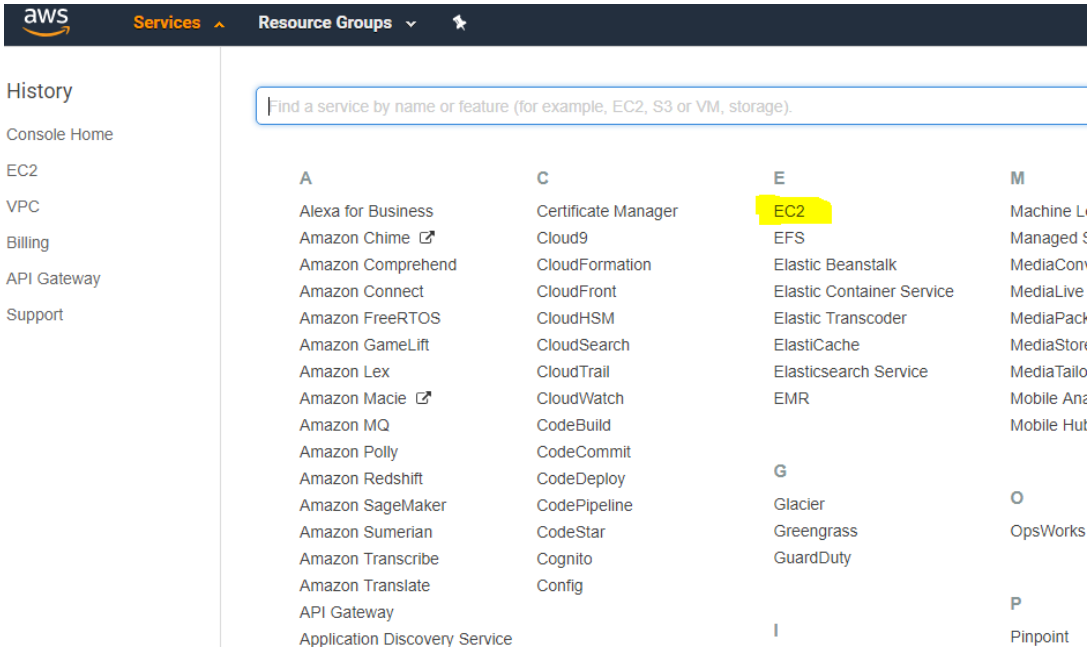
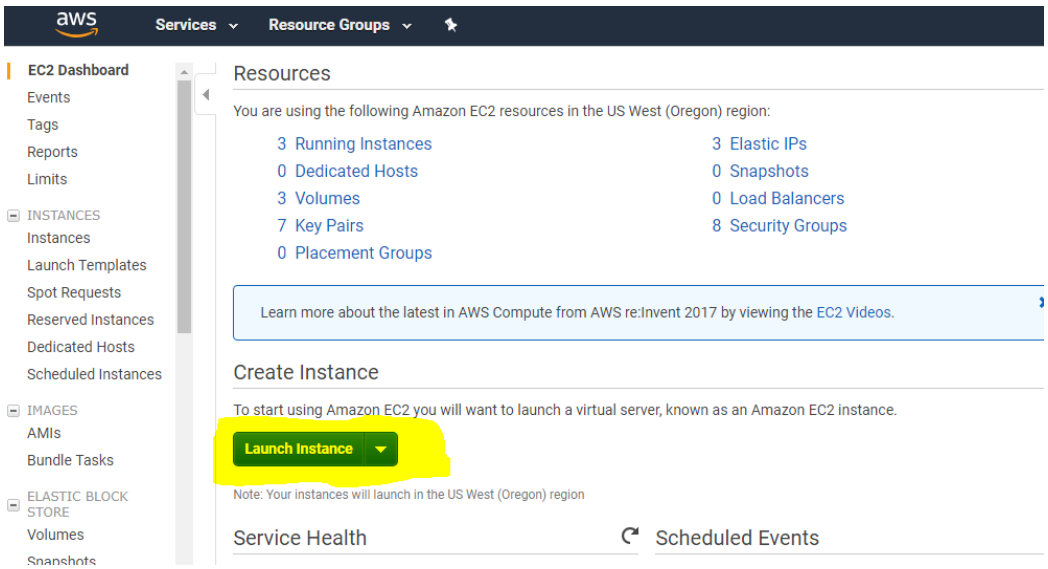
AWS and API Setup

Step 1 Create EC2 Machine

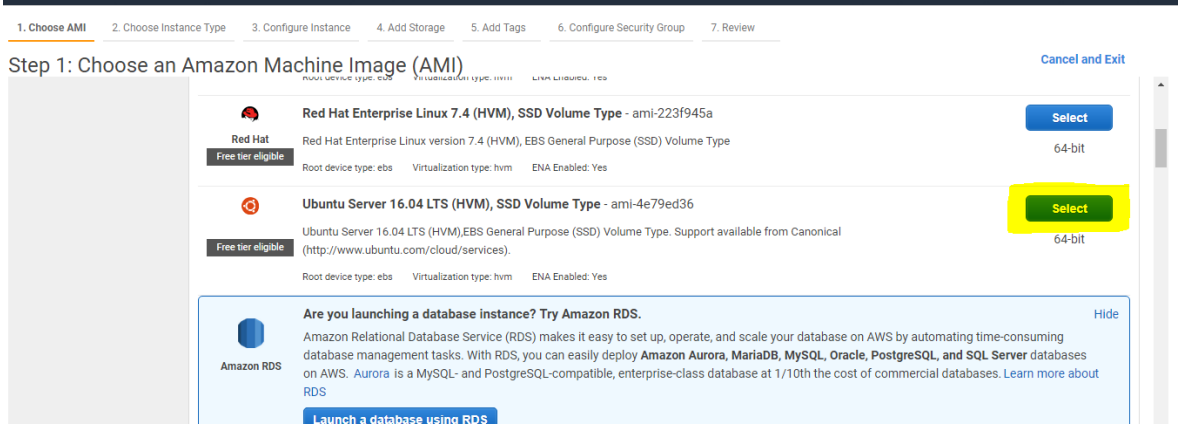
1. Select the EC2 Option from the services dropdown on your aws dashboard



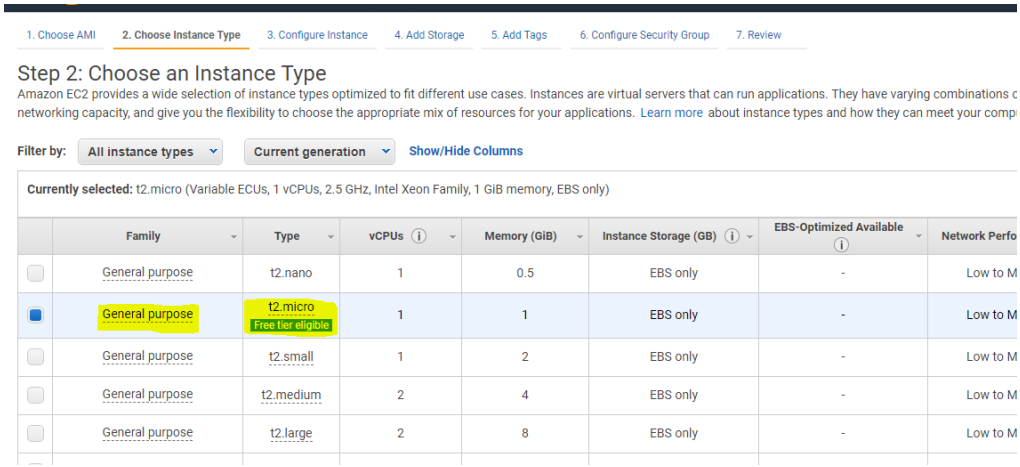
2. On the next screen of EC2 Dashboard, Select the Launch Instance option



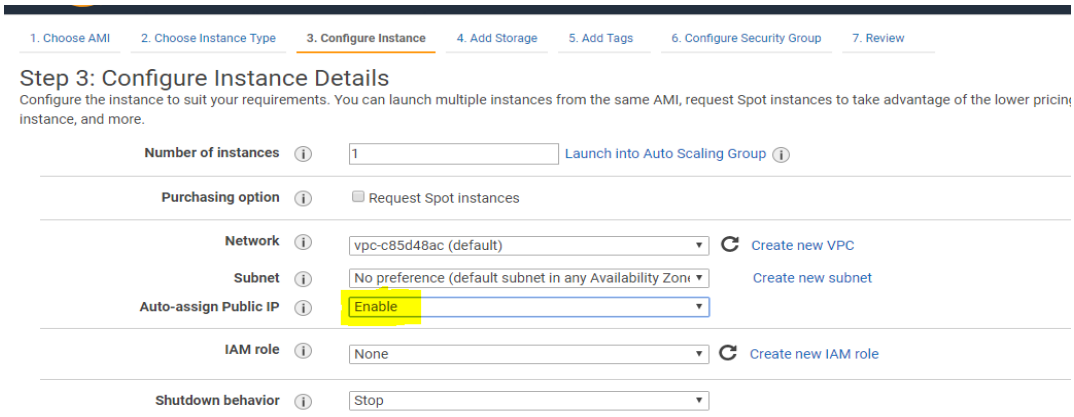
3. In this bootcamp we are using UBUNTU Server 16.04 Machine.



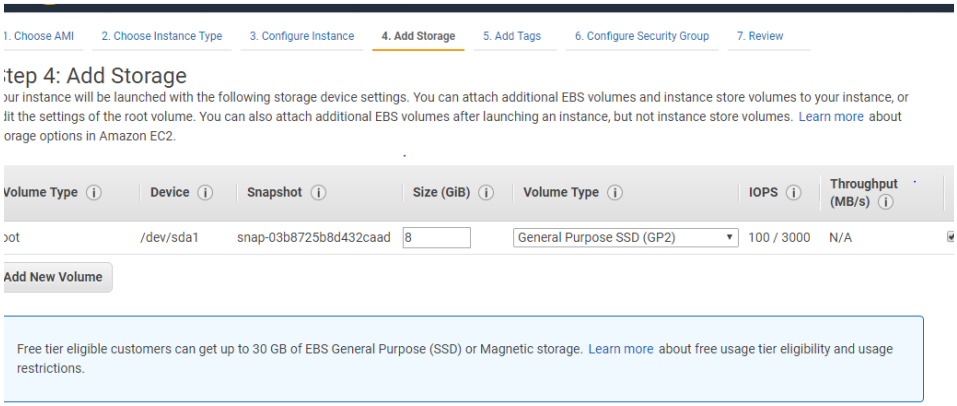
4. Now select the instance type - T2.Micro (It comes under the free tier label)



5. Enable auto assign IP – to open access from Internet

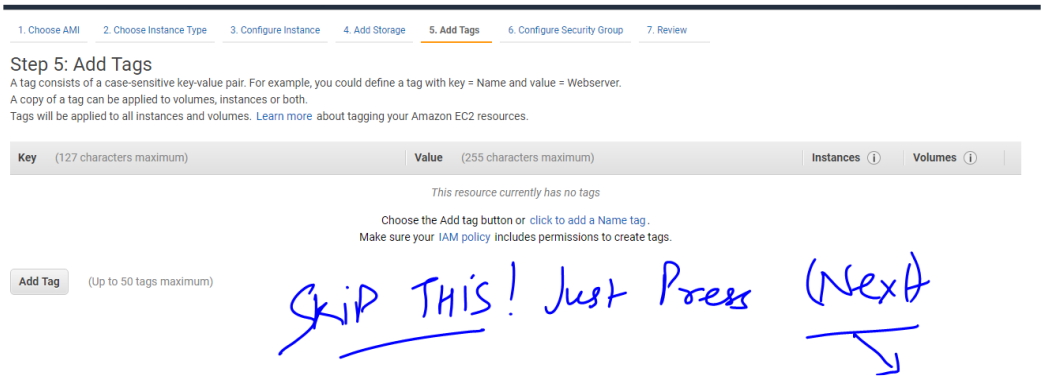


6. Set the default settings

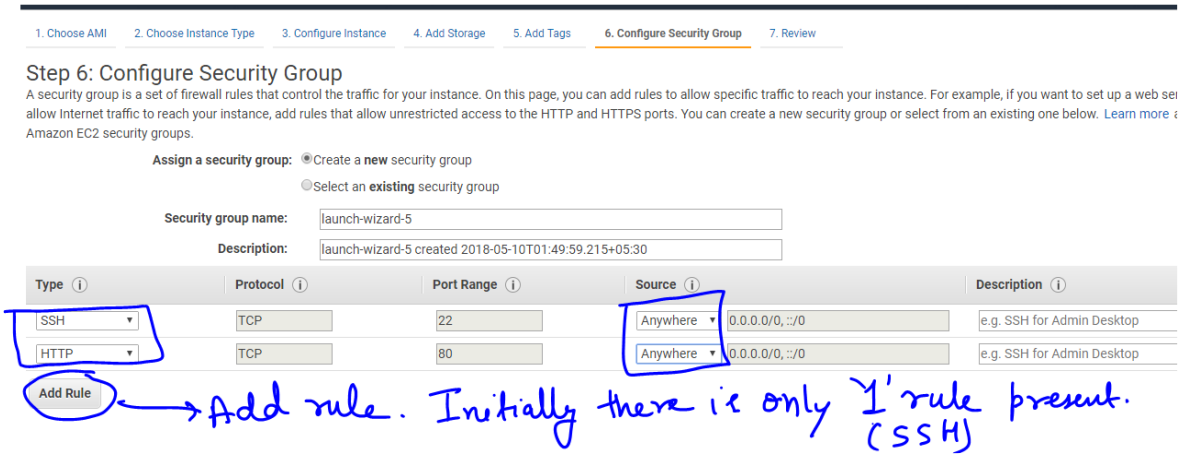


⇒ (Use Default settings for this)

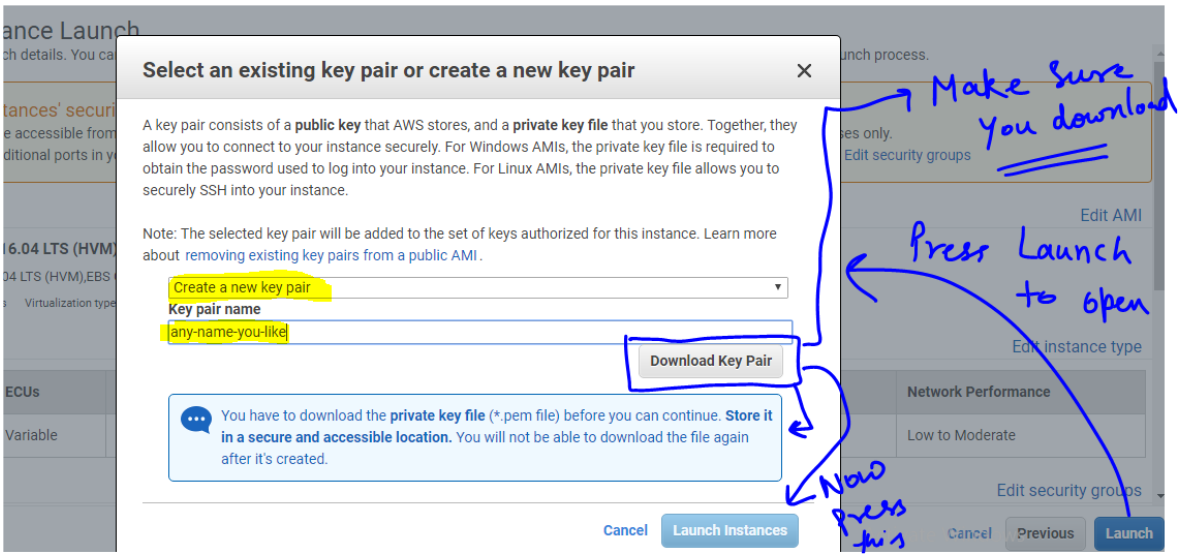
7. In next step “Add Tags” tab, Set the default settings



8. Configure the security groups as shown below



9. Save the .pem key to access the ec2 machine and then launch.



10. Set up an elastic ip for the instance. Elastic IP - remains same (whereas the ip which comes associated by default with the instance changes) when you stop the instance and thus gives you a better control over the instance.

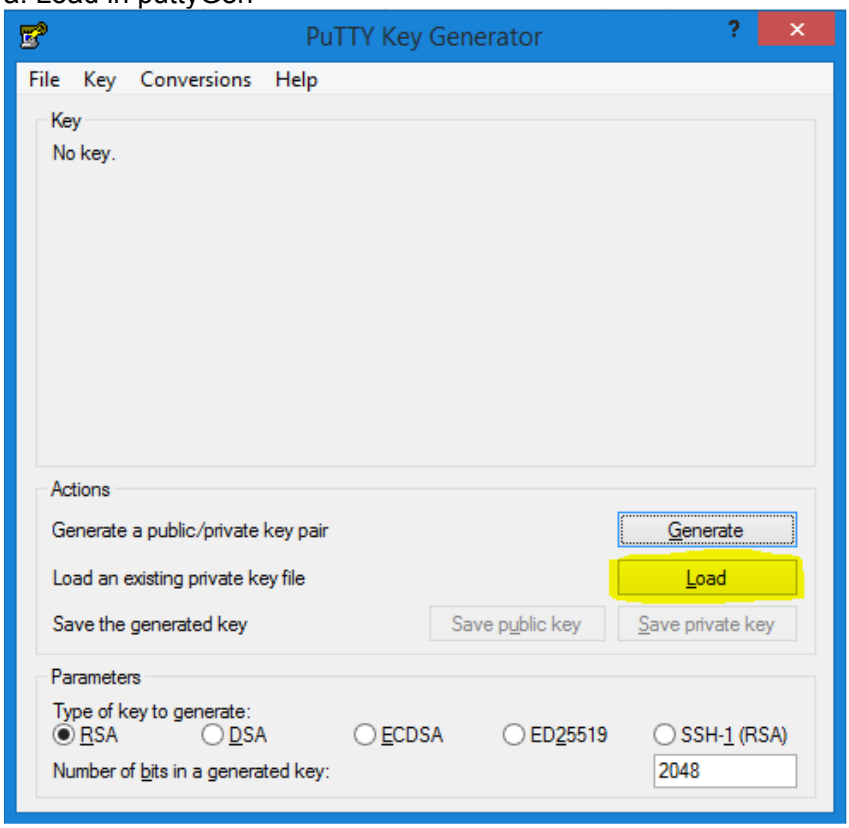
(Watch the ELASTIC-IP-CREATION.gif in files to see how its done)

11. Associate the above elastic IP Address with your ec2 instance.

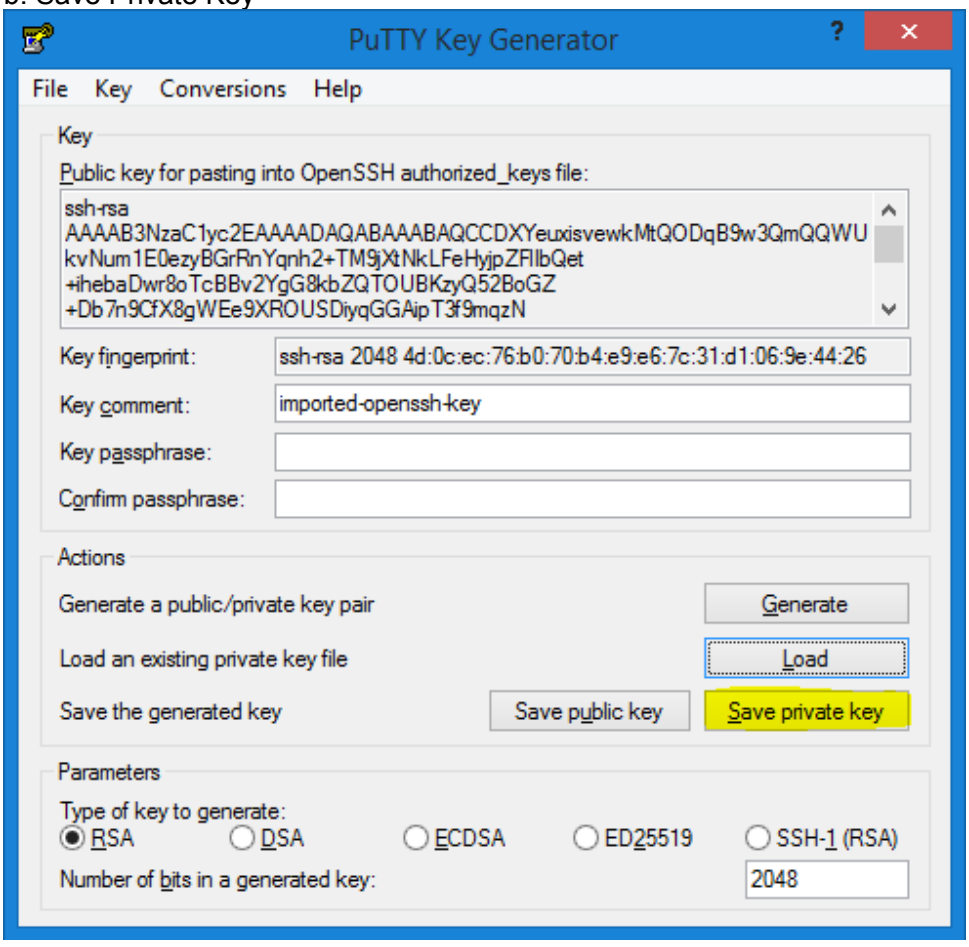
(Watch the ELASTIC-IP-ASSOCIATION.gif in files to see how its done)

Step 2 Connect to instance using Putty

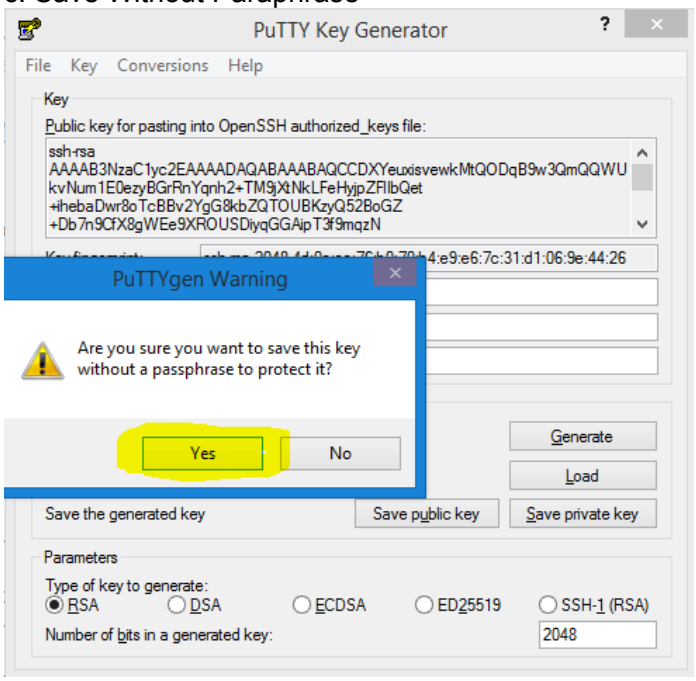
1. Convert key to .ppk from .pem
    a. Load in puttyGen



    b. Save Private Key

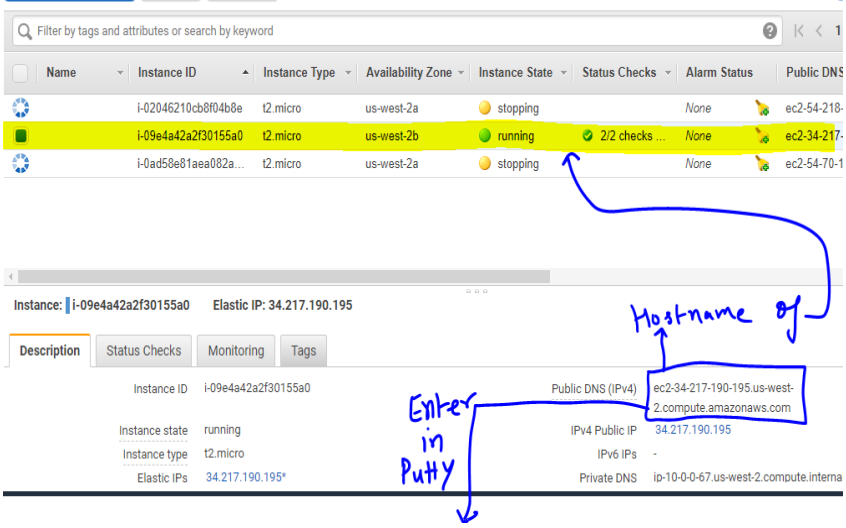


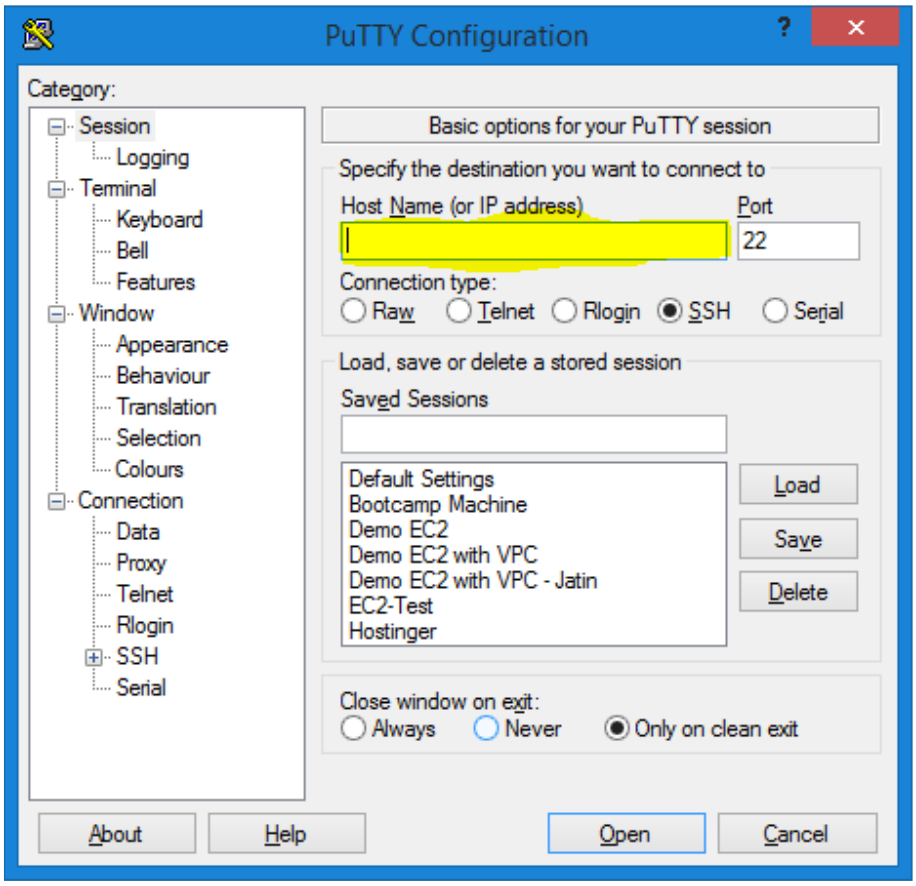
c. Save Without Paraphrase



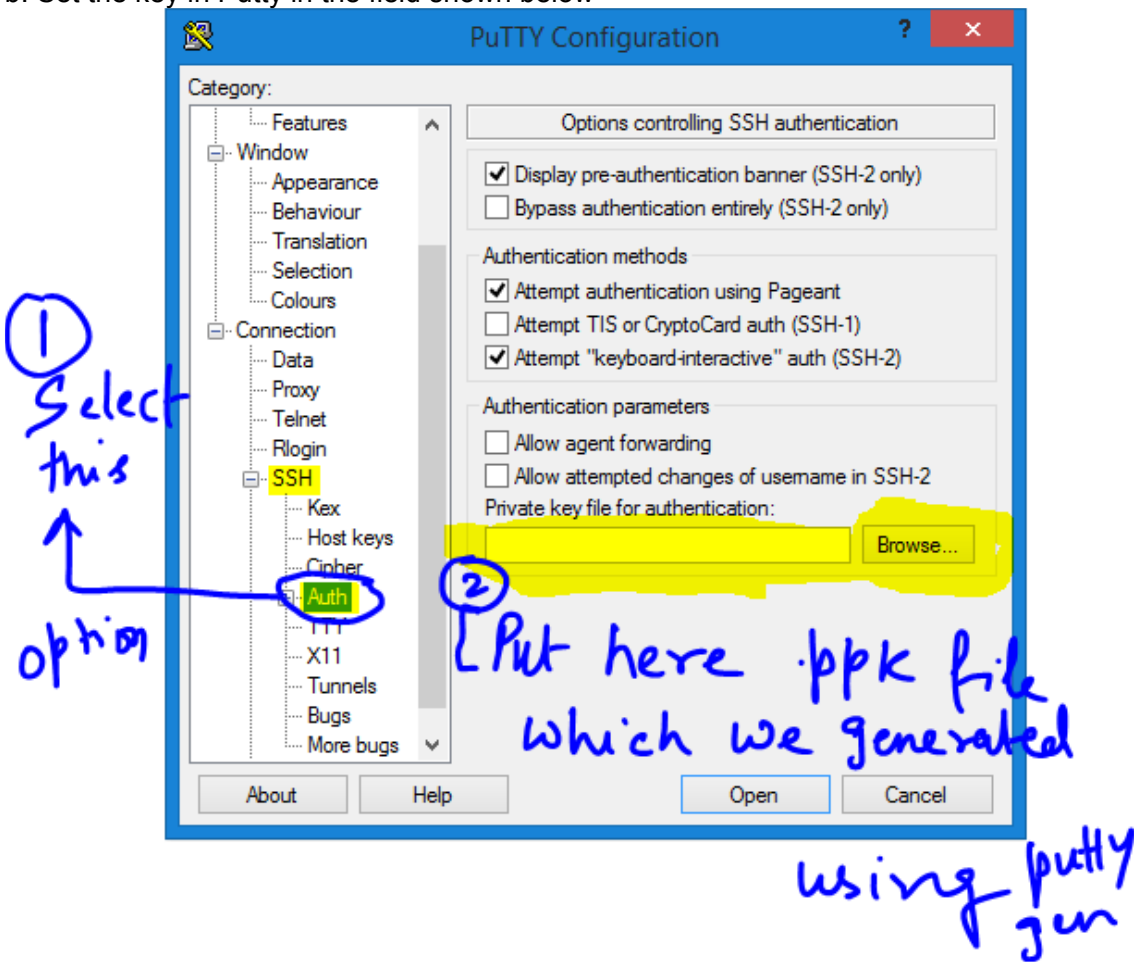
2. Setup Putty

a. Enter the Hostname to the specified field. You will get the hostname from AWS EC2 Dashboard as shown below in the figure.

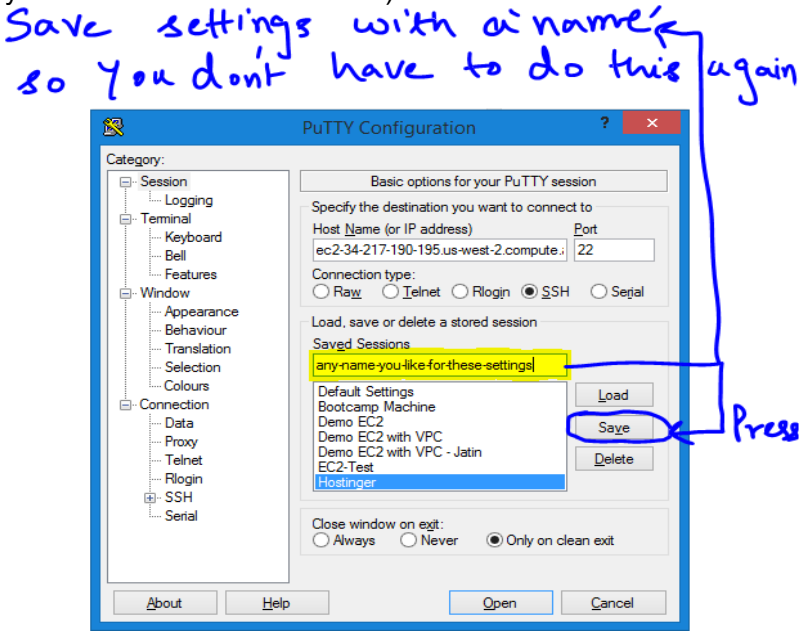

Filter by tags and attributes or search by keyword
Name
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
Alarm Status
Public DNS
i-02046210cb8f04b8e
t2.micro
us-west-2a
stopping
None
i-09e4a42a2f30155a0
t2.micro
us-west-2b
running
2/2 checks ...
None
i-0ad58e81aea082a...
t2.micro
us-west-2a
stopping
None
Instance: i-09e4a42a2f30155a0
Elastic IP: 34.217.190.195
Hostname of
Description
Status Checks
Monitoring
Tags
Instance ID
i-09e4a42a2f30155a0
Public DNS (IPv4)
ec2-34-217-190-195.us-west-2.compute.amazonaws.com
Enter in Putty
Instance state
running
IPv4 Public IP
34.217.190.195
Instance type
t2.micro
IPv6 IPs
-
Elastic IPs
34.217.190.195*
Private DNS
ip-10-0-0-67.us-west-2.compute.interna


PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
Telnet
Rlogin
SSH
Serial
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address)
Port
22
Connection type:
Raw
Telnet
Rlogin
SSH
Serial
Load, save or delete a stored session
Saved Sessions
Default Settings
Bootcamp Machine
Demo EC2
Demo EC2 with VPC
Demo EC2 with VPC - Jatin
EC2-Test
Hostinger
Load
Save
Delete
Close window on exit:
Always
Never
Only on clean exit
About
Help
Open
Cancel

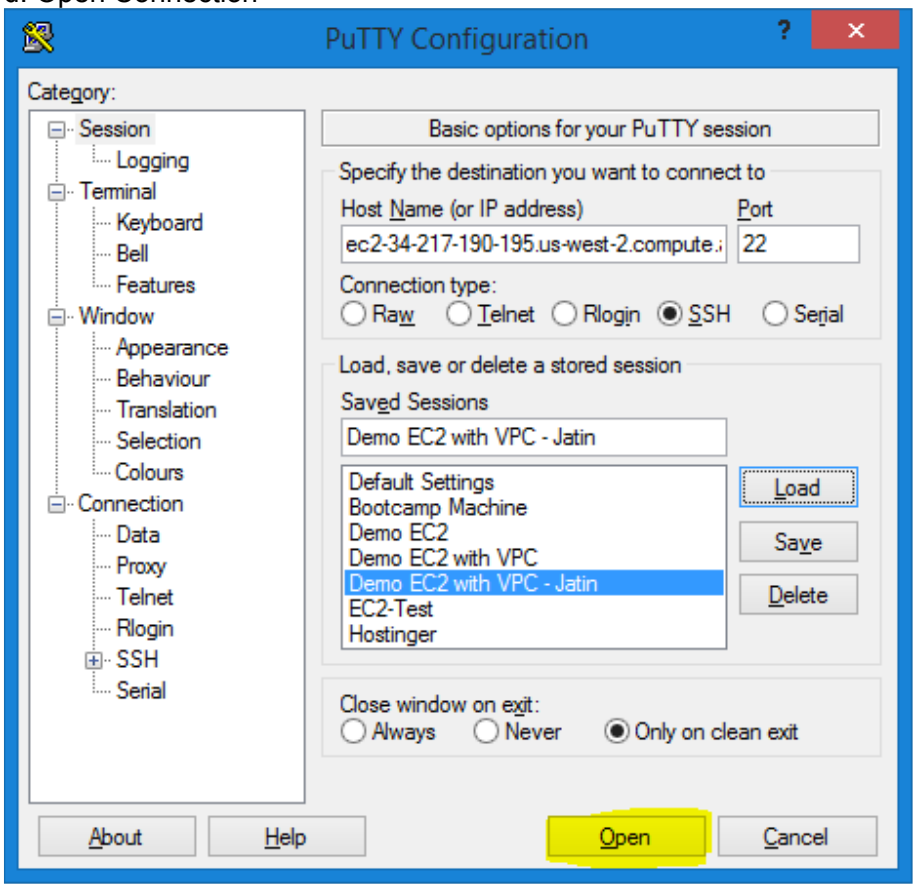
b. Set the key in Putty in the field shown below



c. Save Settings for this, 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.)



d. Open Connection



Step 3 Install Python and its dependencies

1) Install Python
    a. sudo apt-get install python 2.7

2) Install development python dependencies
    a. sudo apt-get install python-setuptools

    Note : Run sudo apt-get update if you face any error like “Failed to fetch the resource”

    b. sudo apt-get install python-dev
    c. sudo apt-get install build essential
    d. sudo apt-get install python-pip

3) Install development related libraries
    a. pip install numpy scipy sklearn
    b. sudo apt-get install python-pil
    c. sudo apt-get install python-joblib
    d. sudo apt-get install python-flask
    e. To install opencv use this command
    curl -s "https://raw.githubusercontent.com/arthurbeggs/scripts/master/install_apps/install_opencv2.sh" | bash

Step 4 Setup File Zilla for File Transfer from Local Machine to AWS EC2

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

(Please watch the Filezilla-Setup.gif for step by step approach on how to connect to an ec2 machine in filezilla for file transfer. It is present in the files)

4. Change Permissions:
    a. sudo chown –R ubuntu:ubuntu /path/to
    b. sudo chown –R 755 /path/to

NOTE: /path/to - refers to the path of which you want to change the permissions. We change the permission so that we and apache are able to write at that location.

/path/to refers to

1. /etc/apache2/sites-available/
2. /var/www/FlaskApplications/

(Please watch the Change-Permissions.gif for step by step approach on how to change permissions. It is present in the files)

Step 5 Configure Apache server on EC2

1) sudo apt-get Install apache2
2) sudo apt-get install libapache2-mod-wsgi

Run "your-ec2-hostname-url" => to => TEST if "Everything is working"

Step 6 Setting up the Flask app

1) Create Directory:
    a. /var/www/FlaskApplications

2) Create another directory:
    a. /var/www/Flaskapplications/SampleApp

3) Change Permissions:
    a. sudo chown –R ubuntu:ubuntu /path/to
    b. sudo chown –R 755 /path/to

NOTE: /path/to - refers to the path of which you want to change the permissions. We change the permission so that we and apache are able to write at that location.

4) Place the .conf file at
    a. /etc/apache2/sites-available/SampleApp.conf
    b. Change – hostname

5) Place the .wsgi file at
    a. /var/www/FlaskApplications/

6) To test the setup, Place 'demo.py' file in /var/www/Flaskapplications/sampleApp/api/

7) Run
    a. sudo a2enmod wsgi
    b. sudo apachectl restart
    c. sudo a2ensite sampleApp

8) Run
    a. sudo service apache2 reload

b. sudo /etc/init.d/apache2 reload
c. sudo service apache2 restart
d. sudo /etc/init.d/apache2 reload