DatumGuy

Presents

Machine Learning Bootcamp 2019

Technical Documentation

Table of Content

How to install python libraries required in the bootcamp

What do I need to install all python libraries?

To install the mentioned libraries you would need to **make sure that your python installation also has pip installed** and setup correctly.

What is pip?

pip is a package-management system used to install and manage software packages written in Python.

How to check if pip is installed?

To check if pip is correctly installed on your system, run the following commands:

- For Windows & Ubuntu both
  - pip

If you get the output similar to what is shown below in the screenshot, it means your pip is working fine and we can move forward with the libraries installation.

```
C:\Users\Rishabh Malhotra>pip

Usage:
  pip <command> [options]

Commands:
  install                     Install packages.
  download                    Download packages.
  uninstall                   Uninstall packages.
  freeze                      Output installed packages in requirements format.
  list                        List installed packages.
  show                        Show information about installed packages.
  check                       Verify installed packages have compatible dependencies.
  config                      Manage local and global configuration.
  search                      Search PyPI for packages.
  wheel                       Build wheels from your requirements.
  hash                        Compute hashes of package archives.
  completion                  A helper command used for command completion.
  help                        Show help for commands.

General Options:
  -h, --help                  Show help.
  --isolated                  Run pip in an isolated mode, ignoring environment variables and user configuration.
  -v, --verbose               Give more output. Option is additive, and can be used up to 3 times.
  -V, --version               Show version and exit.
  -q, --quiet                 Give less output. Option is additive, and can be used up to 3 times (corresponding to WARNING, ERROR, and CRITICAL logging levels).
  --log <path>                Path to a verbose appending log.
  --proxy <proxy>             Specify a proxy in the form [user:passwd@]proxy.server:port.
  --retries <retries>         Maximum number of retries each connection should attempt (default 5 times).
  --timeout <sec>             Set the socket timeout (default 15 seconds).
  --exists-action <action>    Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup, (a)bort).
  --trusted-host <hostname>   Mark this host as trusted, even though it does not have valid or any HTTPS.
  --cert <path>               Path to alternate CA bundle.
  --client-cert <path>        Path to SSL client certificate, a single file containing the private key and the certificate in PEM format.
  --cache-dir <dir>           Store the cache data in <dir>.
  --no-cache-dir              Disable the cache.
  --disable-pip-version-check
                              Don't periodically check PyPI to determine whether a new version of pip is available for download. Implied with --no-index.
  --no-color                  Suppress colored output
```

How to install pip?

If you don't get the above output, you need to install pip. To do that follow the below-given instructions:

1. Save the pip installation script from the url https://bootstrap.pypa.io/get-pip.py on your system.
2. Open the command prompt (or terminal), navigate to the location where you saved the pip installation script & run the following command
   python get-pip.py

Installing numpy

- On Windows
  - pip install numpy
- On Ubuntu
  - sudo pip install numpy

Installing pandas

- On Windows
  - pip install pandas
- On Ubuntu
  - sudo pip install pandas

Installing sklearn

- On Windows
  - pip install sklearn
- On Ubuntu
  - sudo pip install sklearn

Installing opencv

- On Windows
  - pip install opencv-python
- On Ubuntu
  - sudo pip install opencv-python

Installing PIL

- On Windows
  - pip install pillow
- On Ubuntu
  - sudo pip install pillow

Installing flask

- On Windows
  - pip install flask
- On Ubuntu
  - sudo pip install flask

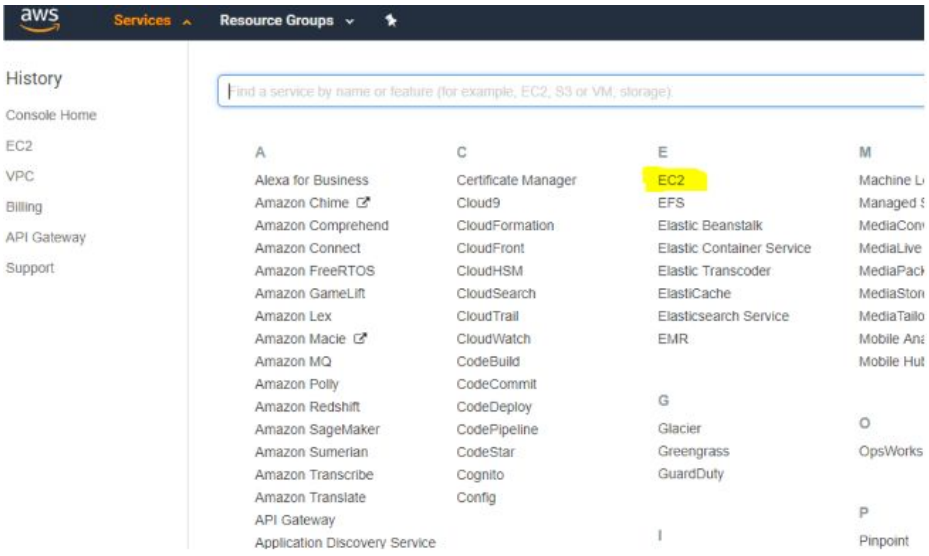
Installing matplotlib

- On Windows
  - pip install matplotlib
- On Ubuntu
  - sudo pip install matplotlib

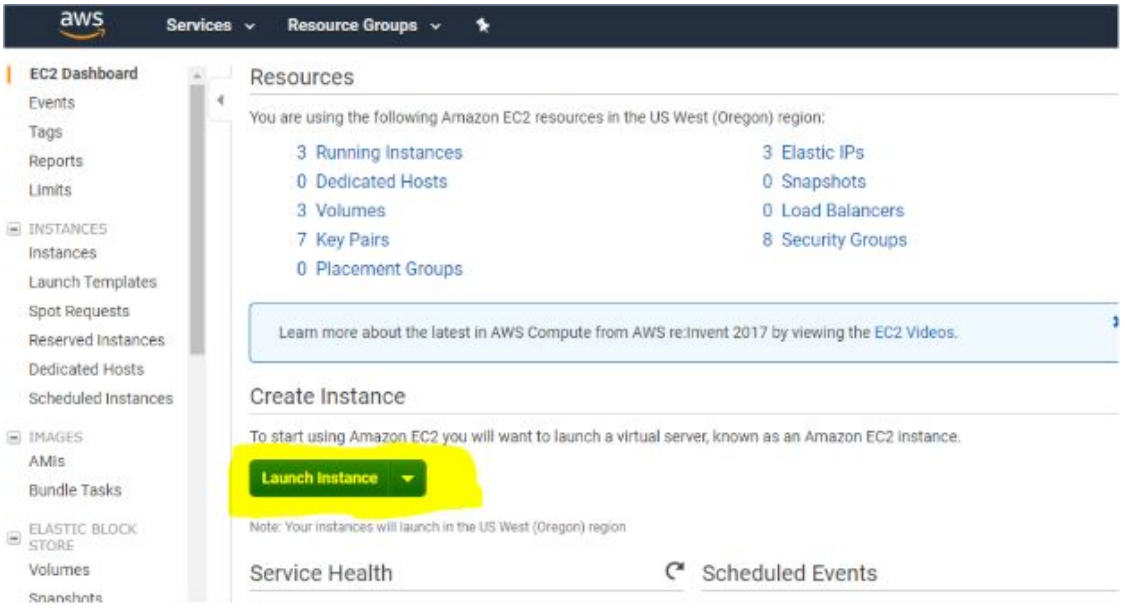
How to set up AWS Instance for this bootcamp?

Creating an EC2 Instance on AWS for Model Training & API Deployment

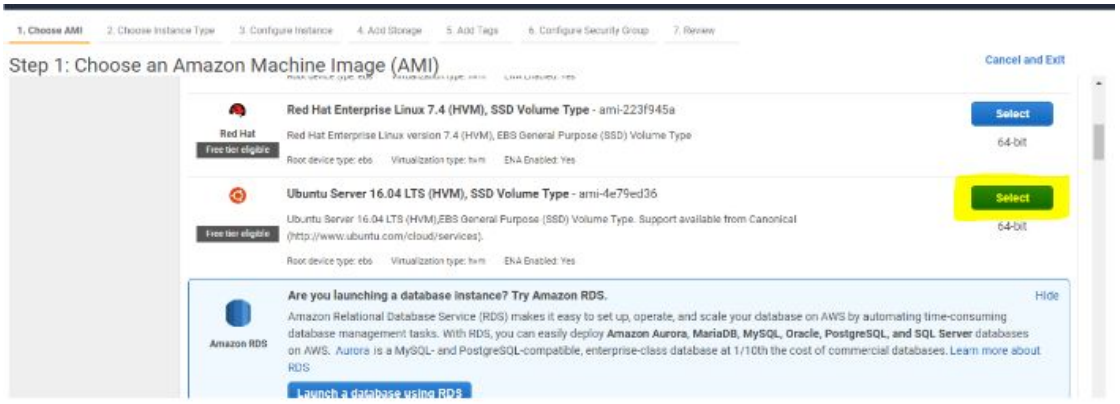
Step1. Select the EC2 option from the services dropdown on your aws dashboard.



Step 2. On the next screen of EC2 Dashboard. Select the Launch Instance Option

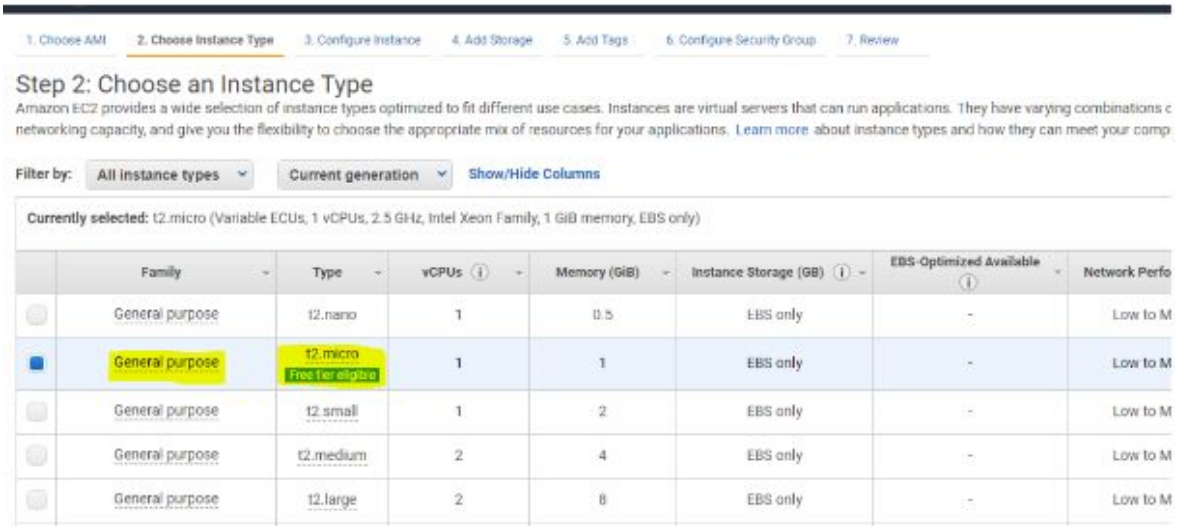


Step 3. In this bootcamp we are using Ubuntu Server 16.04 based machine, so select that from the list of choices for machine image.

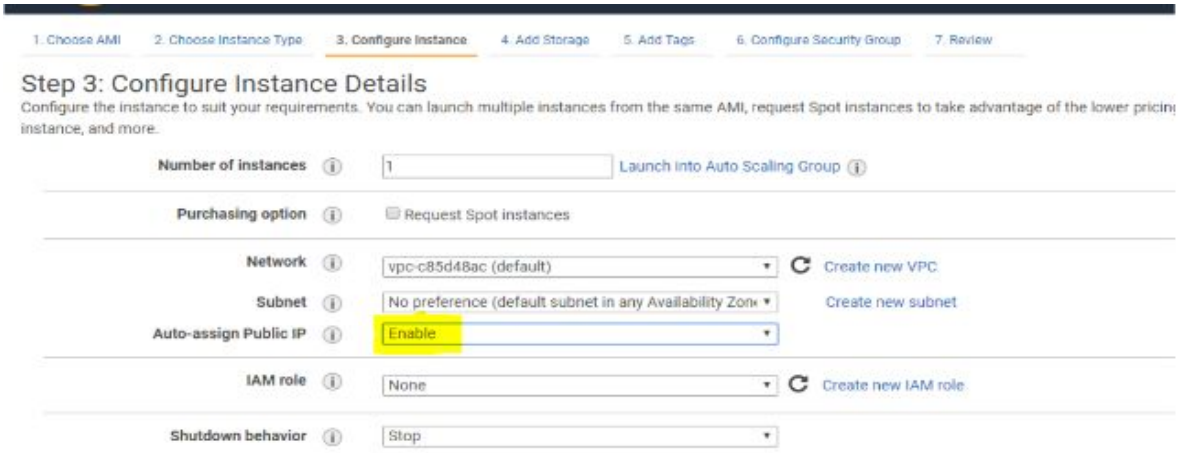


Step 4. Now select the instance type - t2.micro (It comes under the Free Tier Label).

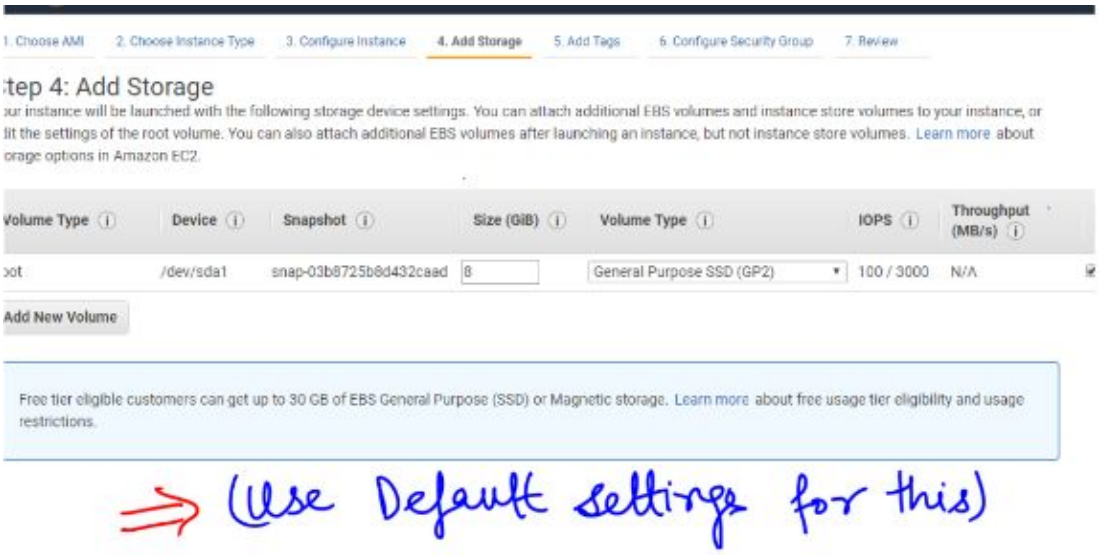
NOTE: One thing to note here is that you might have to change (upgrade) your machine at a later stage when we would be training the model as t2.micro would not be able to handle the load for that. The good news is you can perfectly do that by just stopping the machine through EC2 Dashboard and upgrading it to your specific configuration and then restarting the machine. Don't worry I will tell you how to do this when you would require it.



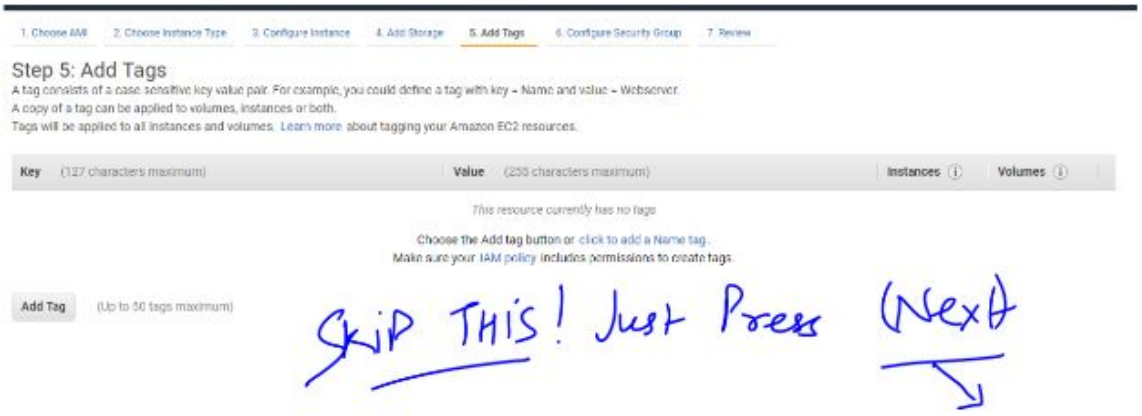
Step 5. Enable Auto Assign IP - to open access to the machine through the internet.



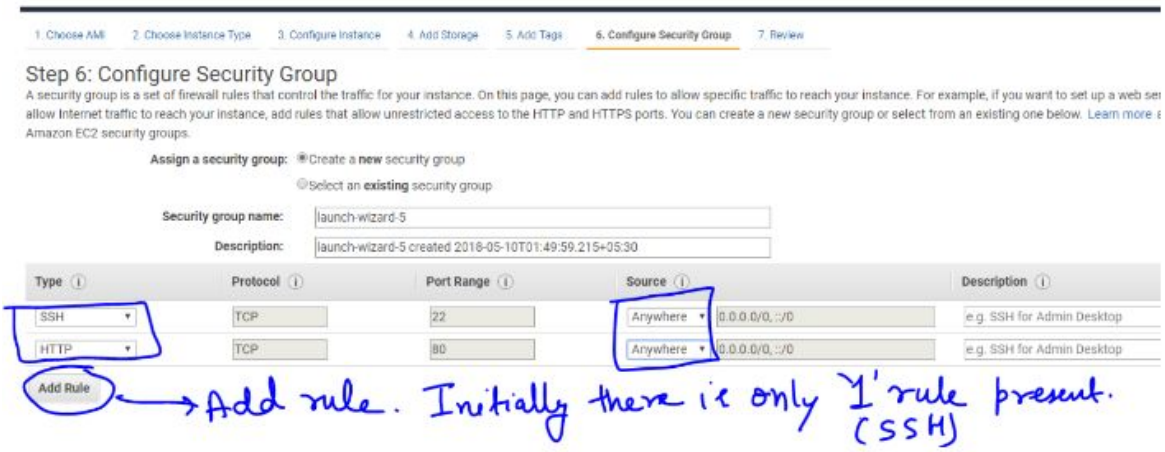
Step 6. For this step of adding storage, just set the Default Settings and continue.



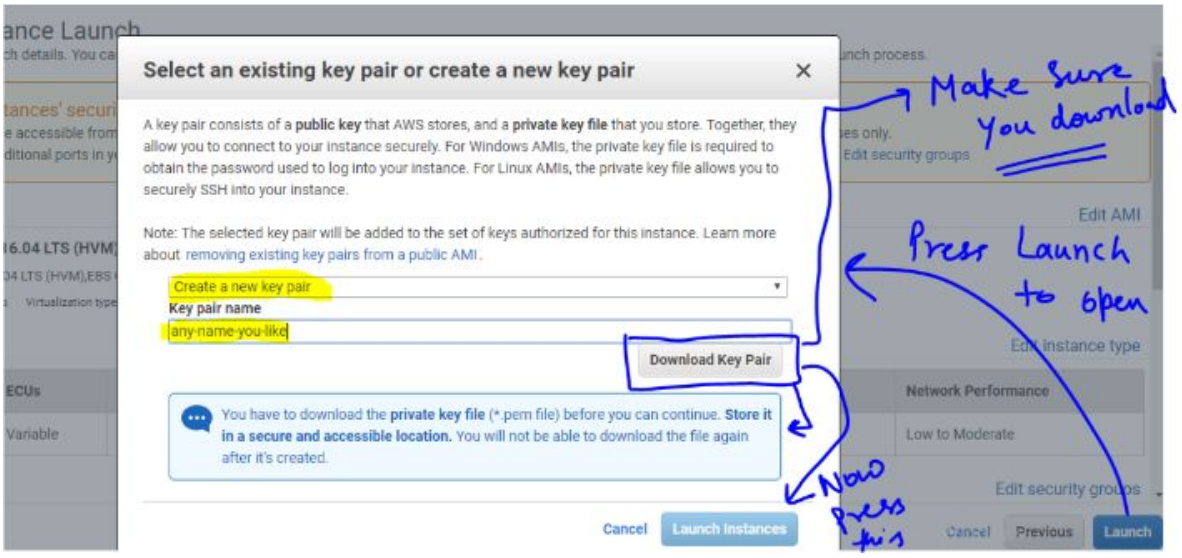
Step 7. In the next step of adding tags, again set the default settings and continue.



Step 8. Configure the security groups as shown below to have 2 rules - SSH & HTTP



Step 9. Save the .pem key to access the EC2 Machine and then launch.



Step 10. Set up an elastic IP for the instance. Elastic IP - remains the same (whereas the ip which comes associated by default with the instance changes) when you restart the instance, thus giving you better control over the instance.

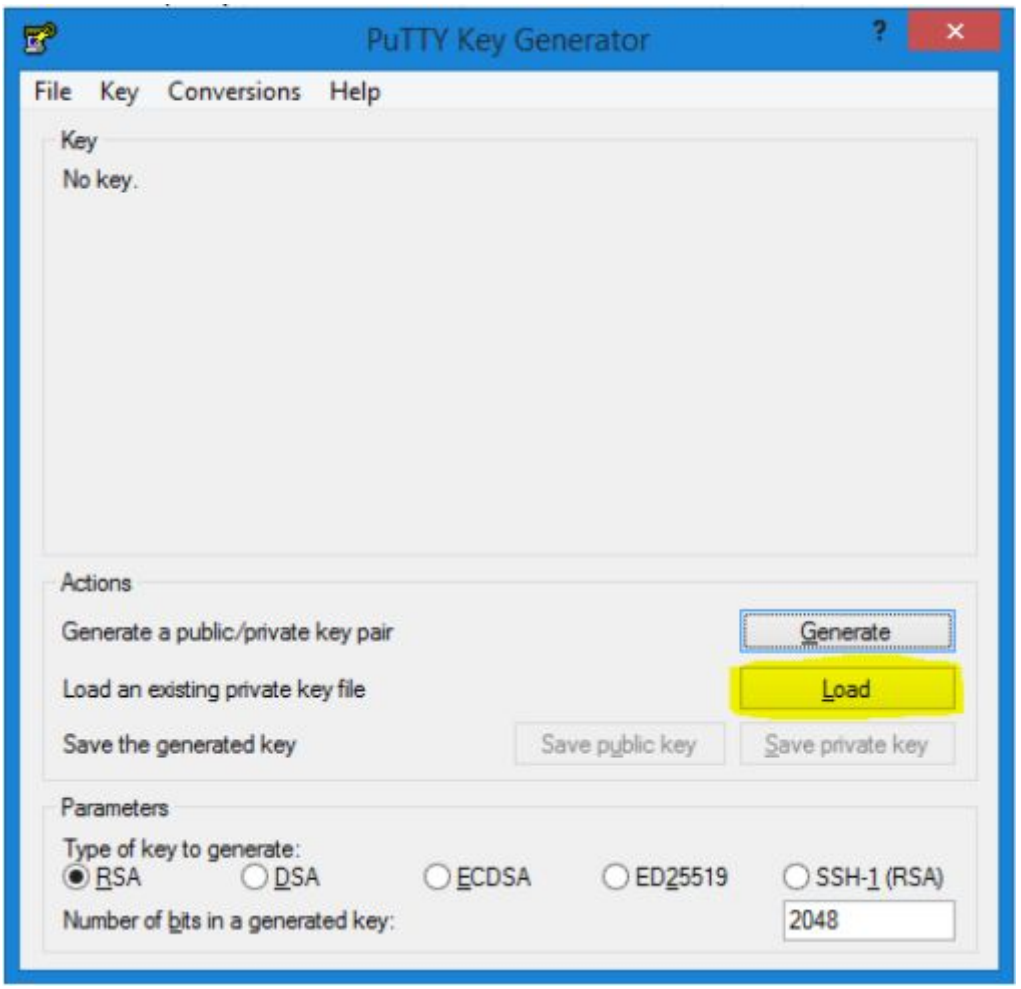
Watch the Elastic-IP-CREATION.gif at the following url to see how it's done - https://bit.ly/Elastic-IP-CREATION

Step 11. Associate the elastic IP Address created in the last step with your ec2 instance.

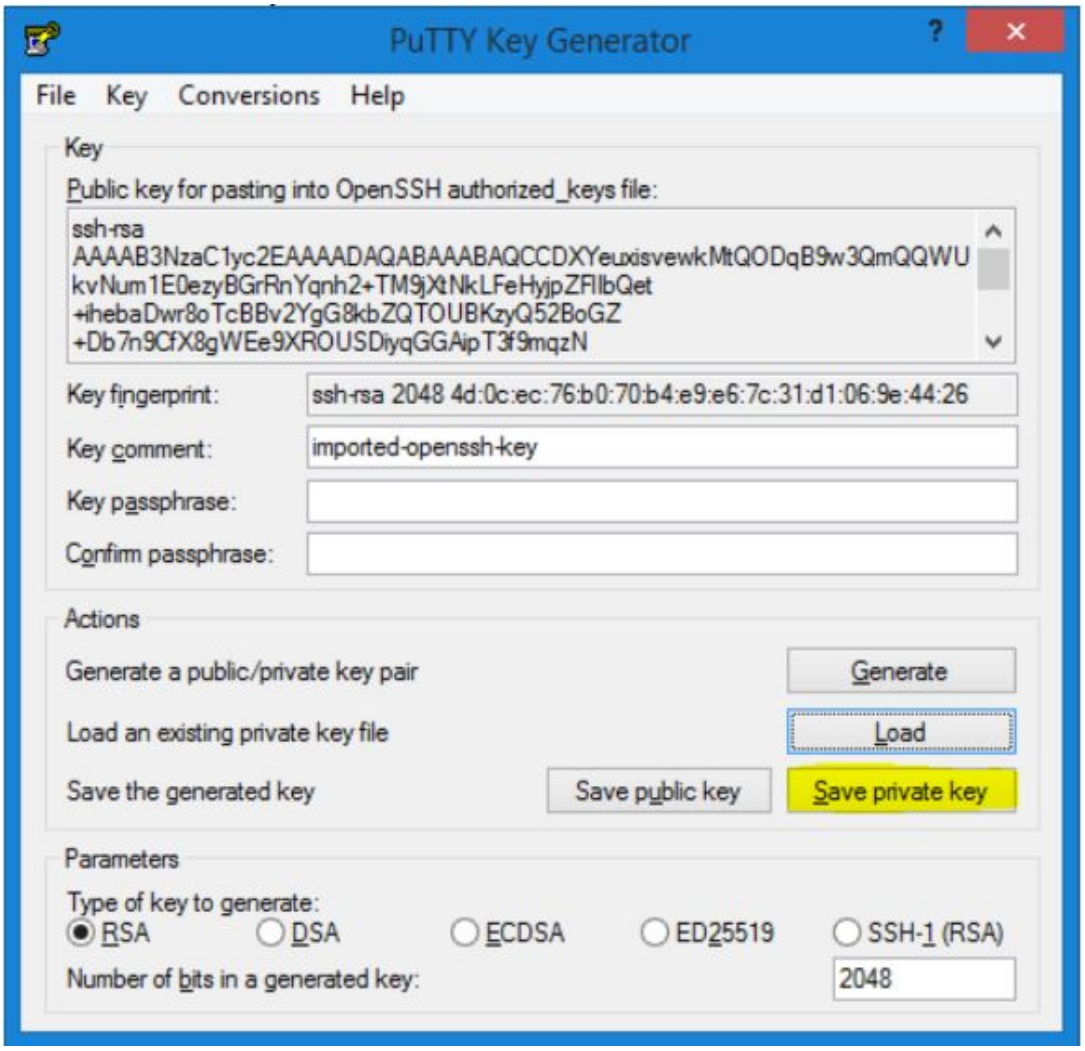
Watch the Elastic-IP-ASSOCIATION.gif at the following url to see how it's done - https://bit.ly/Elastic-IP-ASSOCIATION

How to connect to EC2 Instance using Putty & Puttygen ?

Step 1. Convert key from .pem to .ppk. To do this you need to start puttygen and press the highlighted load button in puttygen. You need to load your .pem key file which you downloaded from aws while creating the EC2 Instance.

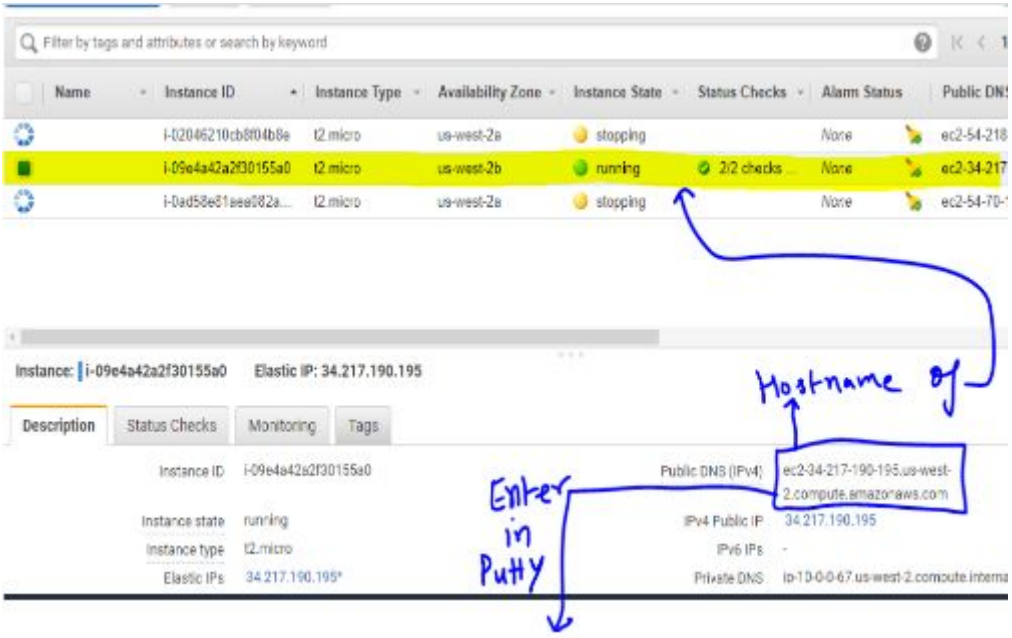


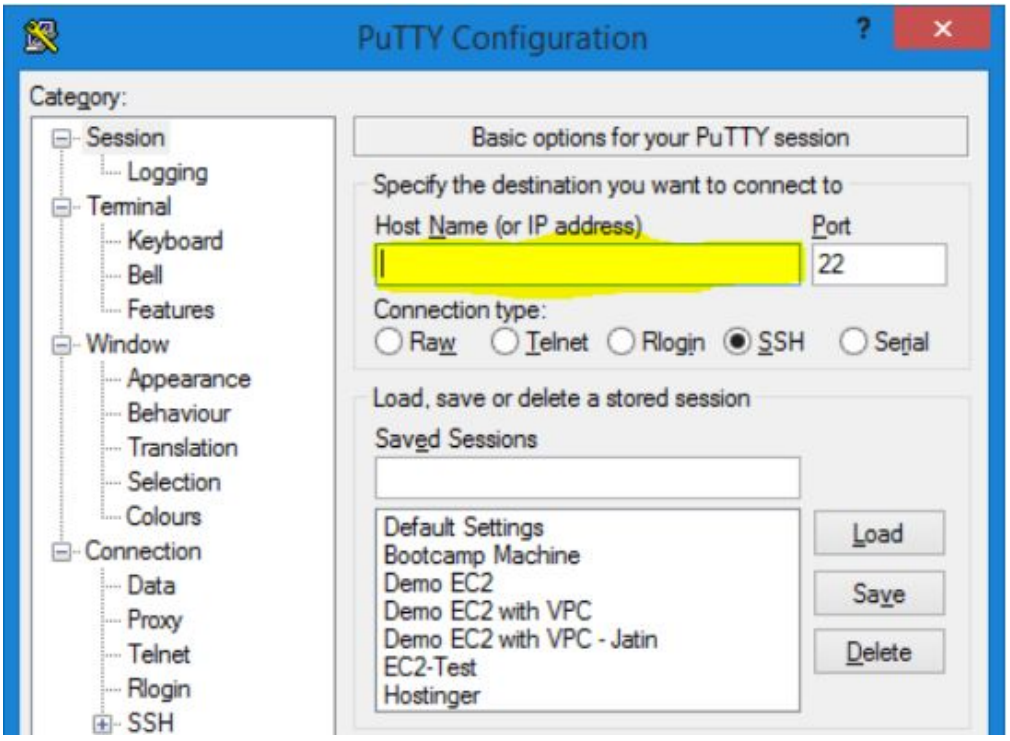
Step 2. Then you need to save the private key by pressing the “Save Private Key” Button.



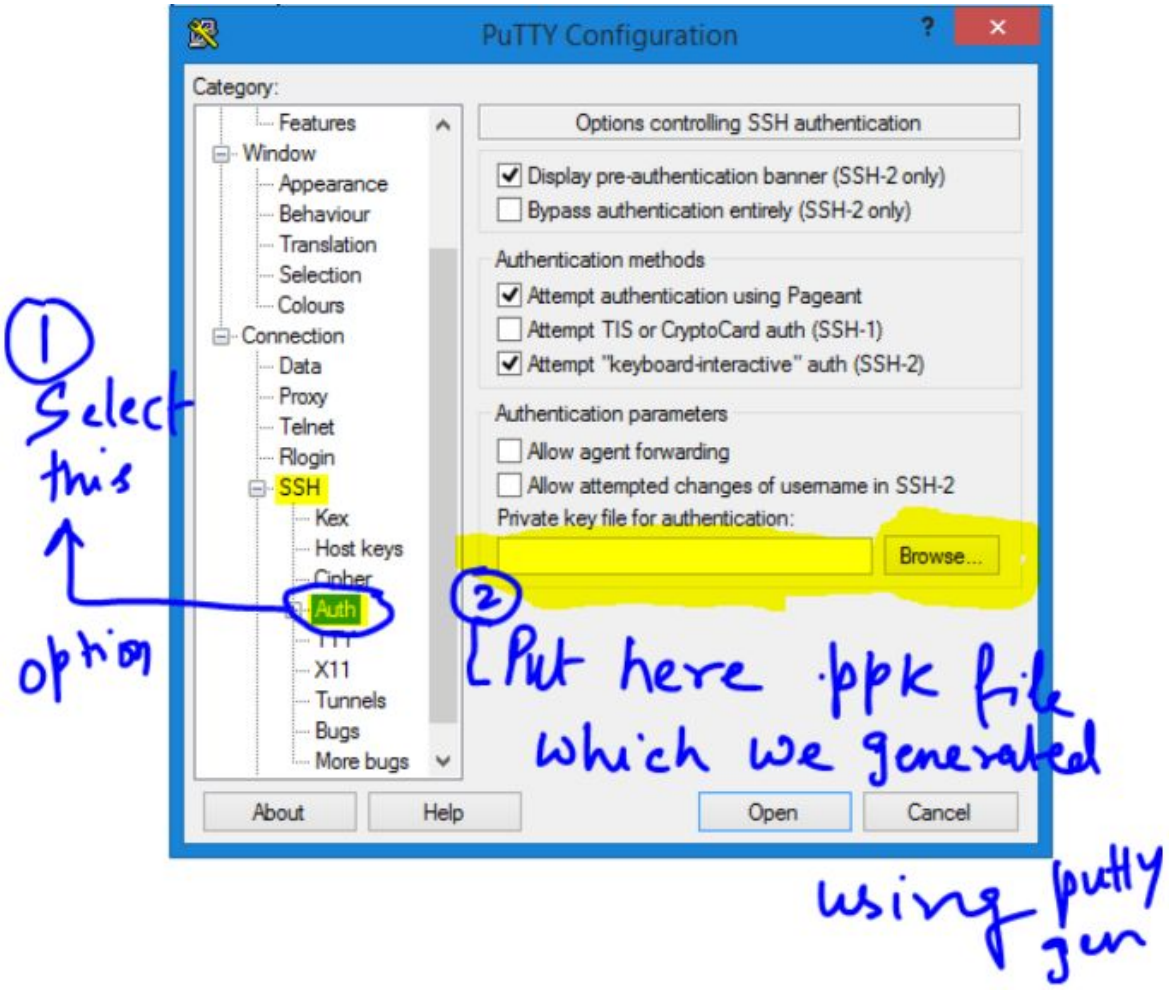
Step 3. You need to save it without the paraphrase. So ignore the warning.

Step 4. Setup Putty by entering the hostname (i.e the url of your instance). You can take a look at the following image and see how you can extract the hostname (or url) of the instance.

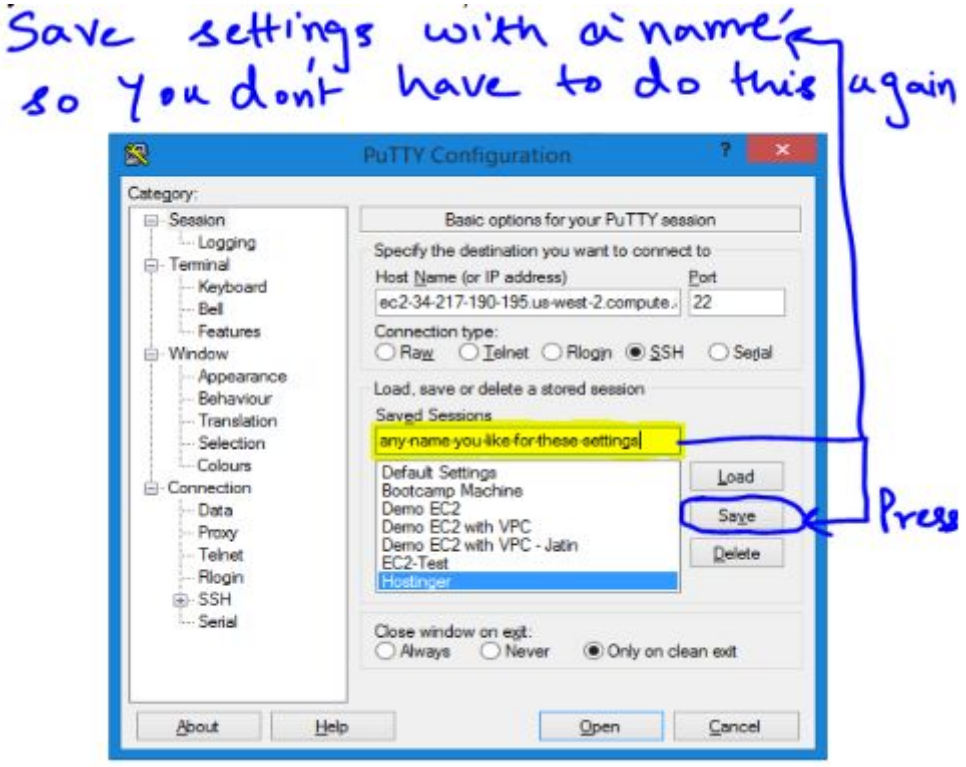




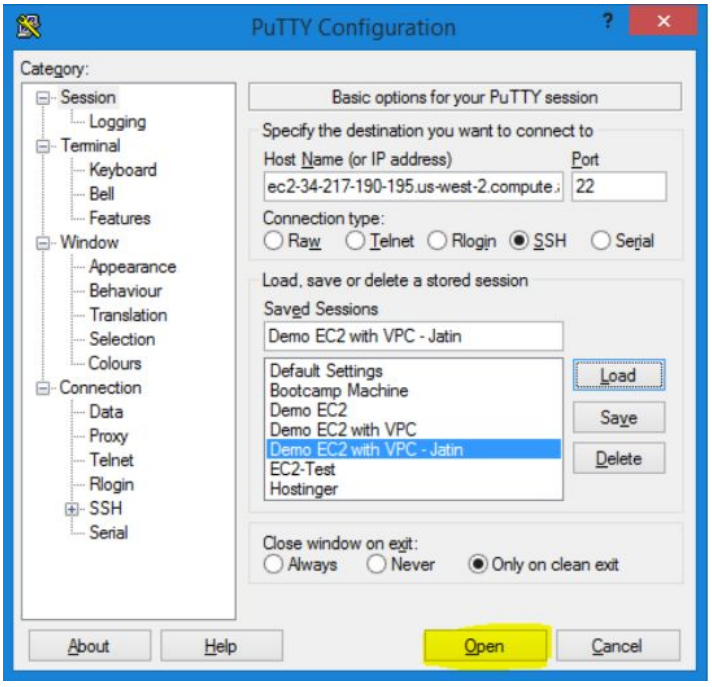
Step 5. Set the .ppk key in putty in the field shown below.



Step 6. Save these settings in putty, so as to avoid this procedure again and again. (Note: Once you save your settings they will be available under saved sessions and you can select and load them)



Step 7. Open the Connection to check if it's working fine or not.



How to set up Filezilla for File Transfer

Step 1. Add new site in site manager
Step 2. Add 'Amazon AWS key (.pem)' in Edit->Settings->SFTP
Step 3. Connect to the new site added using the site manager

Please watch the Filezilla-Setup.gif from the following url for step by step approach on how to connect to an ec2 machine in filezilla for file transfer
https://bit.ly/dg-bootcamp-filezilla-setup

How to configure Apache server on EC2

Step 1. Install apache server using the following command

```
sudo apt-get install apache2
```

Step 2. Install apache webserver gateway interface using the following command

```
Sudo apt-get install libapache2-mod-wsgi
```

Visit https://<your-ec2-hostname-url>.com to see if everything is working fine. If you installed it correctly you will see a apache welcome page.

How to set up the Flask app on EC2 Instance

Step 1. Open the terminal & create the following directory

> /var/www/FlaskApplications

Step 2. Now create the following directory

> /var/www/FlaskApplications/SampleApp

Step 3. Change the permissions of both the above directories using the following commands

1. sudo chown -R ubuntu:ubuntu /path/to/directory
2. sudo chmod -R 777 /path/to/directory

NOTE: /path/to/directory (obviously ;)) refers to the path of directory/file of which you want to change the permissions. We change the permissions so that we and apache (and in return the internet) are able to write at that location.

Step 4. Change the hostname in SampleApp.conf to match your EC2 Instance Hostname (or url)

Step 5. Place the SampleApp.conf file at

> /etc/apache2/sites-available/SampleApp.conf

Step 6. Place the .wsgi file at

> /var/www/FlaskApplications/

Step 7. Place the demo.py file at

> /var/www/FlaskApplications/SampleApp/api

Step 8. Execute the following command to initialize the apache server and the api

1. sudo a2enmod wsgi
2. sudo apachectl restart
3. sudo a2ensite sampleApp

Step 9. Execute the following command to start the server

1. sudo service apache2 reload
2. sudo /etc/init.d/apache2 reload