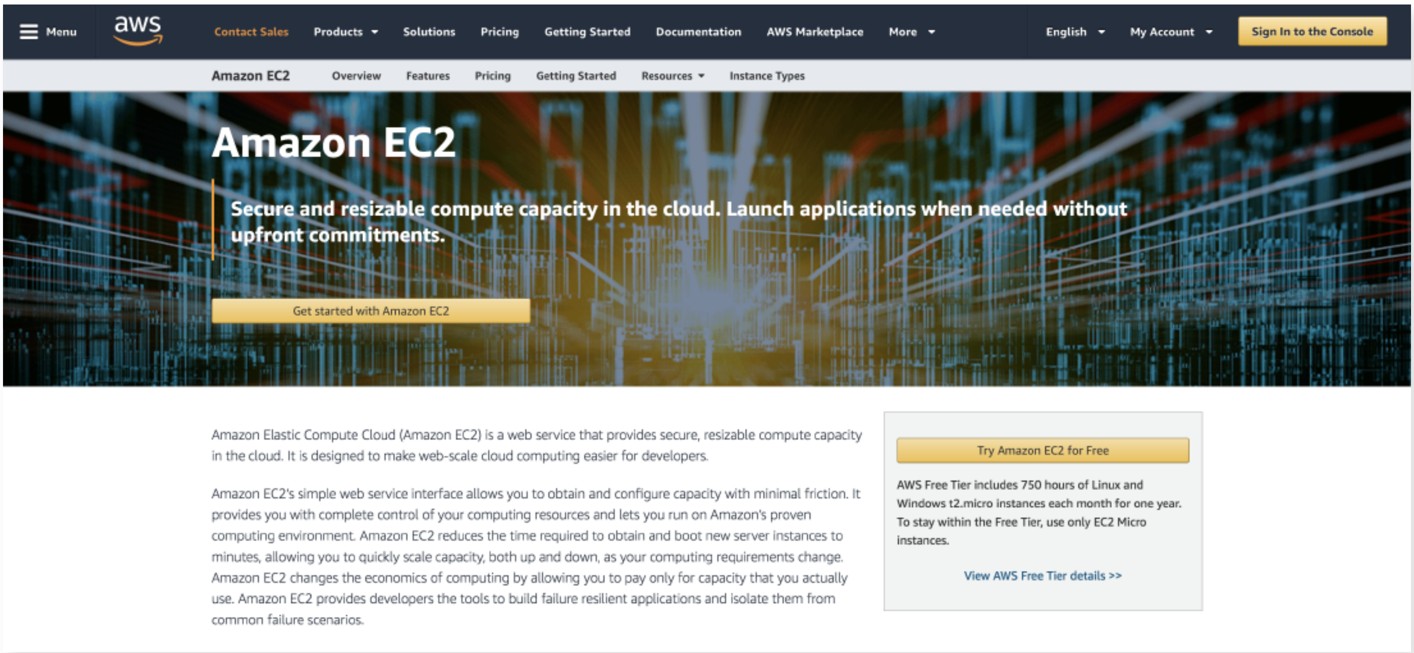
Welcome
to our walkthrough of
Setting up an AWS EC2 Instance

What is an EC2 Instance: An EC2 Instance is a virtual server in Amazon’s Elastic Compute Cloud – EC2 – for running applications on AWS infrastructure.

Part 1: Sign up, Set up and Connect

You need to sign up if you haven’t registered yet with AWS (and you can absolutely use the free tier!) so head on over to https://aws.amazon.com/ec2/ to get started.



Amazon EC2

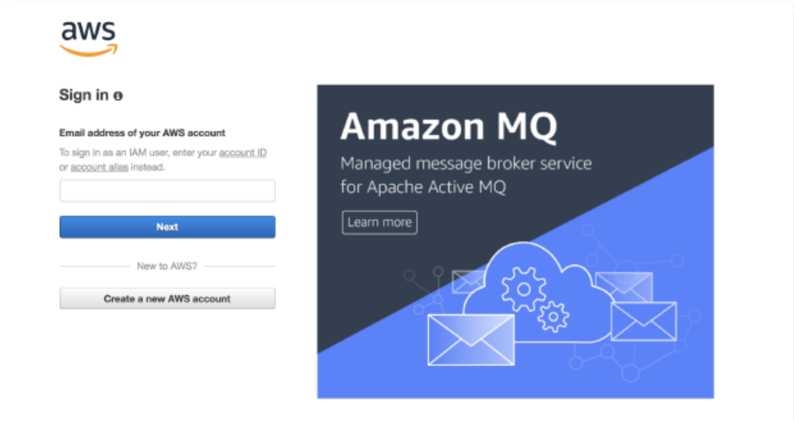
Secure and resizable compute capacity in the cloud. Launch applications when needed without upfront commitments.

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.

Amazon EC2’s simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon’s proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers the tools to build failure resilient applications and isolate them from common failure scenarios.

AWS Free Tier includes 750 hours of Linux and Windows t2.micro instances each month for one year. To stay within the Free Tier, use only EC2 Micro instances.

You can select the top right to sign in to the console. This will allow you to sign in or set up an account if you do not have one.



Sign in

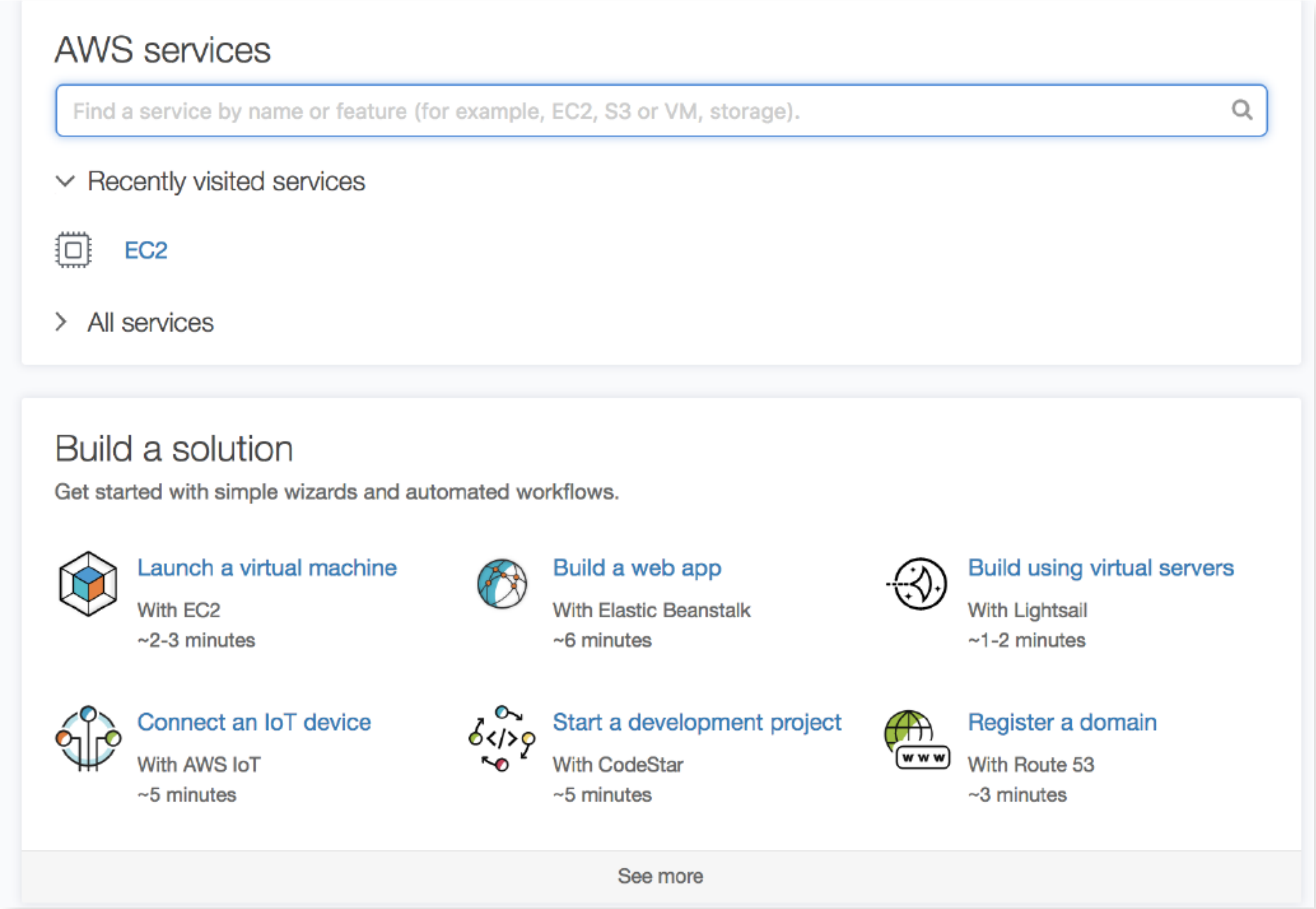
Email address of your AWS account

To sign in as an IAM user, enter your account ID or account alias instead.

Amazon MQ

Managed message broker service for Apache Active MQ

Once you sign in or register you will log into the console on AWS and see the following:



AWS services

Find a service by name or feature (for example, EC2, S3 or VM, storage).

Recently visited services

EC2

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

Build using virtual servers
With Lightsail
~1-2 minutes

Connect an IoT device
With AWS IoT
~5 minutes

Start a development project
With CodeStar
~5 minutes

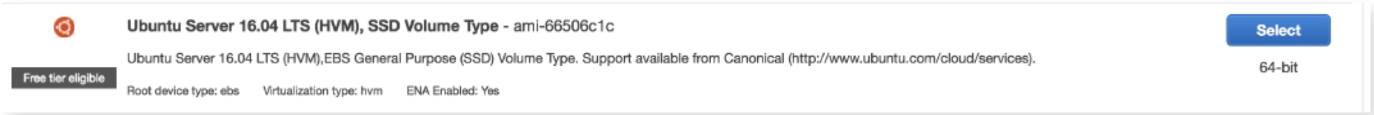
Register a domain
With Route 53
~3 minutes

See more

Here you can select the launch a virtual machine with EC2 option (you can see my recently visited services EC2) When you click on that option it will navigate you to a list of instances for step 1.

Step 1: Choose an Amazon Machine Image

You will see a list of options open up and we are going to choose the Ubuntu Server, #5 in the list. Please find:



Free tier eligible

Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-66506c1c

Ubuntu Server 16.04 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   ENA Enabled: Yes

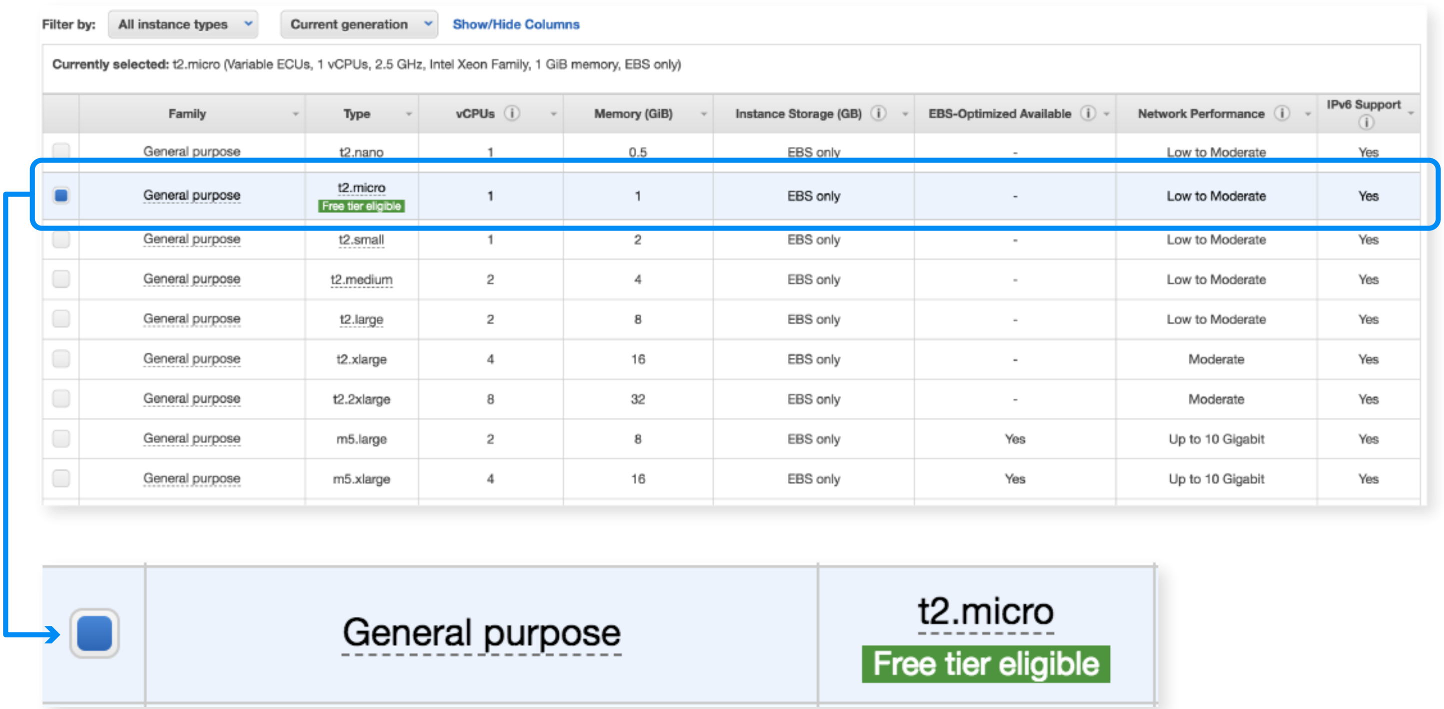
Select

64-bit

Now we have to go through the steps to finish the setup and to customize the instance. You can follow along through each part. Since we are operating on the free tier we will be operating with the services offered for that tier in Step 2: Choose an Instance Type:

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you felixibility to choose the appropiate mix of resources for your applications.



Filter by: All instance types | Current generation | Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
□	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
■	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
□	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
□	General purpose	m5.large	2	8	EBS only	Yes	Up to 10 Gigabit	Yes
□	General purpose	m5.xlarge	4	16	EBS only	Yes	Up to 10 Gigabit	Yes

■	General purpose	t2.micro Free tier eligible

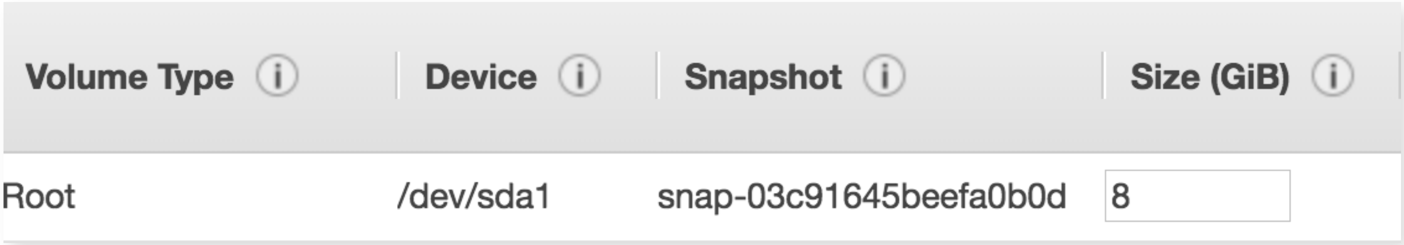
We can now click on the next button (bottom right hand corner) we will move to

Step 3: Configure Instance Details

Here we can leave everything for the purpose of our tutorial on the default settings. Let’s move on to the next step.

Step 4: Add Storage

The instance has a 30GB limit but for the purpose of our project we won’t need to use all 30. Let’s use 16GB. The default size is 8 and you can change it to 16 by simply entering 16 under the options for Size (GiB). When you are done setting the storage move on to the next step.



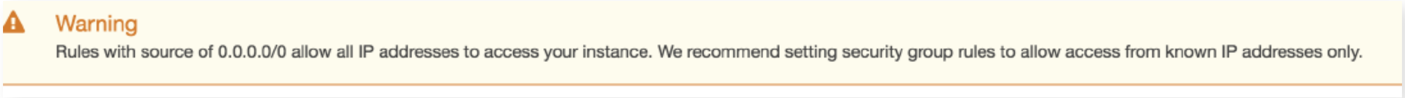
Volume Type	Device	Snapshot	Size (GiB)
Root	/dev/sda1	snap-03c91645beefa0b0d	8

Step 5: Add Tags

For the tags it's not a requirement to establish any tags so you don't have to assign them but it may help you manage your instances on Amazon but if you would like more information about tags please see here:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html?icmpid=docs_ec2_console

Step 6: Configure Security Group

This option allows you to assign a security group to the AWS EC2 instance. One of key pieces of information here is establishing the Source or the IP addresses that are allowed to connect to the instance. **I recommend listening to the warning by Amazon that states:**



⚠ 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 only.

You can change the source to My IP so that it uses your IP and doesn't allow all IP addresses to access your instance.

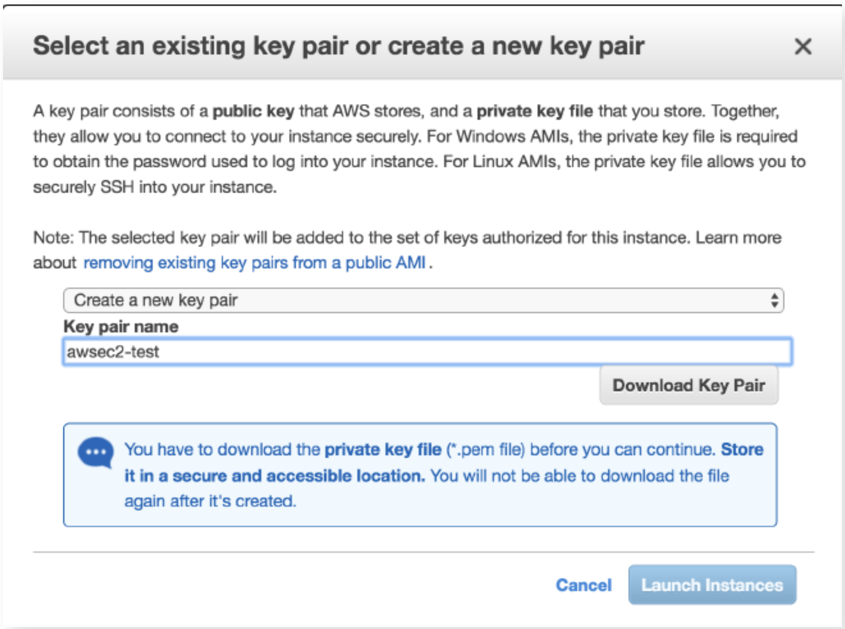
Almost there!

Step 7: Review

Creation of the .pem file. This is important so please follow the following instructions before launching the instance.

Create a new key pair which will download a file (make sure you download the key pair) as it's used to access the instance. Save this file and make sure it's in a secure location as it will needed each time to connect. You can create a folder so that it makes navigating to the .pem file easier. In the walkthrough video I created a folder that I can navigate/CD to in the future in the terminal so that the .pem when connecting.

I have a key pair created and named it awsec2-test. You can give it the name that you would like but again save the file in a location that you can navigate to as you will need to be in that location to access the instance.



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

awsec2-test

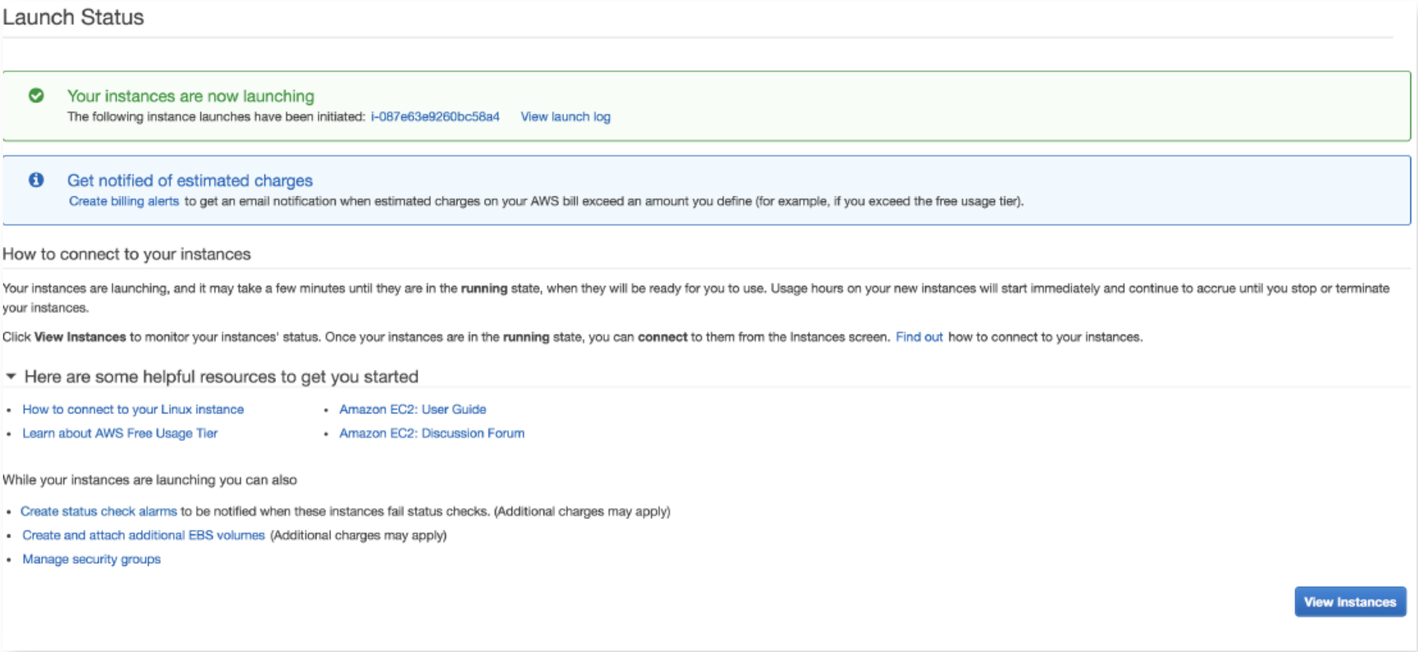
Download Key Pair

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.

Once you have the key pair downloaded, you can launch the instance. It will navigate you to our next step.

Step 8: Review Instance Launch

Here you can review the steps that we went through to set up the instance. If everything looks acceptable we can click Launch. Give the instance a moment to start after launching it. It will bring you after launch to a page such as:



Launch Status

Your instances are now launching
The following instance launches have been initiated: i-087e63e9260bc58a4 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.

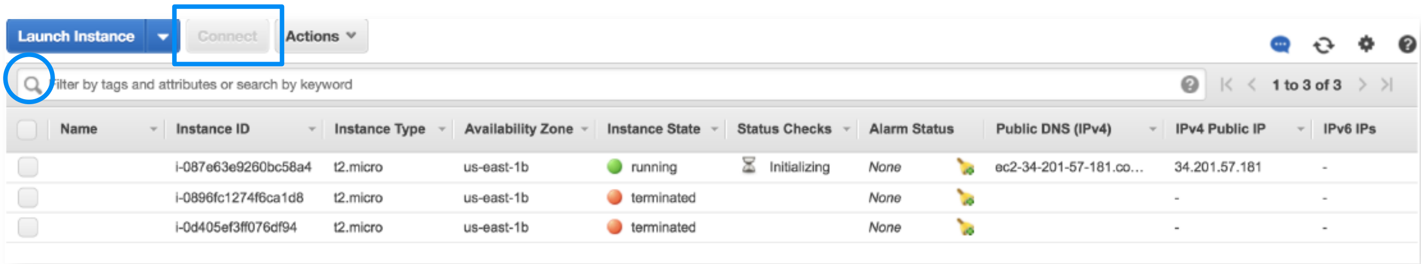
▾ Here are some helpful resources to get you started

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

While your instances are launching you can also

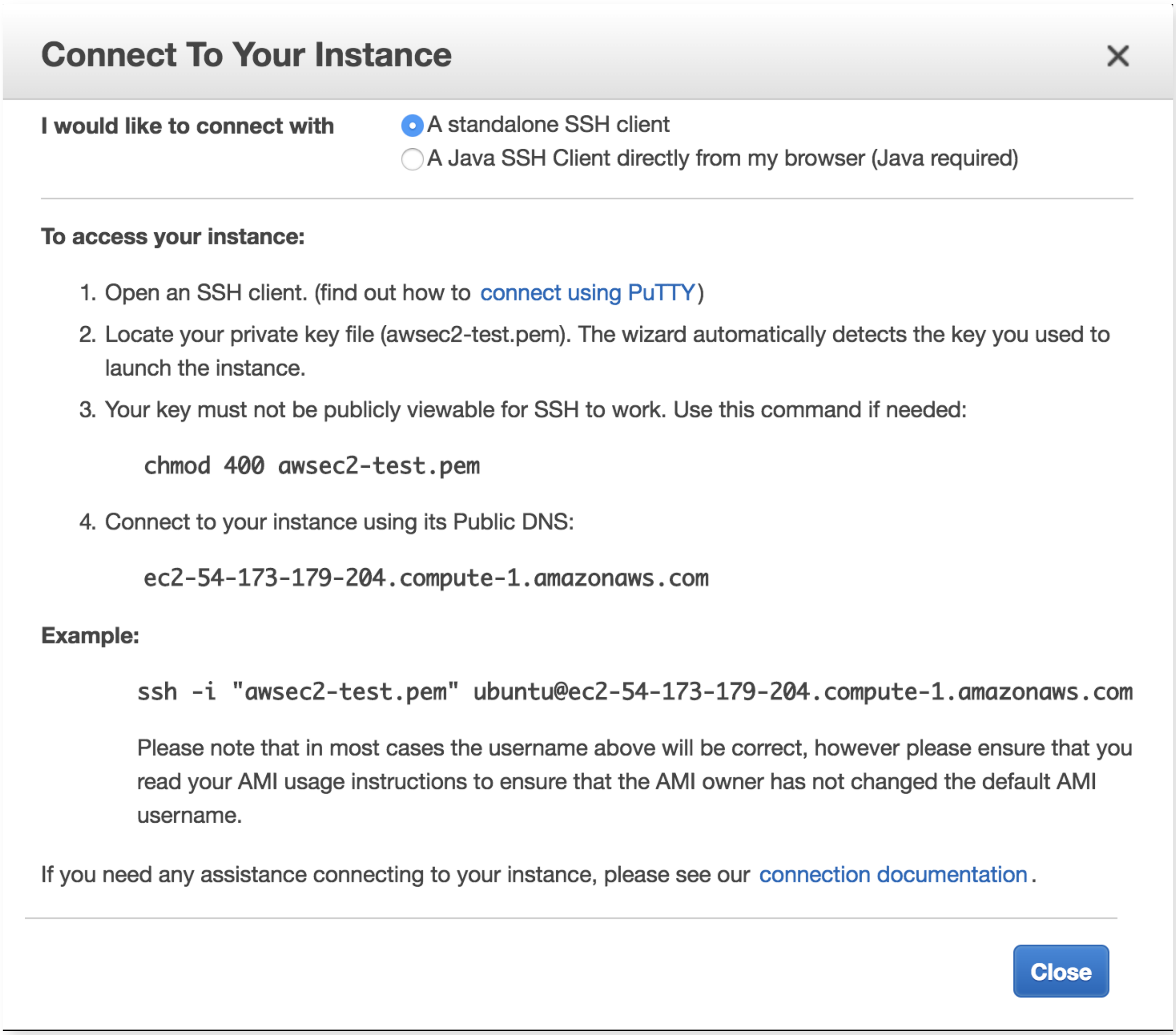
- Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)
- Create and attach additional EBS volumes (Additional charges may apply)
- Manage security groups

In the bottom right hand corner click the blue button for View Instances. Here you will find the instances that you set up (I have a few already and if this is your first you will only see one). For example you can see instances running and terminated. When you are ready to connect to your instance click on the box next to the name of the instance you have running and then click on connect.



□	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs
□		i-087e63e9260bc58a4	t2.micro	us-east-1b	● running	Initializing	None	ec2-34-201-57-181.co...	34.201.57.181	-
□		i-0896fc1274f6ca1d8	t2.micro	us-east-1b	● terminated		None		-	-
□		i-0d405ef3ff076df94	t2.micro	us-east-1b	● terminated		None		-	-

Again click on the selector for the name and it will open the option to connect. This will open up a list of directions and instructions on how to connect.



I would like to connect with

● A standalone SSH client

○ A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to connect using PuTTY)
2. Locate your private key file (awsec2-test.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 awsec2-test.pem
```

4. Connect to your instance using its Public DNS:

```
ec2-54-173-179-204.compute-1.amazonaws.com
```

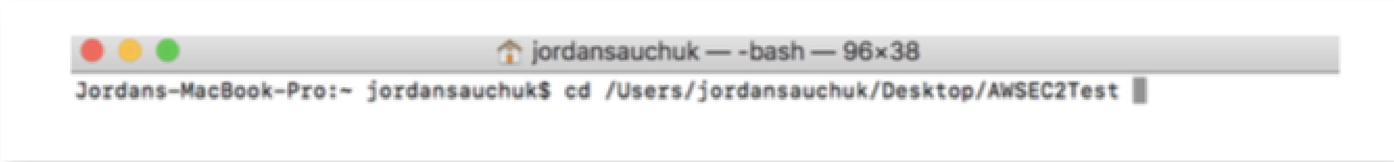
Example:

```
ssh -i "awsec2-test.pem" ubuntu@ec2-54-173-179-204.compute-1.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

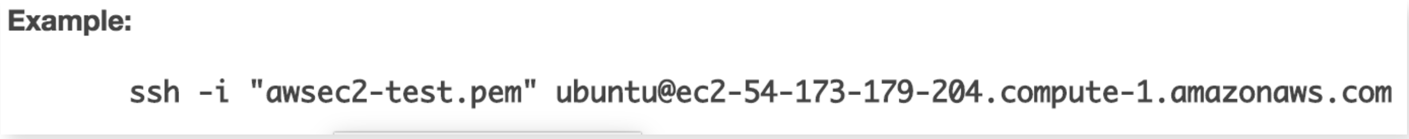
If you need any assistance connecting to your instance, please see our connection documentation.

Now we are ready to connect using the directions that AWS EC2 provides. Since i'm using the Mac OS I will be operating in the terminal using SSH. If you are on Linux you can use the SSH option and on Windows you can see the use of Putty (click the link in the instructions for using PuTTY). Now we can open up a terminal and navigate to the location that the .pem file is in. I stored my .pem in a folder, so I will CD there.



```
Jordans-MacBook-Pro:~ jordansauchuk$ cd /Users/jordansauchuk/Desktop/AWSEC2Test
```

Once you are located in the directory of the .pem file you can use the instructions from Connect to Your Instance. You can copy and paste in the terminal the example as it will work in most cases. Yours will be different depending on the names and settings but copy the example from your output message. Mine is:



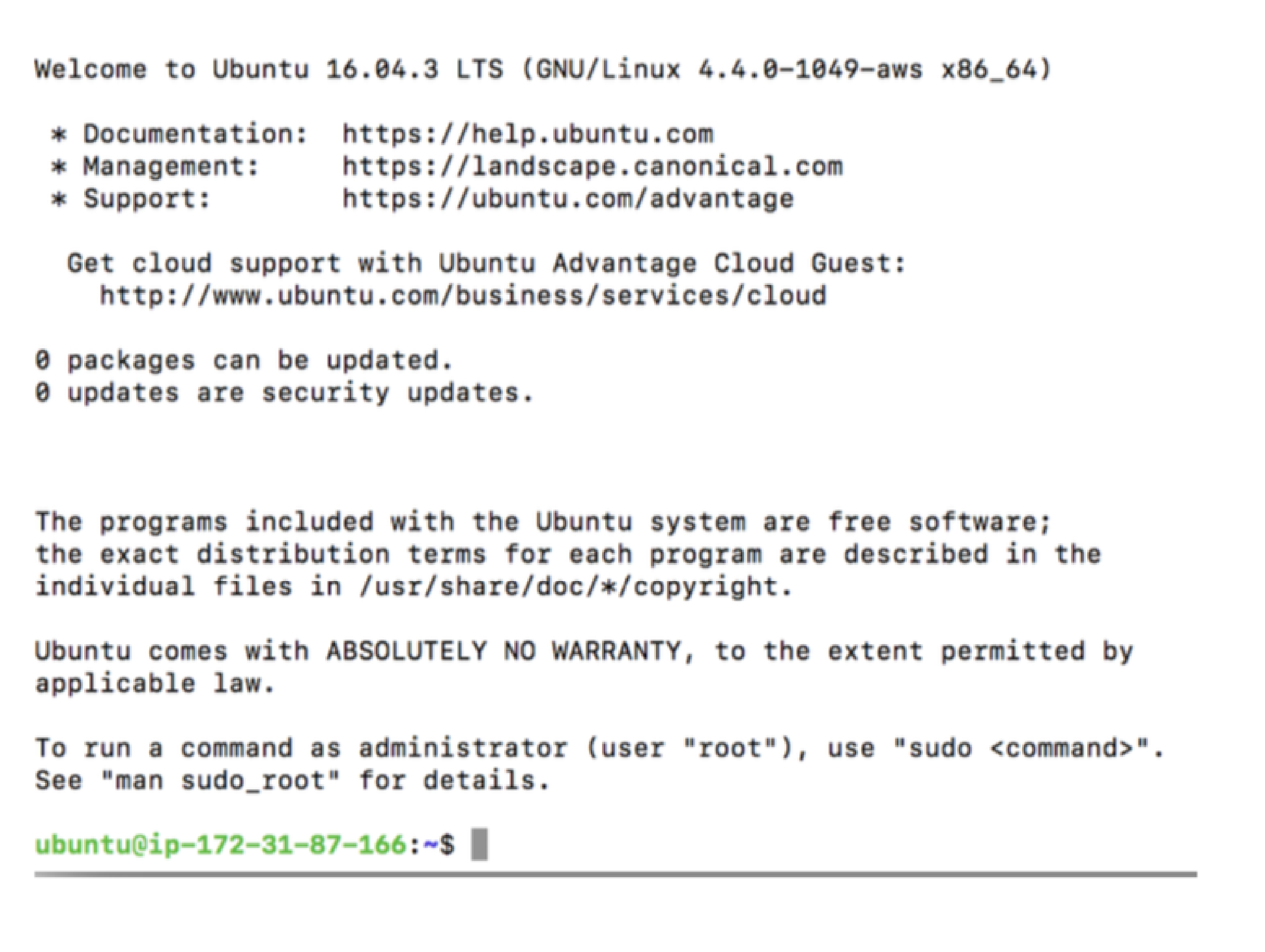
Example:

```
ssh -i "awsec2-test.pem" ubuntu@ec2-54-173-179-204.compute-1.amazonaws.com
```

Navigate back to the terminal and paste the command in and run it. You can authenticate it if it has the message pop up for ECDSA but typing in yes. If you have any connectivity issues, check the IP/security settings that you used.

If a public key error displays as in the video use the connect directions, specifically step 3:
3. Use the chmod 400 awsec2-test.pem command in the terminal and then re-run the ssh connection.

With that you will be connected to the Ubuntu AWS EC2 Instance. Great Job!



```
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-1049-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

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.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-87-166:~$
```

Part 2: Anaconda, Jupyter and Regression

Since we have our EC2 instance connected we have to treat it as a brand new OS to run our linear regression and we will use Anaconda to install the required packages along with Jupyter notebooks.

Step 1: Install Anaconda

Run the wget command and Anaconda distribution to download. We are using a Linux/Ubuntu OS and 64 bit so we will use:
wget https://repo.continuum.io/archive/Anaconda3-5.1.0-Linux-x86_64.sh

You can also find other versions here as well for reference:
https://repo.continuum.io/archive/index.html

Step 2: After the download please run:

bash Anaconda3-5.1.0-Linux-x86_64.sh

This will bring up the license information and options along with the installation path along with prepending the installation path to the bash file.

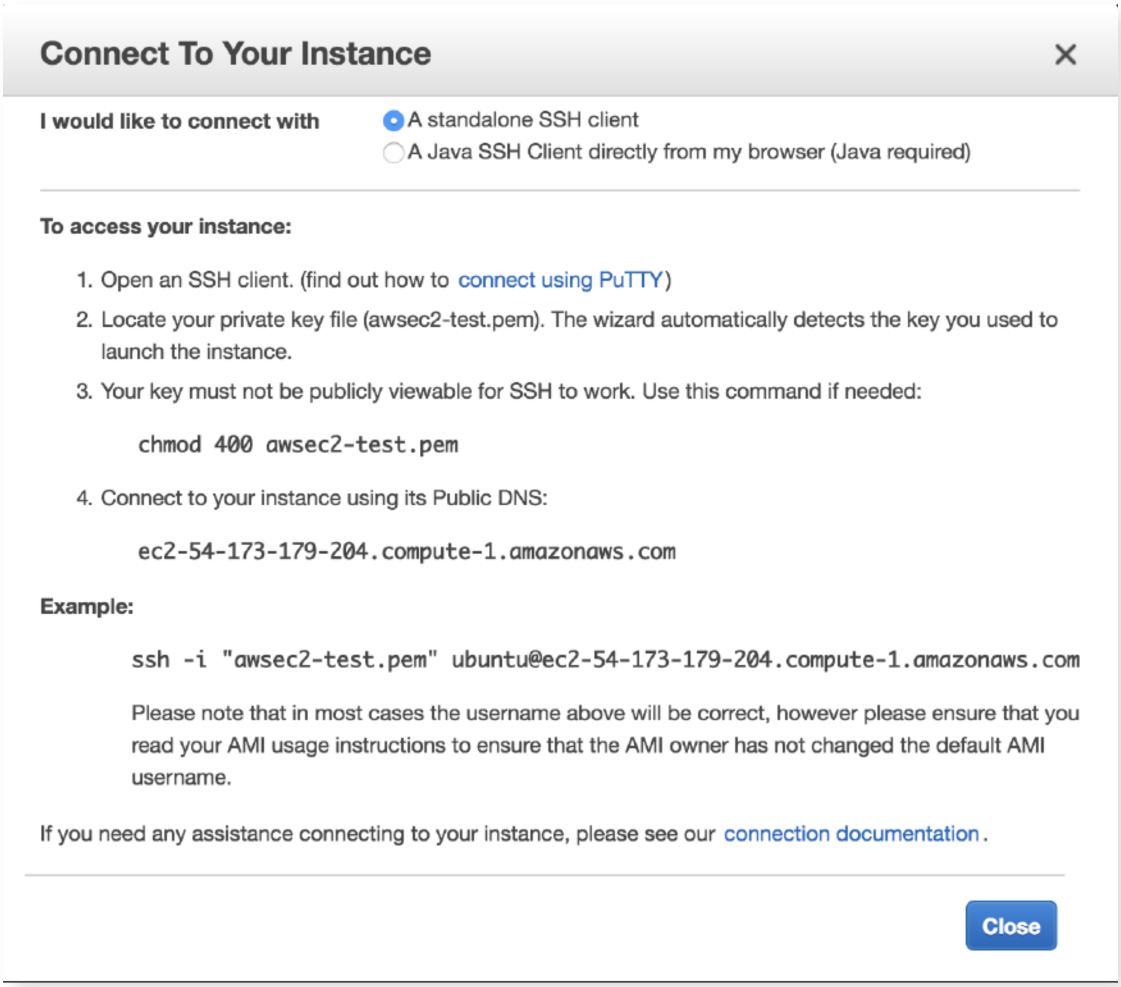
You do not need to install VS code for the purpose of this tutorial. You can enter clear in the terminal to remove the information so that you have a clean terminal.

Step 3: We need to use the following command due to a possible conflict with javascript:

jupyter notebook --no-browser –port=8888

Step 4: Open a new terminal and run the following:

*Note that the awsec2-test is the pem file created and the direction of ubuntu@ec2-etc is from the directions specified to connect to your instance, so it will be different but look something like the example and info in #4 below:



Connect To Your Instance ×

I would like to connect with ● A standalone SSH client
○ A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to connect using PuTTY)
2. Locate your private key file (awsec2-test.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 awsec2-test.pem
```

4. Connect to your instance using its Public DNS:

```
ec2-54-173-179-204.compute-1.amazonaws.com
```

Example:

```
ssh -i "awsec2-test.pem" ubuntu@ec2-54-173-179-204.compute-1.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our connection documentation.

ssh -i awsec2-test -L 8000:localhost:8888 ubuntu@ec2–54–172–178–44.compute-1.amazonaws.com

Step 5: Open a Jupyter notebook in the browser

Open a browser and navigate to the following directions:

localhost:8000

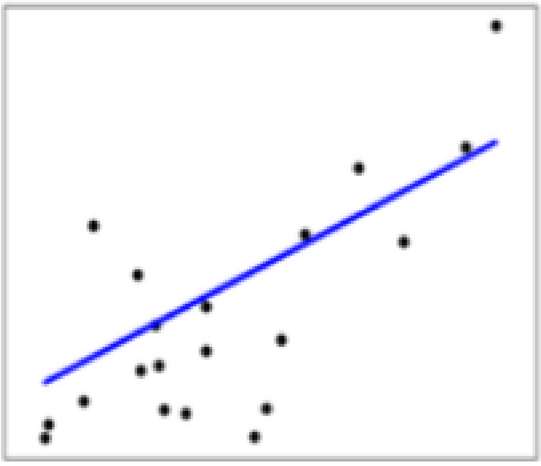
Navigate back to the terminal where you can find the token information (the first and original terminal that we were operating from). Use the token information to log into your Jupyter notebook.

Congratulations! You are now working in a Jupyter notebook and can run the regression code below!

To help understand linear regression further from:
http://scikit-learn.org/stable/modules/linear_model.html

"LinearRegression fits a linear model with coefficients $w = (w_1, ..., w_p)$ to minimize the residual sum of squares between the observed responses in the dataset, and the responses predicted by the linear approximation. Mathematically it solves a problem of the form:

$$\min_{w} ||Xw - y||_2{}^2$$



OLS – Ordinary Least Squares code:

```
from sklearn import linear_model
reg = linear_model.LinearRegression() reg.fit ([[0, 0], [1, 1], [2, 2]], [0, 1, 2])
reg.coef_
```

Ridge regression code:

```
from sklearn import linear_model
reg = linear_model.Ridge (alpha = .5) reg.fit ([[0, 0], [0, 0], [1, 1]], [0, .1, 1]) reg.coef_
reg.intercept_
```

http://scikit-learn.org/stable/modules/linear_model.html