replacing the instance ID with your own, type:

    aws2 ec2 terminate-instances --instance-ids **i-05046e663deb9fa13**

43. You should see a 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"
            }
        }
    ]
}
```

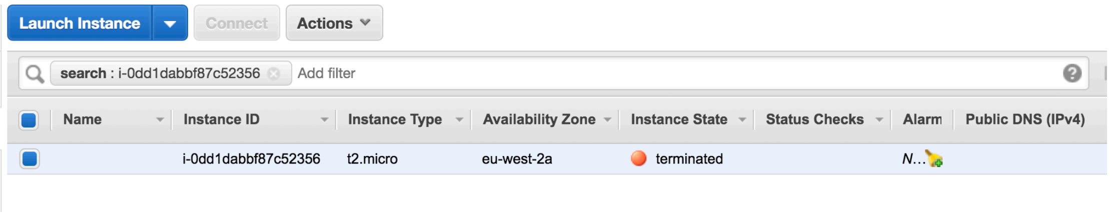
44. Your SSH session to the server will die, and the server will no longer be running.



```
[ec2-user@ip-172-31-18-28 ~]$
Broadcast message from root@ip-172-31-18-28
        (unknown) at 15:59 ...

The system is going down for power off NOW!
Connection to ec2-3-10-20-159.eu-west-2.compute.amazonaws.com closed by remote h
ost.
Connection to ec2-3-10-20-159.eu-west-2.compute.amazonaws.com closed.
(base) m900775:~ juliewe$
```

45. It is really important to check on the AWS console that this instance has actually been terminated (or stopped). If it does not shut down in a reasonable amount of time from giving the command to the AWS CLI, you can terminate it in the console. Click on Instance state and select

terminate or stop. **YOU WILL BE CHARGED BY AWS FOR ANY INSTANCES THAT ARE LEFT RUNNING SO THIS IS REALLY IMPORTANT.**



search : i-0dd1dabbf87c52356 Add filter

☐	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm	Public DNS (IPv4)
☐		i-0dd1dabbf87c52356	t2.micro	eu-west-2a	● terminated		N...	

46. **Congratulations**! You have completed both of these exercises.