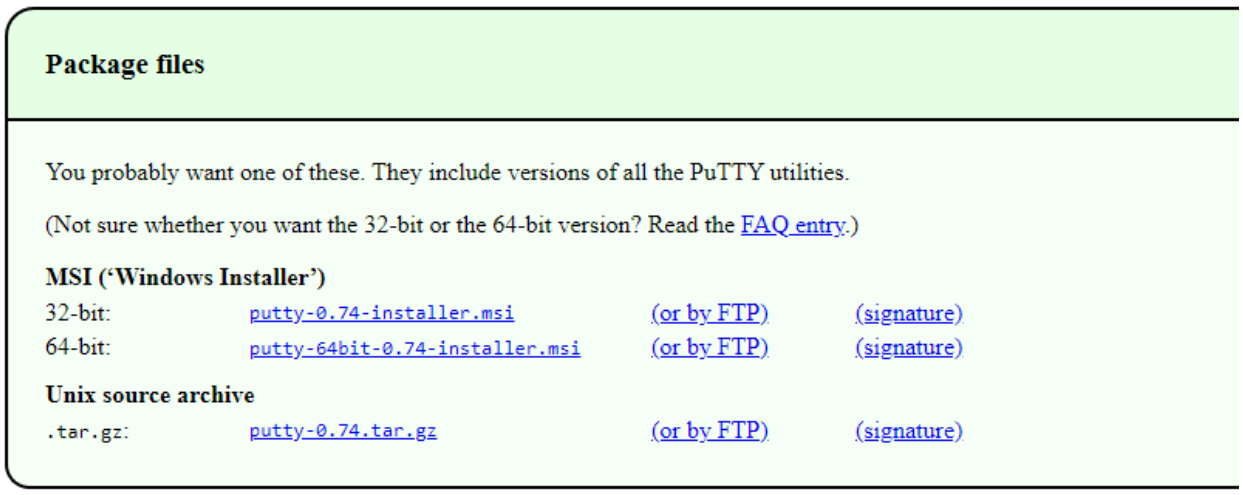
AWS EC2 Instruction

Step 0: Install PuTTY in windows

PuTTY is a free implementation of SSH for Windows. You can download the PuTTY installer from the following link: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html



Package files

You probably want one of these. They include versions of all the PuTTY utilities.

(Not sure whether you want the 32-bit or the 64-bit version? Read the FAQ entry.)

MSI ('Windows Installer')

32-bit:	putty-0.74-installer.msi	(or by FTP)	(signature)
64-bit:	putty-64bit-0.74-installer.msi	(or by FTP)	(signature)

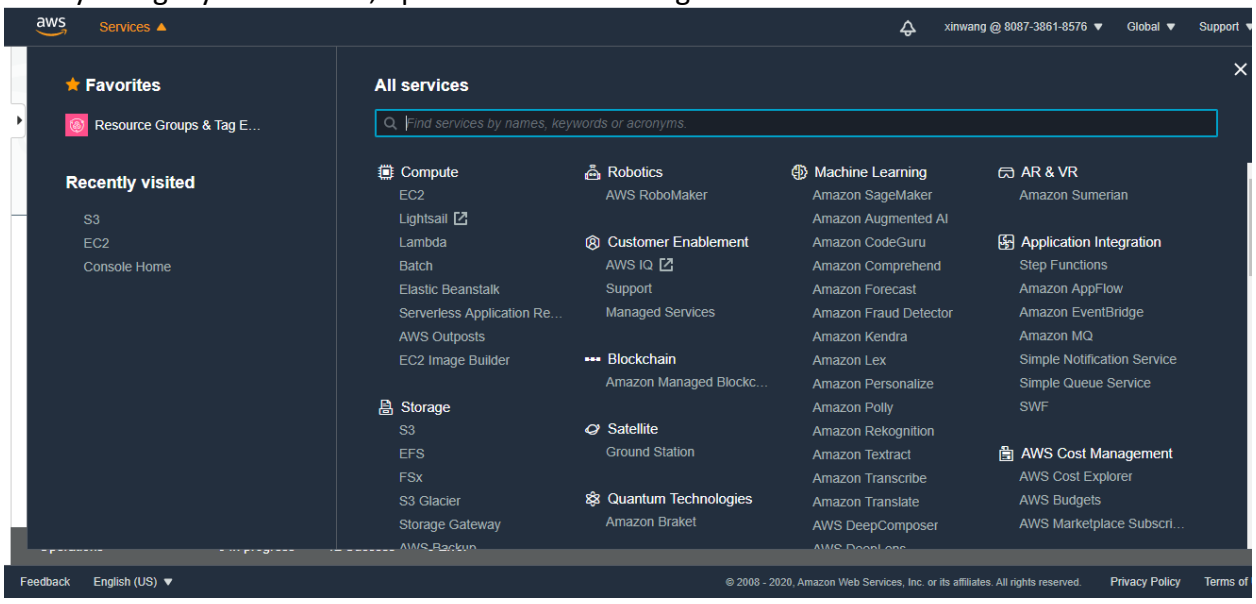
Unix source archive

.tar.gz:	putty-0.74.tar.gz	(or by FTP)	(signature)

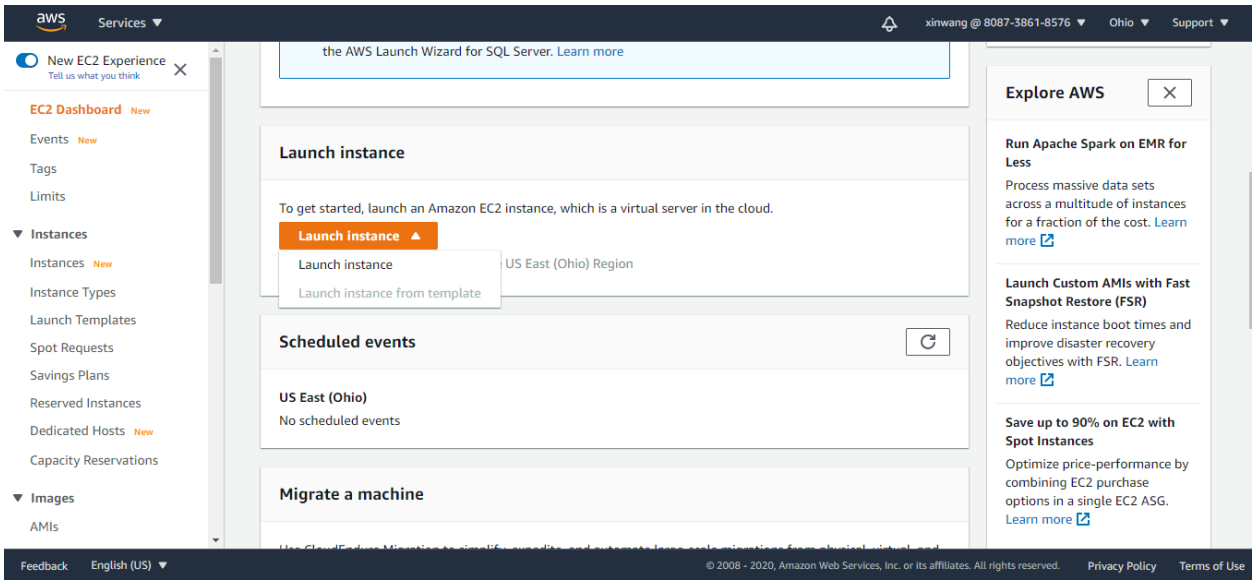
Choose the version you need for your windows system accordingly.

Step 1: Login in Amazon AWS and launch an EC2 instance.

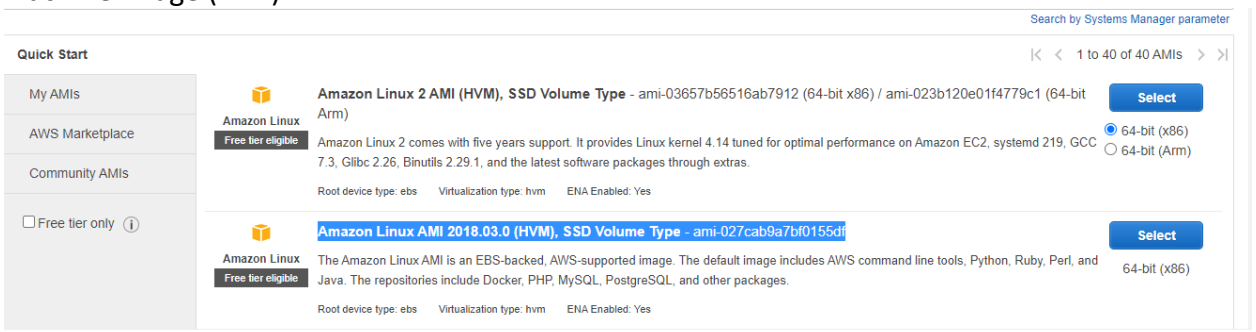
After you login your account, open the services and go to EC2 dashboard.



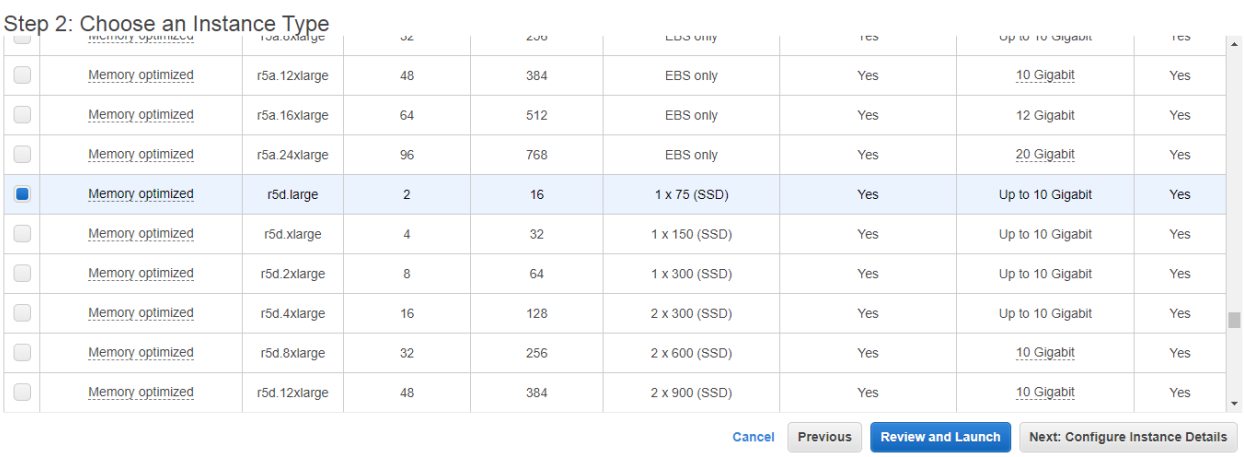
Click "Launch instance" under the drawer of "Launch instance".



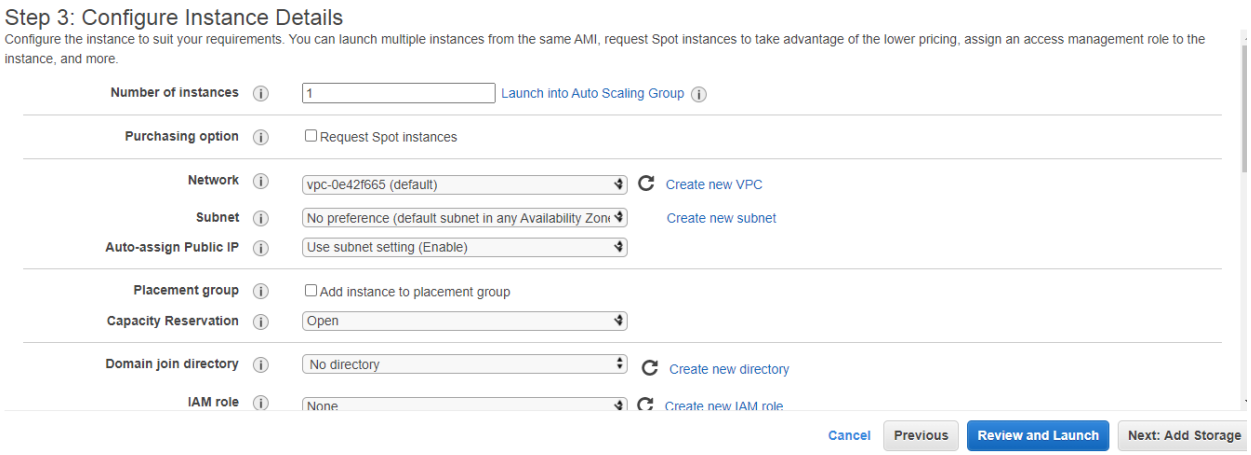
Select “**Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type** - ami-027cab9a7bf0155df” as Amazon Machine Image (AMI).



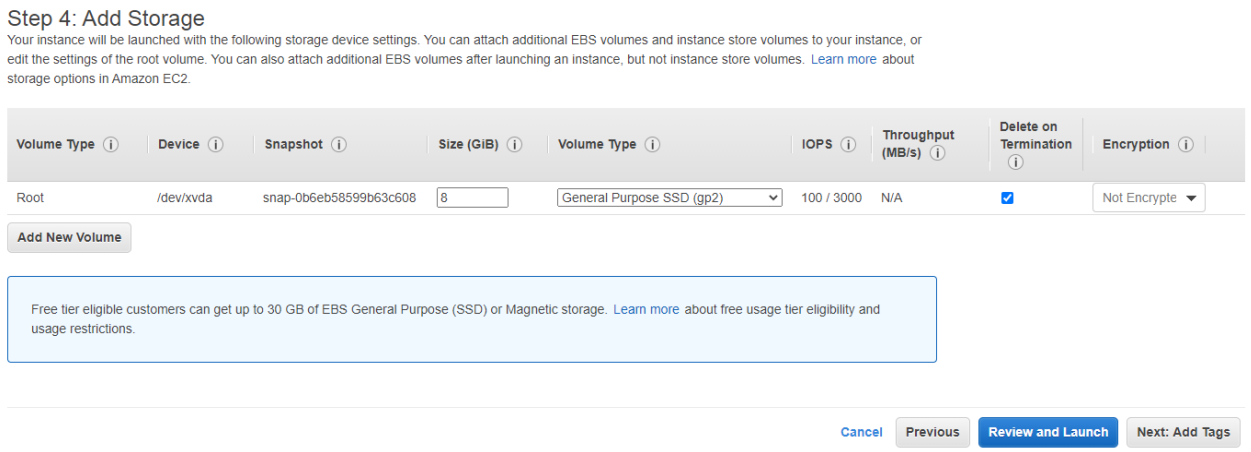
Select “r5d.large” as your instance type and then click “Next: Configure Instance Details”.



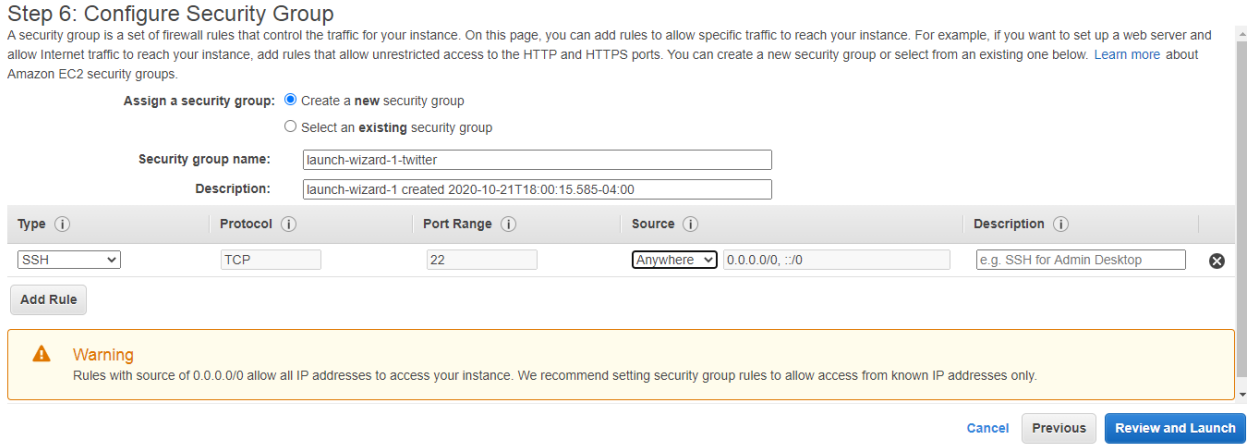
Don’t change any setting and click “Next: Add storage”.



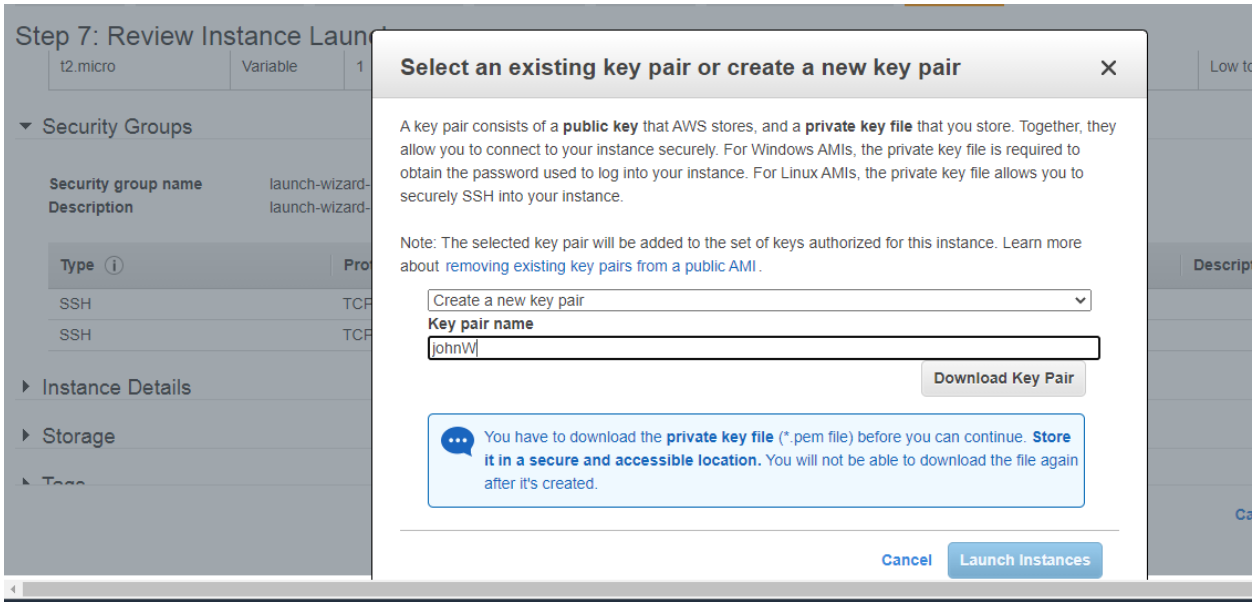
Click “Next: Add Tags



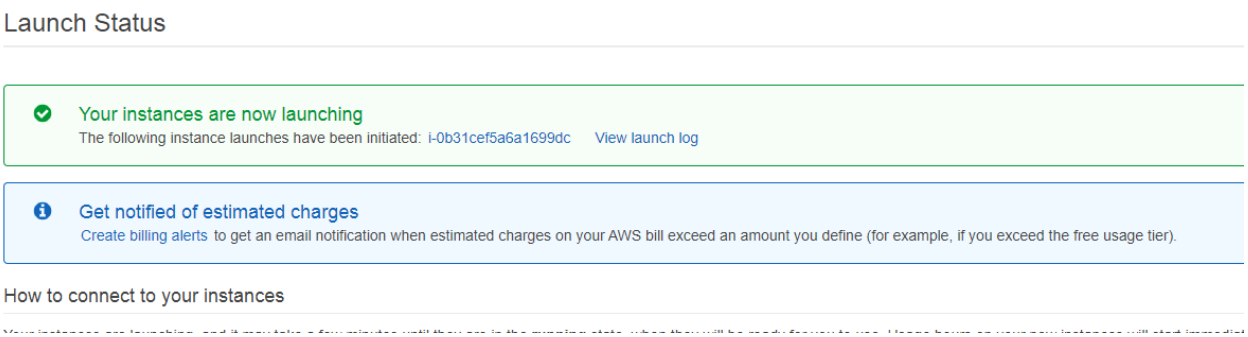
dg



Move to next page and click “Launch”. Choose “Create a new key pair” and give a key pair name in the pop-up window.



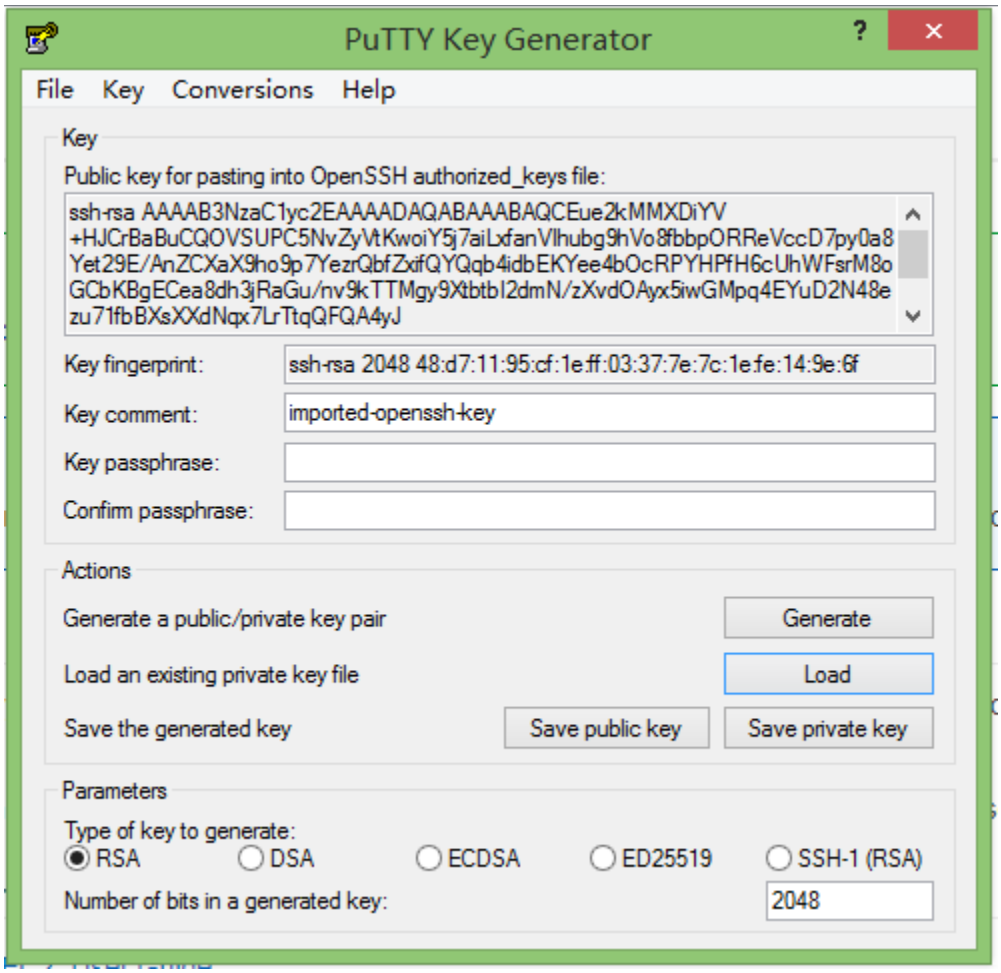
And then click “Download Key Pair”. Please keep your key pair in a safe place. It will be used to connect to EC2 from the local machine. After you download the file, click “ Launch Instances”.



Now you have launched EC2 instance.

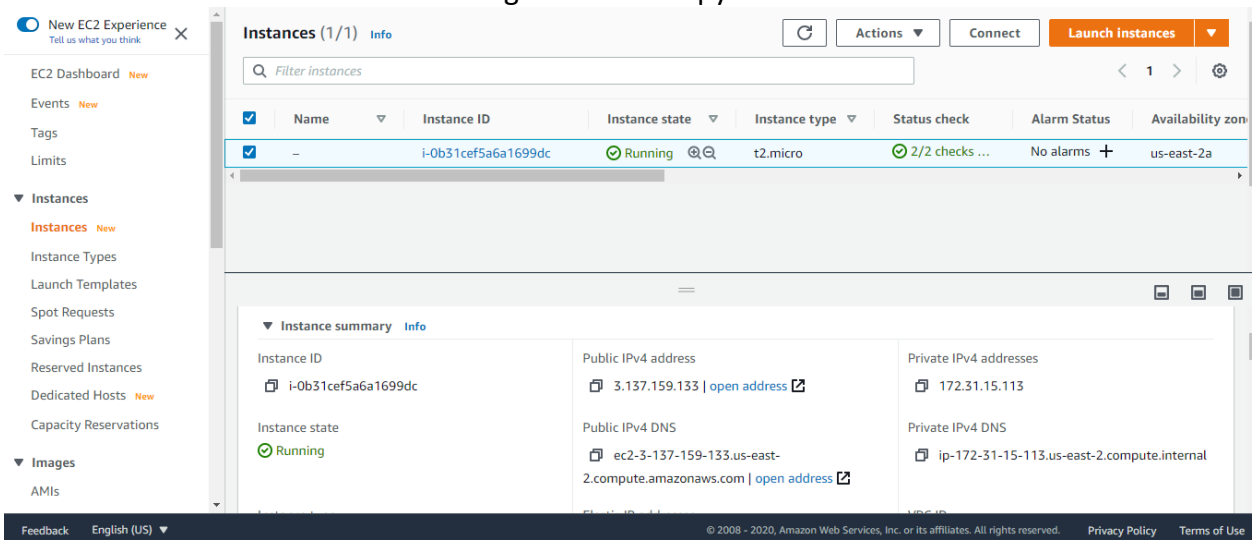
Step 2: Convert public key to private key

Open PuTTY Key Generator and click “Load” to locate the key pair file you download. And then click “Save private key”.

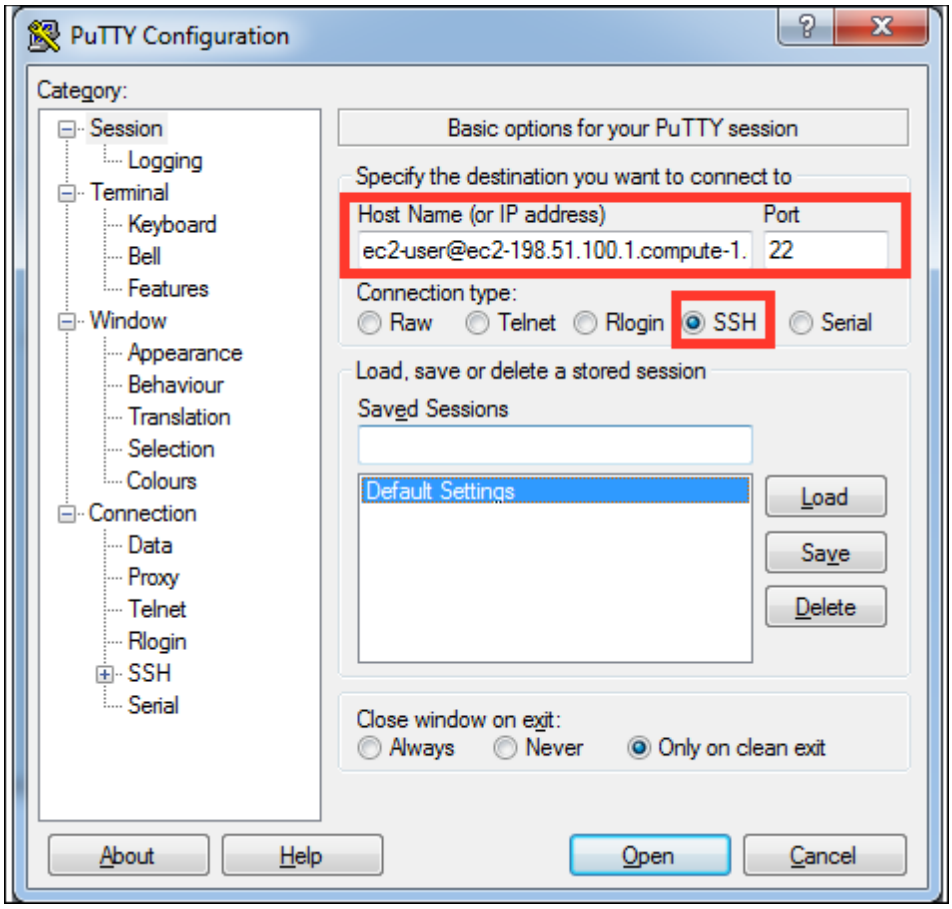


Step 3: Connect to EC2 from local

Go to EC2 dashboard and check running instances. Copy “Public IPv4 DNS”.

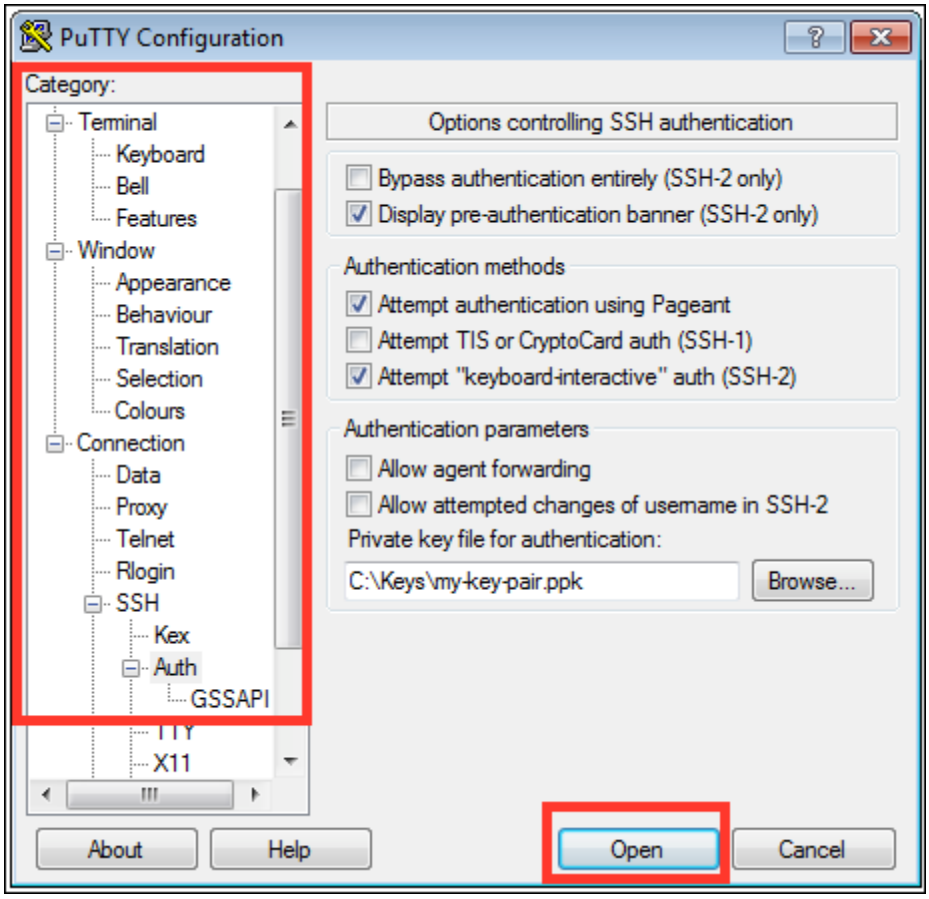


Open PuTTY. At blank of host name, type “ec2-user@” and paste the address you just get.

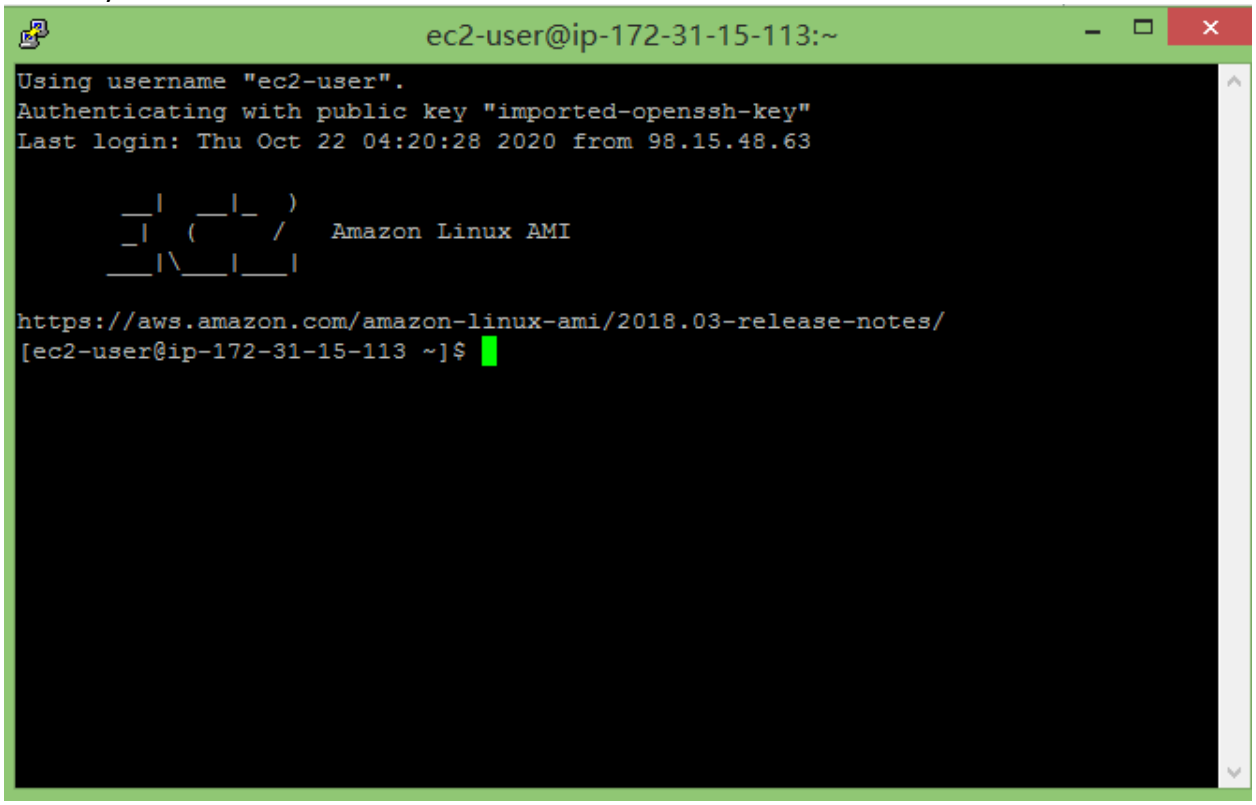


In the **Category** pane, choose **Connection**, **SSH** and **Auth**. Complete the following:

- Choose **Browse**, select the .ppk file that you generated for your key pair, and then choose **Open**.
- Choose **Open** to start the PuTTY session.



So now you can connect to EC2 instance from local.



Step 4 AWS configuration

Run this command to quickly set and view your credential, region and output format. The following example show sample values.

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPL
Default region name [None]: us-west-2
Default output format [None]: json
```

Replace AWS access key id and AWS secret access key to your own ones.

Step 5 Install Python 3, pip 3, and the EB CLI on instance

In Linux command line, type as shown in the figure.

```
[ec2-user@ip-172-31-15-113 ~]$ sudo yum install python36
```

Type “y” and click enter.

```
Transaction Summary
================================================
Install  1 Package (+3 Dependent packages)

Total download size: 15 M
Installed size: 48 M
Is this ok [y/d/N]:
```

Install pip for python 3. Download the installation script.

```
[ec2-user@ip-172-31-15-113 ~]$ curl -O https://bootstrap.pypa.io/get-pip.py
```

Run the script with Python.

```
[ec2-user@ip-172-31-15-113 ~]$ python3 get-pip.py --user
```

Use pip to install the EV CLI

```
[ec2-user@ip-172-31-15-113 ~]$ pip3 install awsebcli --upgrade --user
```

Step 6 Install Python packages for the project

```
[ec2-user@ip-172-31-15-113 ~]$ pip3 install tweepy
[ec2-user@ip-172-31-15-113 ~]$ pip3 install pandas
```

Step 7 Load data and python program into EC2 instance

Use the following command to copy an object from Amazon S3 to your instance.

[ec2-user ~]$ aws s3 cp s3://my_bucket/my_folder/my_file.ext my_copied_file.ext

Eg:

```
[ec2-user@ip-172-31-15-113 ~]$ aws s3 cp s3://ircovid19project/twitter-dataCollection/hate/hate_5000-1.csv hate_5000-1.csv
```

Use the same method to load python program “Tweets_scraper.py” into EC2 instance.

Make sure both data and python program are in instance already using the command below.

```
[ec2-user@ip-172-31-15-113 ~]$ ls
get-pip.py  hate_5000-1.csv  Tweets_scraper.py
```

Step 8 change data file name and output file name in the python code and add your Twitter API token into the code.

```
[ec2-user@ip-172-31-15-113 ~]$ nano Tweets_scraper.py
```

Change the load file name to the one you copy to EC2 instance. Add your consumer key, consumer secret, access token and access token secret into the code.

This link is for how to get API Keys and Tokens for Twitter:

https://www.slickremix.com/docs/how-to-get-api-keys-and-tokens-for-twitter/

```
  GNU nano 2.5.3           File: Tweets_scraper.py                  Modified

import pandas as pd
import tweepy
import datetime
import time

# load data
users = pd.read_csv("hate_5000-1.csv")
user_ids = users['User ID']


# tweet authorization
consumerKey ="YOUR CONSUMER KEY"
consumerSecret ="YOUR CONSUMER SECRET"
accessToken ="YOUR ACCESS TOKEN"
accessTokenSecret ="YOUR ACCESS TOKEN SECRET"

auth = tweepy.OAuthHandler(consumerKey, consumerSecret)
auth.set_access_token(accessToken, accessTokenSecret)


^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Linter ^_ Go To Line
```

Change the output file name accordingly. If '**hate_5000-1.csv**" is your input file, your output file name should be "**hate_5000-1-result.csv**". Use PgDn to move to the bottom of the code.

```
  GNU nano 2.5.3          File: Tweets_scraper.py                    Modified

            #print(user_data_frame.shape)
            print("=======DONE=======")
            #time.sleep(120)
            user_count += 1
            totalTime += toc-tic
        except:
            pass
toc2 = time.perf_counter()
print(f"Total consumed tweet collection time: {totalTime:0.4f} seconds.")
print(f"Total consumed running time: {toc2 - tic1: 0.4f} seconds.")
print("The number of valid users: "+str(user_count))
data_dataframe = pd.DataFrame(data=data)
data_dataframe.to_csv("hate_5000-1-result.csv")

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Linter ^_ Go To Line
```

After these, press Ctrl+X, y and Enter in order.

```
Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
 Y Yes
 N No           ^C Cancel
```

Step 9 Run the python program

```
[ec2-user@ip-172-31-15-113 ~]$ python3 Tweets_scraper.py
```

Step 10 Save your output file into S3 storage.

Use the command to save file into S3 storage. Please put into running_results folder.

[ec2-user@~] $ aws s3 cp hate_5000-1-result.csv s3://ircovid19project/ twitter-dataCollection/running_results/hate_5000-1-result.csv