Starting and Provisioning an EC2 Instance

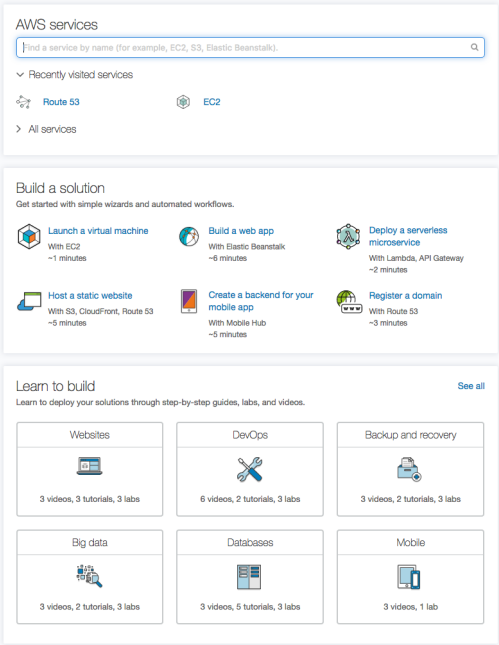
Aim

This course will be taught using Jupyter [1] notebooks hosted on an Amazon Web Services (AWS) Elastic Cloud Compute (EC2) instance. The aim of these instructions is to: Start an EC2 instance with the correct access permissions and use the key provided to log in, run some pre-defined scripts to provision the EC2 instance with Anaconda Python [2] and the tools required for the course and, finally, to start the Jupyer notebook server and connect to it using a web browser.

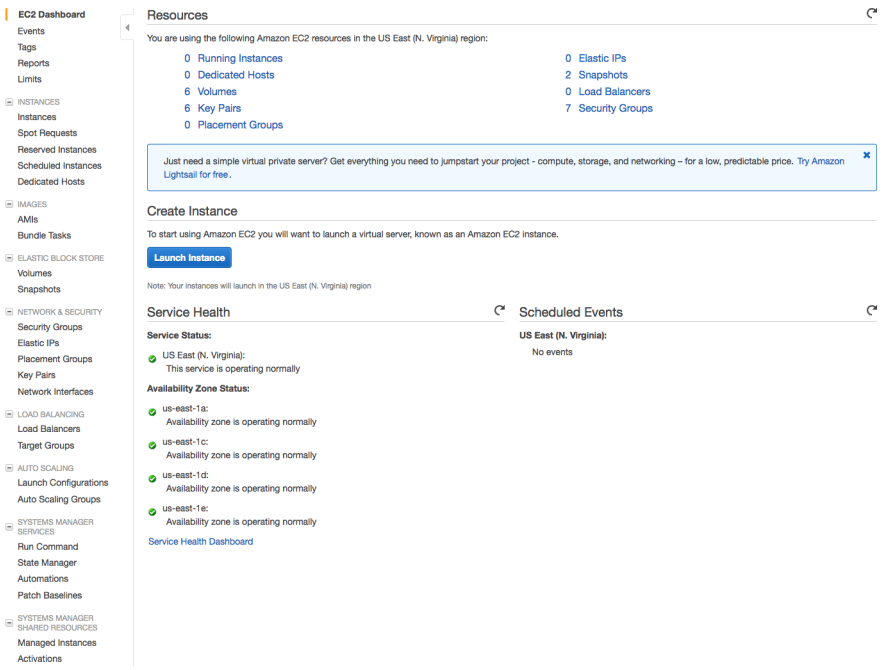
First: Stanting an EC2 instance

1. Point your browser at https://aws.amazon.com/ and click "sign into the console" in the top right hand corner.

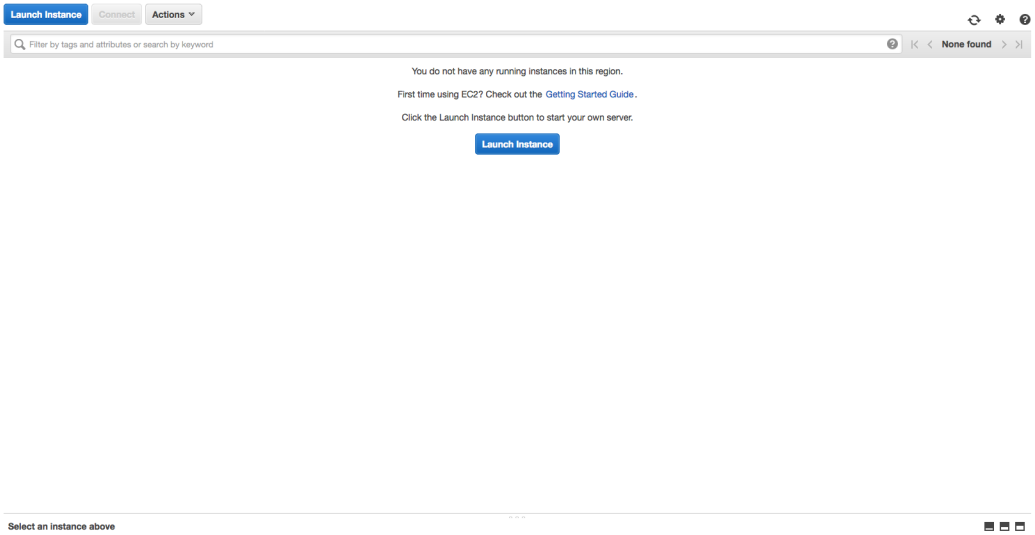
2) Log in with the credentials you generated using the pre-course instructions, once authenticated this will take you to the console which will look *similar* to this:



3) Type "ec2" into the text box under "AWS services" and click the first option in the drop down box. This will land you on a page that looks similar to this (of course you will not have existing snapshots etc...):

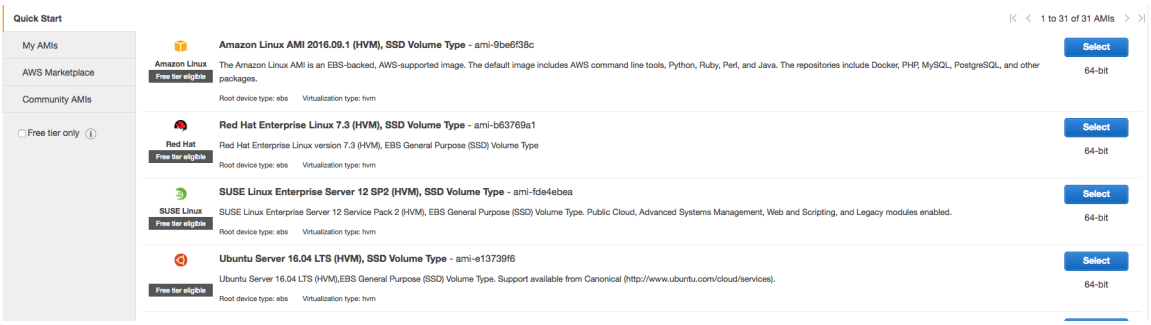


3. If it is not expanded out already (ie there is a "+" sign) click on "Instances" and in the menu below it click on "Instances" and you will be greeted by this screen:



You can now click on "Launch Instance" on either blue and white button.

4. You will be greeted by a page that looks like this:



click select next to the "Ubuntu Server 16.04 LTS (HVM), SSD Volume Type" option. The next page asks you to click a radio button next to the "size" of the machine you want to start. **While you are**

experimenting use "t2.micro" which gives you a 1GB 1CPU EC2 instance. **For the course** 1GB will not cut it, select "m4.large". The on demand pricing [3] is 10.8 cents an hour.

Currently selected: m4.large (6.5 ECUs, 2 vCPUs, 2.4 GHz, Intel Xeon E5-2676v3, 8 GiB memory, EBS only)

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
General purpose	t2.small	1	2	EBS only	-	Low to Moderate
General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
General purpose	t2.large	2	8	EBS only	-	Low to Moderate
General purpose	t2.xlarge	4	16	EBS only	-	Moderate
General purpose	t2.2xlarge	8	32	EBS only	-	Moderate
General purpose	m4.large	2	8	EBS only	Yes	Moderate
General purpose	m4.xlarge	4	16	EBS only	Yes	High
General purpose	m4.2xlarge	8	32	EBS only	Yes	High
General purpose	m4.4xlarge	16	64	EBS only	Yes	High
General purpose	m4.10xlarge	40	160	EBS only	Yes	10 Gigabit
General purpose	m4.16xlarge	64	256	EBS only	Yes	20 Gigabit
General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate
General purpose	m3.large	2	7.5	1 x 32 (SSD)	-	Moderate
General purpose	m3.xlarge	4	15	2 x 40 (SSD)	Yes	High
General purpose	m3.2xlarge	8	30	2 x 80 (SSD)	Yes	High

Cancel | Previous | Review and Launch | Next: Configure Instance Details

Once your selection has been made click the **gray** button "Next: Configure Instance Details".

5) The next page can be left as all defaults... Click the **gray** button "Next: Add Storage"

6) Again, this can be left as all defaults. Click the **gray** button "Next: Add Tags"

7) *Again*, no need to add tags for the purposes of this course... Once more click the **gray** button "Next: configure Security Group"

8) Now we have work to do! We need to configure our instance to be able to serve the Jupyter notebook via HTTPS on port 8888. You will see a page similar to that below. Make sure "Create a new security group" is checked, later you can select "select and existing group" to save you time! Enter a simple name for the Security group name like "j_sever". Enter something descriptive for the Description.

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

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 want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. 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: j_server

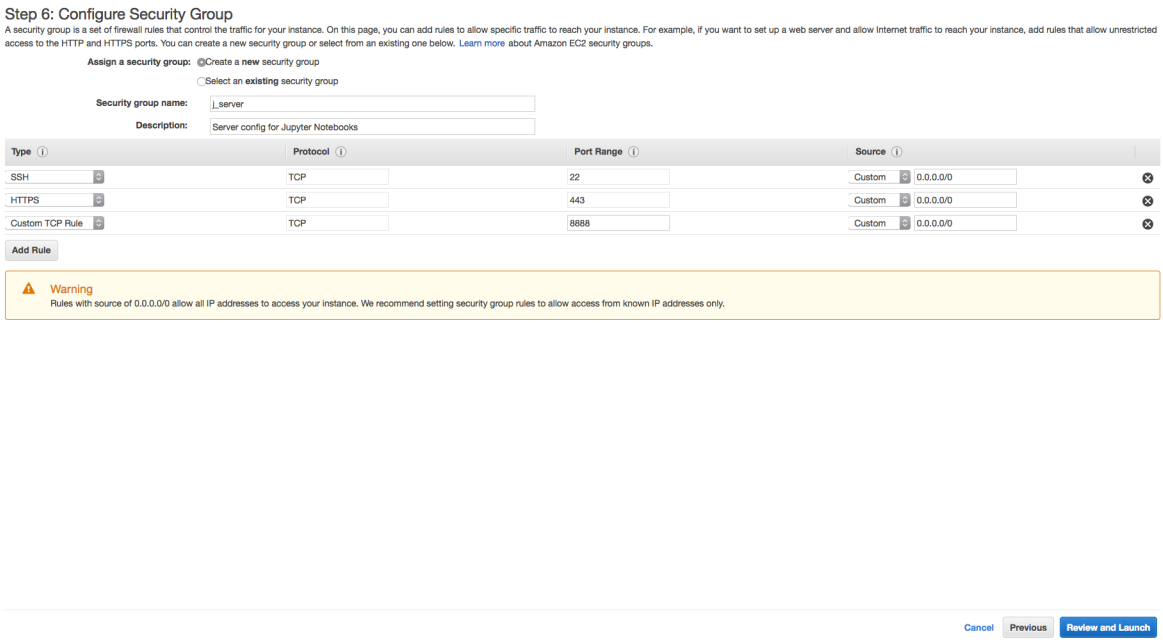
Description: Server config for Jupyter Notebooks

Type	Protocol	Port Range	Source
SSH	TCP	22	Custom 0.0.0.0/0

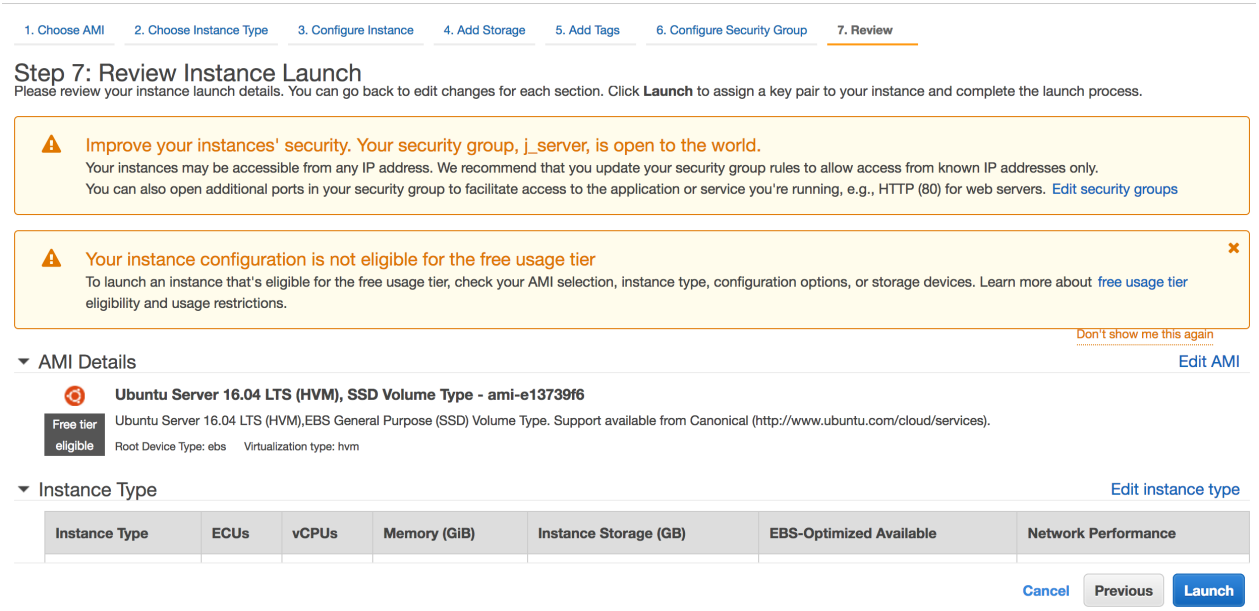
Add Rule

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 only.

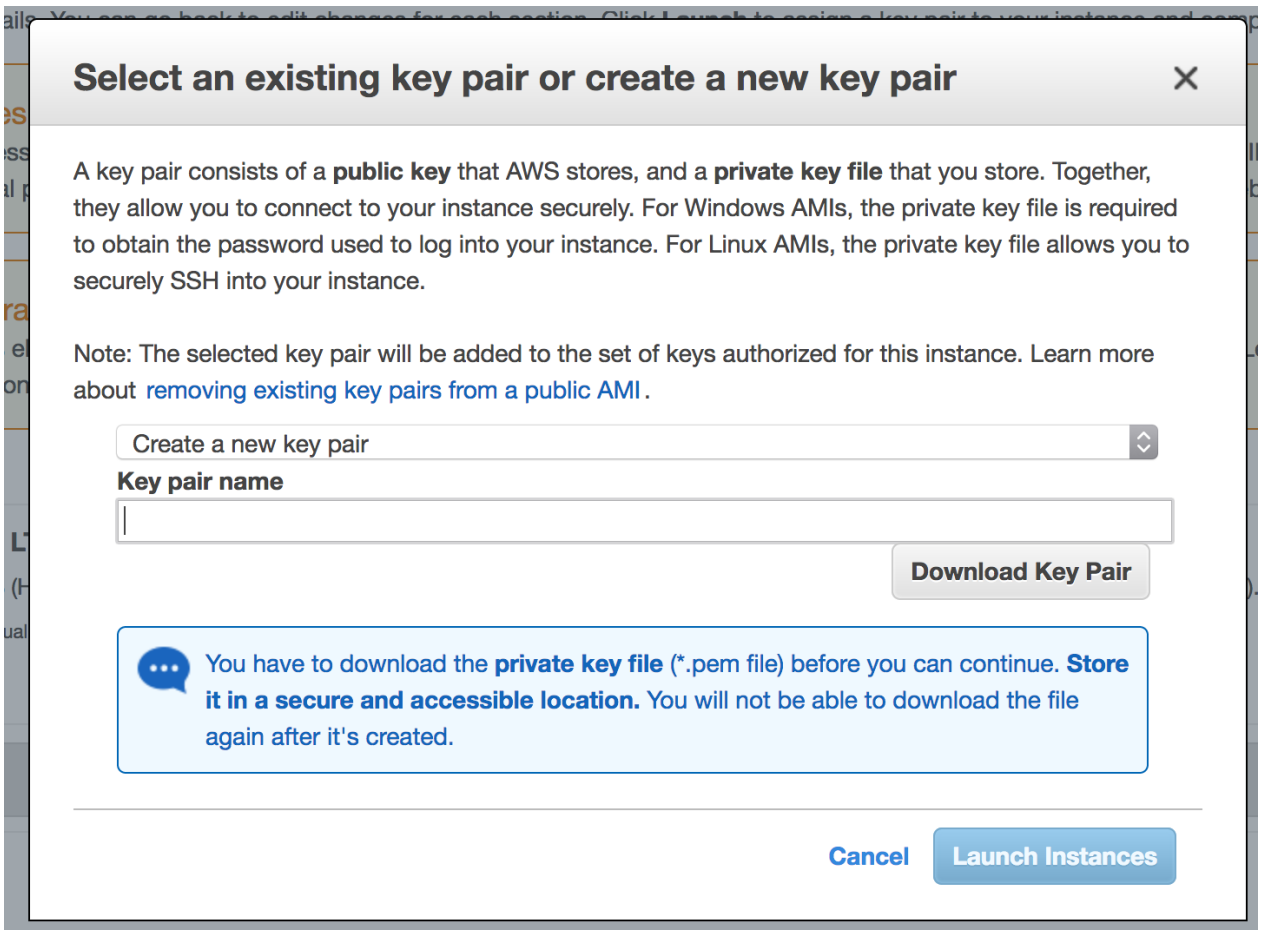
We need to configure to allow connections from any IP to ports 443 (HTTPS) and 8888 (where Jupyter listens). Click "Add Rule". This will create a new row. On the new row the drop down box on the left will default to "Custom TCP Rule". Click it and select "HTTPS". Then click "Add Rule" again but this time, in the new row leave the drop down on "Custom TCP Rule". In that same, third, row enter "8888" into "Port Range" column and "0.0.0.0/0" in the source column. Once done it should look like this:



9. Now we can click the **blue** button "Review and Launch". You will be presented with a page like this for one final check.



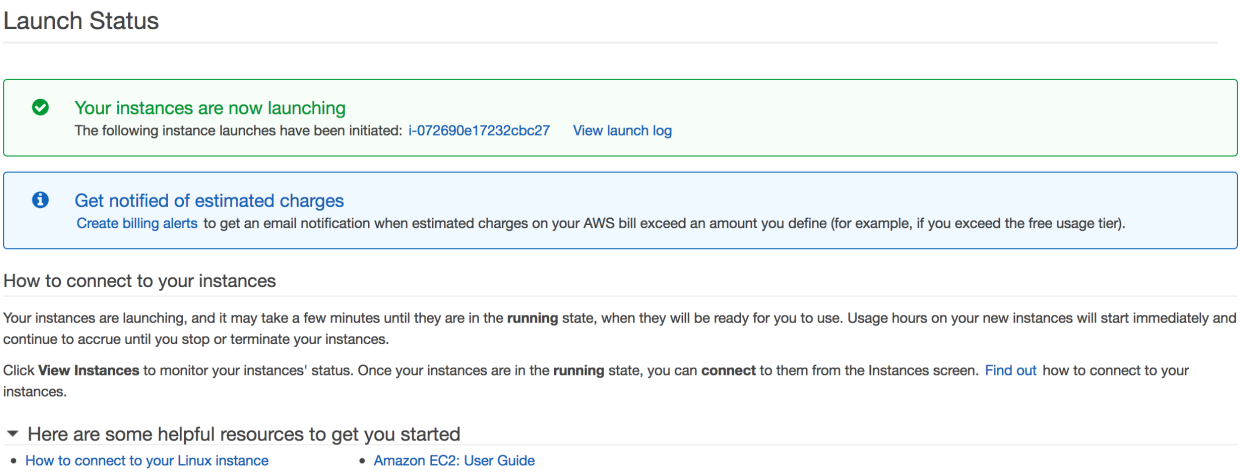
10. click Launch and a dialog pops up like this:



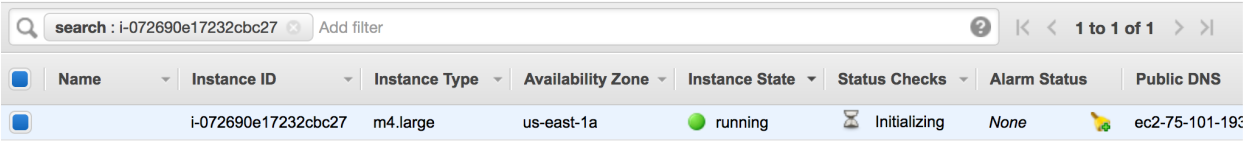
The top drop down can be either "use existing pair", "create new pair" or "proceed without pair". Select "Create new pair", in the text entry below think of a good name (eg "jupyter") and click on download key pair. A save dialog will pop up *REMEBER WHERE YOU SAVED IT!* we will call this "/path/to/key/" in future reference so the key is at "/path/to/key/key_name.pem".

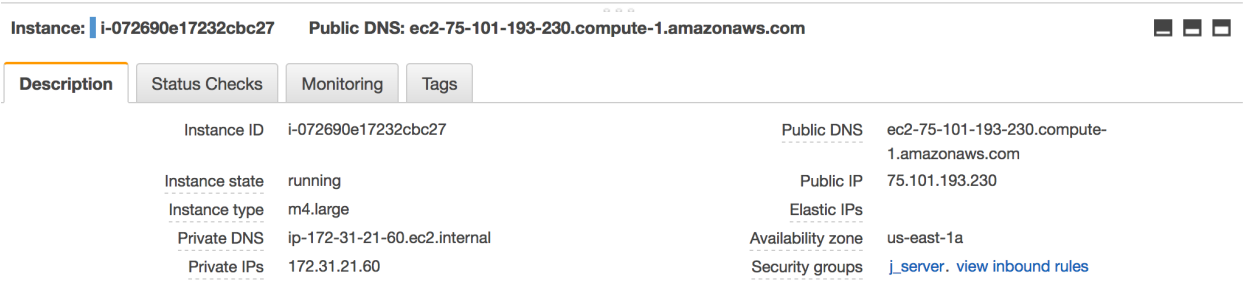
Once you have downloaded the key the **blue** "Launch Instance" button will be un-grayed and you can click it!

11. You will now have a screen like this (after a spinning wheel screen):



click on the hyperlink starting with "i", in our case "i-072690e17232cbc27". You will get a window like the one below. *Some entries may be blank until the instance comes up*.





Second: Logging into your instance

12) You will need an SSH client. These instructions will vary from client to client. Initially this tutorial will have instructions for a terminal based client available on Linux and MacOS/OSX.

Select and copy the domain name next to "Public DNS". First we we need to change the permissions on the key file.

to Go to a terminal and type:

chmod 400 /path/to/key/key.pem (replace with your actual path to your key).

"ssh -i /path/to/key/key_name.pem ubuntu@" and copy paste the domain name to the end of the file.. Hit enter and answer "yes" when it asks if you want to continue connecting. **YES! You are now logged into your instance!**.

What you will have is something like:

```
➜  ~ chmod 400 keys/jupyter.pem
➜  ~ ssh -i /Users/scollis/keys/jupyter.pem ubuntu@ec2-75-101-193-230.compute-1.amazonaws.com
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-53-generic 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-172-31-21-60:~$
```

Third: Provisioning your instance

13) We now need to load on software we use for the course. This will involve executing a shell script located in the course GitHub Repository.

In the shell type: "git clone https://github.com/openradar/AMS_radar_in_the_cloud", hit enter/return this will clone the remote repository into your local machine.

Then, at the prompt type: "source AMS_radar_in_the_cloud/ec2_setup/setup_ec2.sh" you will have a terminal that will look something like this:

```
➜ ~ chmod 400 keys/jupyter.pem
➜ ~ ssh -i /Users/scollis/keys/jupyter.pem ubuntu@ec2-75-101-193-230.compute-1.amazonaws.com
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-53-generic 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-172-31-21-60:~$ git clone https://github.com/openradar/AMS_radar_in_the_cloud
Cloning into 'AMS_radar_in_the_cloud'...
remote: Counting objects: 178, done.
remote: Compressing objects: 100% (104/104), done.
remote: Total 178 (delta 50), reused 0 (delta 0), pack-reused 58
Receiving objects: 100% (178/178), 943.48 KiB | 0 bytes/s, done.
Resolving deltas: 100% (76/76), done.
Checking connectivity... done.
ubuntu@ip-172-31-21-60:~$ source AMS_radar_in_the_cloud/ec2_setup/setup_ec2.sh
```

Hit enter/return and all kinds of magic will start happening!

The script will pause for some time after "jupyter-1.0.0- 100% ..".

14) After some time the script will prompt you for a password. Enter something, enter it again.. **remeber it!**

Fourth: Starting the Jupyter notebook

15. After the script finishes it should finish with a set of lines like this:

```
---------------------
Installation complete
---------------------

To Run execute
. ~/.bashrc
source activate ams-workshop
jupyter notebook --certfile=~/certs/mycert.pem --keyfile ~/certs/mycert.key
You can then point your browser at : https://ec2-A-B-C-D.compute-1.amazonaws.com:8888
where  https://ec2-A-B-C-D.compute-1.amazonaws.com is the FQDN of the instance
Note: You will need to add a certificate exception in your browser
Our GUESS (prone to breakage) is:
https://ec2-75-101-193-230.compute-1.amazonaws.com:8888
(ams-workshop) ubuntu@ip-172-31-21-60:~$
```

In the command line run the lines (By copy pasting if you choose):

. ~/.bashrc

source activate ams-workshop

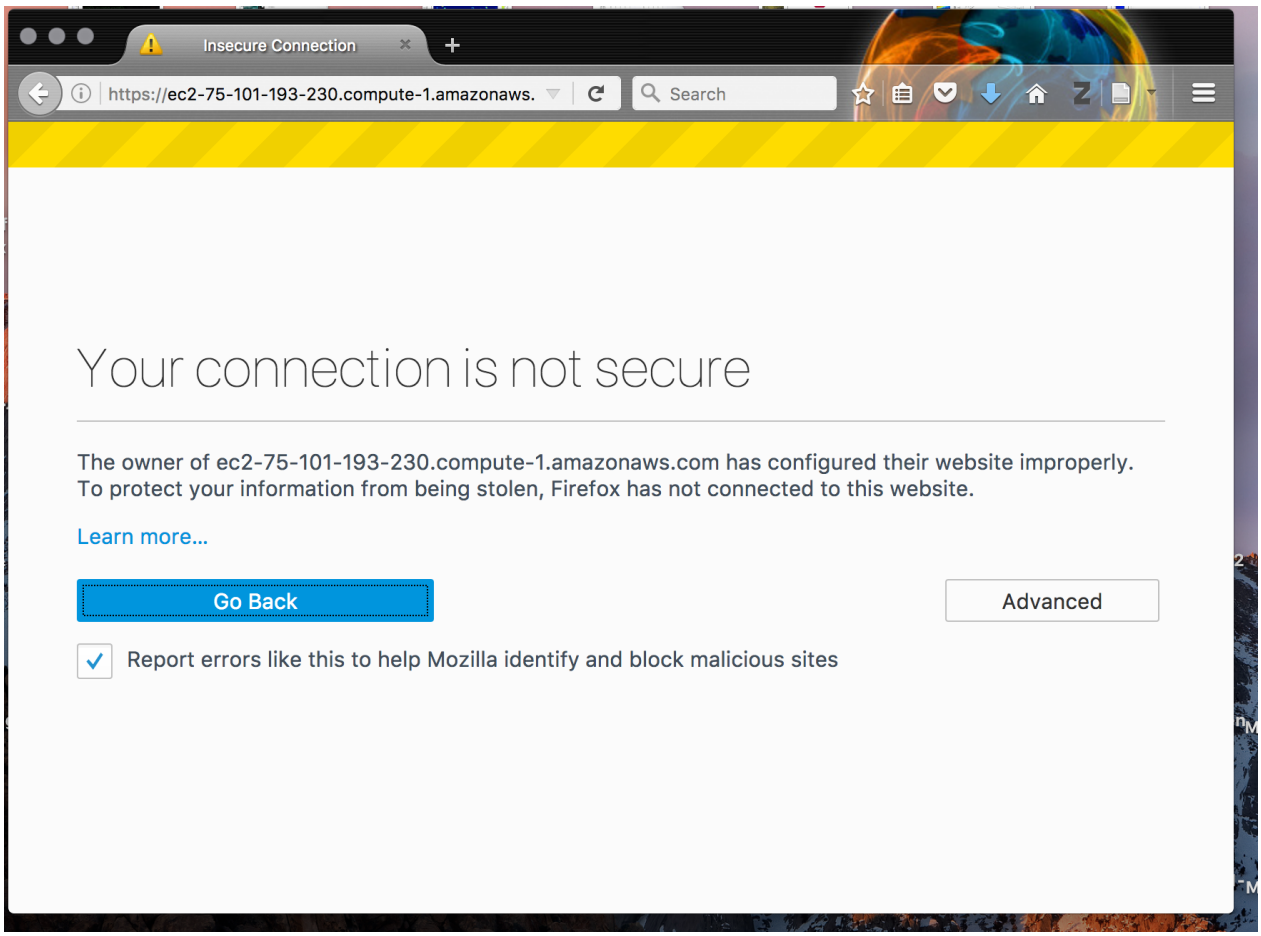
jupyter notebook --certfile=~/certs/mycert.pem --keyfile ~/certs/mycert.key

16) The Jupyter notebook has now started! Huzzah! Your terminal should look like this:

```
--------------------
Installation complete
--------------------

To Run execute
. ~/.bashrc
source activate ams-workshop
jupyter notebook --certfile=~/certs/mycert.pem --keyfile ~/certs/mycert.key
You can then point your browser at : https://ec2-A-B-C-D.compute-1.amazonaws.com:8888
where  https://ec2-A-B-C-D.compute-1.amazonaws.com is the FQDN of the instance
Note: You will need to add a certificate exception in your browser
Our GUESS (prone to breakage) is:
https://ec2-75-101-193-230.compute-1.amazonaws.com:8888
(ams-workshop) ubuntu@ip-172-31-21-60:~$ . ~/.bashrc
ubuntu@ip-172-31-21-60:~$ source activate ams-workshop
(ams-workshop) ubuntu@ip-172-31-21-60:~$ jupyter notebook --certfile=~/certs/mycert.pem --keyfile ~/certs/mycert.key
[I 23:40:24.632 NotebookApp] Writing notebook server cookie secret to /run/user/1000/jupyter/notebook_cookie_secret
[I 23:40:24.685 NotebookApp] Serving notebooks from local directory: /home/ubuntu
[I 23:40:24.685 NotebookApp] 0 active kernels
[I 23:40:24.685 NotebookApp] The Jupyter Notebook is running at: https://[all ip addresses on your system]:8888/
[I 23:40:24.685 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

One last step.. You can see a line after "Our GUESS (prone to breakage) is:". This is the expected location of your Jupyter notebook server.. Copy that line to the clipboard and open a browser window.. Our example here uses Firefox. Paste the address in the "Search or enter address" text entry box and press enter. You should get a screen like this:



ONLY EVER DO THIS FOR SITES YOU COMPLETELY TRUST. Click "Advanced" and then click "Add Exception". A window will drop down, click "Confirm security exception". **Bingo** you should now be presented with a page asking for your password.. enter it and you are good to go!

[1] http://jupyter.org/

[2] https://www.continuum.io/downloads

[3] https://aws.amazon.com/ec2/pricing/on-demand/