Fundamental Understanding of Amazon EC2 Report

Genese Solution Pvt. Ltd.

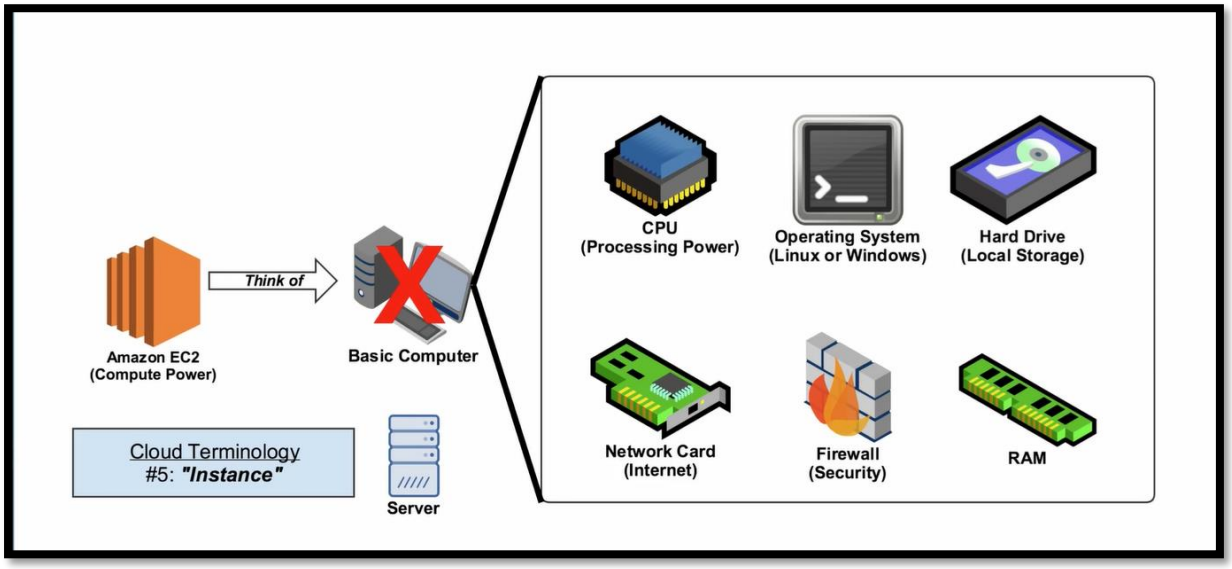
Table of Contents

Fundamental Understanding of Amazon EC2

Overview

In general, Amazon EC2 is a virtual computer that can be used for whatever you like. Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. We can develop and deploy applications faster. Amazon Elastic Compute Cloud (Amazon EC2) is a web server that provides secure, resizable compute capacity in the cloud. It is intended to make web-scale cloud computing easier for developers. It's simple web service interface allows us to obtain and configure capacity with minimal friction. EC2 provides complete control of our computing resources and Amazon computing environment.

Amazon EC2 Comparison with Basic Computer



Source: https://www.youtube.com/watch?v=izyqJPl9wW8

Actually, Amazon EC2 offers the wide and deepest compute platform with options of processor, storage, networking, operating system, and purchase model. It offers the fastest processors in the cloud with **400 Gbps ethernet networking**. Amazon EC2 has the most powerful GPU instances for machine learning training and graphics workloads and also the lowest cost-per-inference instances in the cloud. More SAP, HPC, Machine Learning, and Windows workloads run on AWS than any other cloud.

Amazon EC2 users:

Netflix, Volkswagen, Airbnb, Expedia, Finra, Cathay Pacific, Allergan, Celgene.

Building Blocks

- Amazon EC2 has **faster innovation and increased security** with AWS Nitro System.
- Amazon EC2 provides **choice of processors** such as latest version Intel Xeon, AMD EPYC, and AWS Graviton CPUs to find the best balance of performance and price for your workloads.

- Amazon EC2 has **high performance storage** called Amazon Elastic Block Store (EBS) that allow us to optimize storage performance and cost for our workloads.
- Amazon EC2 offers first and only 400 Gbps enhanced **Ethernet networking** for compute instances. It enables us to get dramatically higher packet per second (PPS), lower network jitter, and lower latency. Elastic Fabric Adapter is avaialable for high performance computing (HPC) applications is a network interface.
- Amaxon EC2 has choices of **multiple purchasing models** with On-Demand, Spot Instances, and Savings Plan. For instance, we can save up to 90% for fault-tolerant workloads with Spot Instances,

Advantages of EC2

- EC2 provides secure and resizable compute capacity in cloud and instances scaled up and down.
- It is integrated with other services such as S3 and RDS.
- We can stop using it after completion of workload.
- We can provision and launch an Amazon EC2 instance within minutes.
- We have to **Pay for what we use.**
- Instances can be lunched in one more regions and zones and it also supports different OS. 25 regions and 81 availability zones are globally available until now.
- It works with **Amazon Virtual Private Cloud** to provide secure network to resources.

Amazon EC2 working mechanism

There are three major parts of Amazon EC2.

- **Launch: Firstly**, Select the instance type with basic configuration which include os, applications servers or applications. We fix security setting to control the network traffic that can flow into and out of your instance.
- **Connect instance: Secondly**, connect to our instance by logging and accessing the computer desktop.
- **Application:** After that, we connect our instance, then we can start using it. We can run commands to install softwares, add storage, copy and organize files.

Amazon EC2 Instance types

Amazon EC2 instance can be categorized into five groups.

General purpose instances

- provide a balance of compute, memory, and networking resources.
- use for application servers, gaming servers, backend servers for enterprise applications, small and medium databases. t-type and m-type

Compute Optimized instances

- Ideal for compute bound applications that benefit from high performance processors.
- Instances belonging to this family are well suited for batch processing workloads, media transcoding and high-performance web servers,

- Uses for high performance computing (HPC), scientific modeling, dedicated gaming servers and ad server engines, machine learning inference and other compute intensive applications.
- c-type

Memory optimized instances

- Deliver fast performance for workloads that process large datasets in memory.
- It holds all the data and instructions that a central processing unit (CPU) needs to be able to complete actions.
- For e.g r-type, x-type and z-type

Accelerated computing instances

- Use hardware accelerators, or coprocessors, to perform some functions more efficiently than is possible in software running on CPUs.
- Examples of these functions include floating-point number calculations, graphics processing, and data pattern matching.
- For e.g- tf-type, g-type and p-type

Storage optimized instances

- Designed for workloads that require high, sequential read and write access to large datasets on local storage.
- Storage optimized instances include distributed file systems, data warehousing applications, and high-frequency online transaction processing (OLTP) systems.
- For e.g d-type, h-type and i-type

Key Terminology

- Virtual computing environments called **instances.**
- Preconfigured templates for your instances, known as **Amazon Machine Images (AMIs)**, that package the bits you need for your server for both the operating system and additional software.
- Various configurations of CPU, memory, storage, and networking capacity for your instances called **instance types.**
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as **Regions and Availability Zones**
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using **security groups**
- Static IPv4 addresses for dynamic cloud computing, known as **Elastic IP addresses**
- Metadata called as tags which create and assign to Amazon EC2 resources.
- Virtual networks you can create that are logically isolated from the rest of the AWS Cloud, and that can optionally connect to our own network, known as **virtual private clouds (VPCs)**
- Secure login information for your instances using key pairs (AWS stores the public key, and you store the private key in a secure place)
- Storage volumes for temporary data that's deleted when we stop, hibernate, or terminate instance, known as **instance store volumes.**

- Persistent storage volumes for your data using Amazon Elastic Block Store known as **Amazon EBS volumes.**
- **AWS Nitro System** is a combination of dedicated hardware and lightweight hypervisor which enables faster innovation and enhanced security.

Best practices for EC2

Security

- Responsible for access to AWS resources and APIs using identity federation, IAM users, and IAM roles.
- Establish credential management policies and procedures for creating, distributing, rotating, and revoking AWS access credentials.

Storage

- Understand the implications of the root device type for data persistence, backup, and recovery.
- Use separate Amazon EBS volumes for the operating system versus your data. Ensure that the volume with your data persists after instance termination.
- Use the instance store available for your instance to store temporary data.

Resource management

- Use instance metadata and custom resource tags to track and identify your AWS resources.

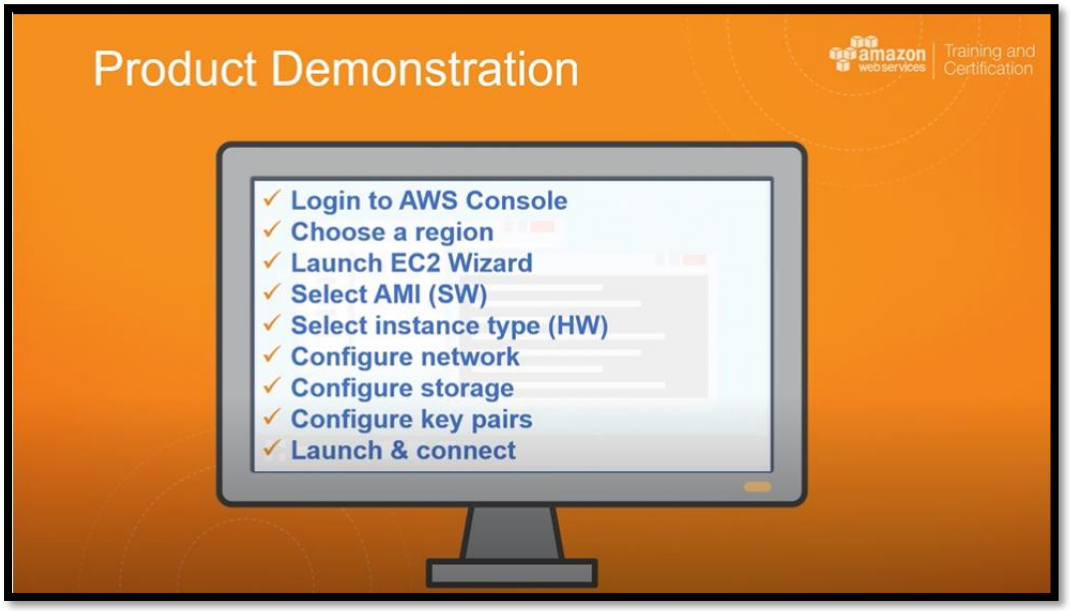
Backup and recovery

- Regularly back up your EBS volumes using Amazon EBS snapshots, and create an Amazon Machine Image (AMI) from your instance to save the configuration as a template for launching future instances.
- Deploy critical components of your application across multiple Availability Zones, and replicate your data appropriately.

Networking

- Set the time-to-live (TTL) value for your applications to 255, for IPv4 and IPv6.

Amazon EC2 Product Demonstration outlines



Source: https://www.aws.training/Details/Video?id=16382

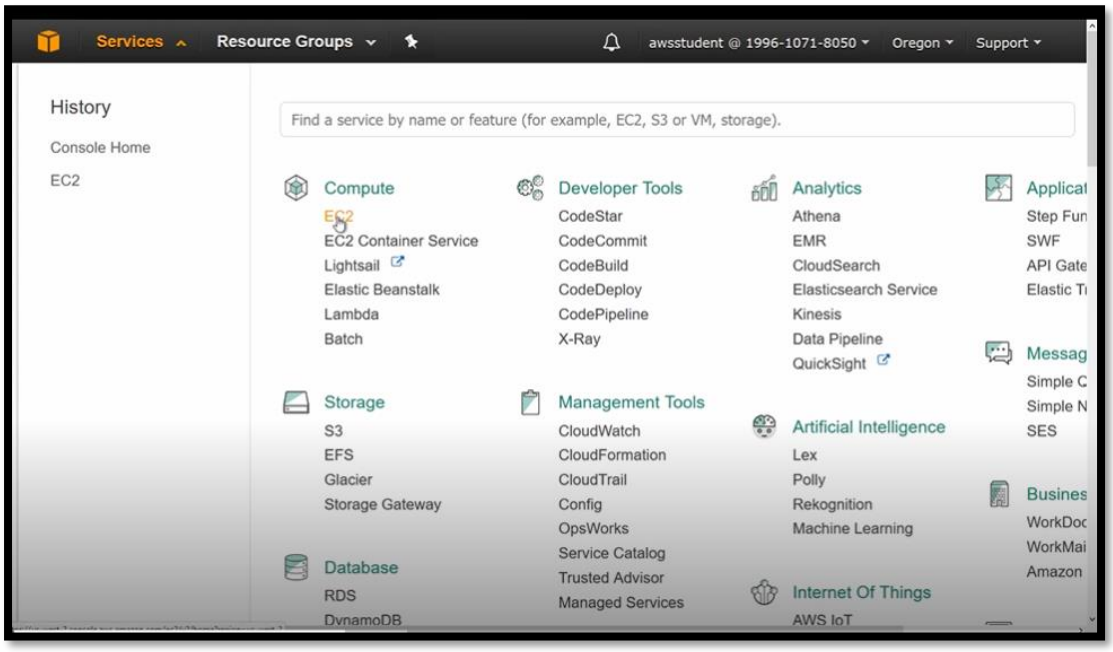
Amazon EC2 Demo

Firstly Amazon EC2 completes in two stages: Amazon instance setup and its connection with Windows & PuTTY.

I. Amazon EC instance creation

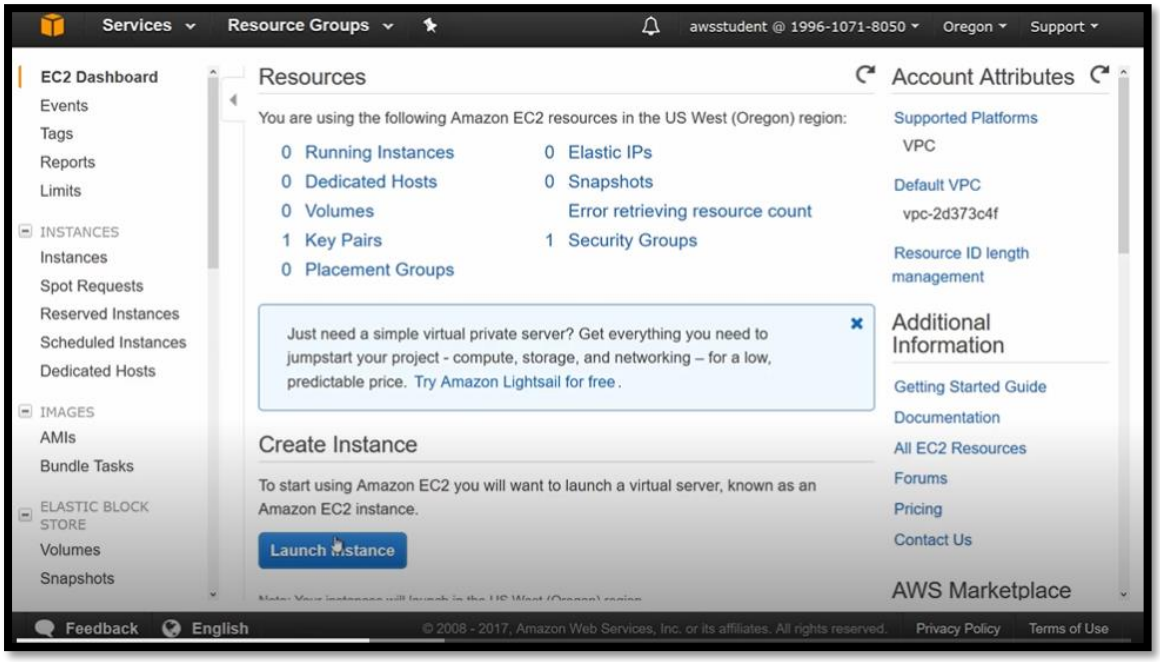
First, we should **have AWS account** and log into **Amazon EC2 console** at https://console.aws.amazon.com/ec2/ as mentioned below fig 1

Fig 1: Amazon service for EC2



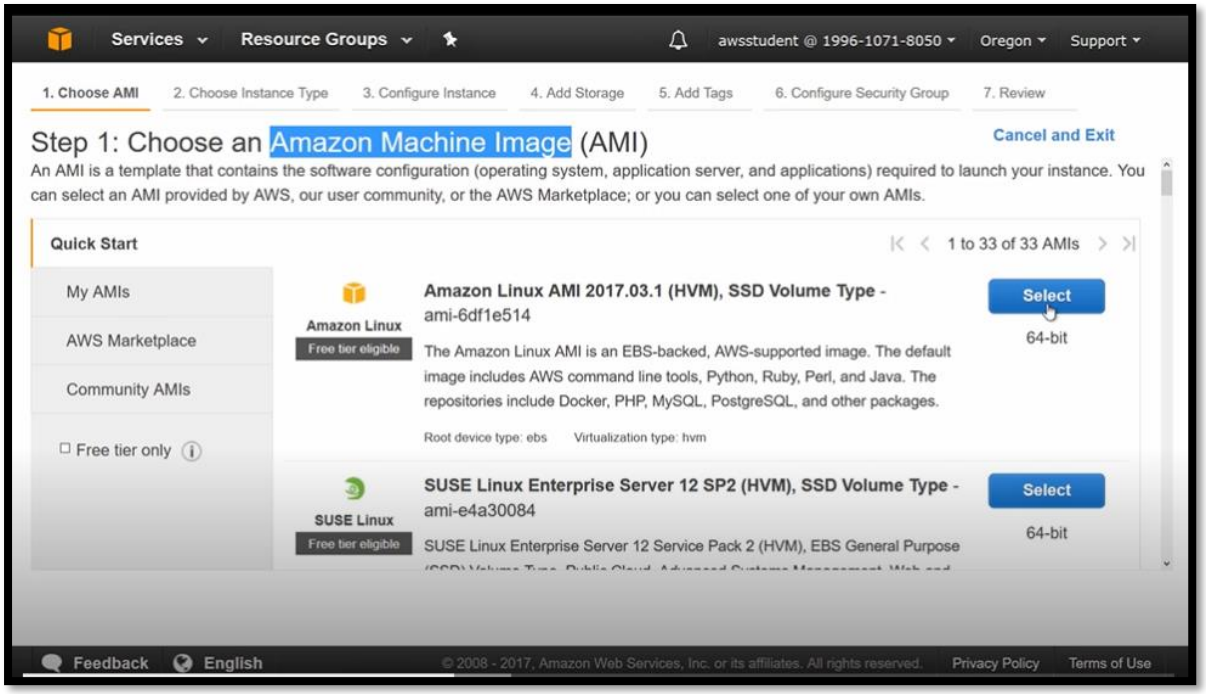
Secondly, From the console dashboard, choose **Launch Instance**(fig2)

Fig 2: Console dashboard for launch instance



Then, The Choose an Amazon Machine Image (AMI) page appears with a list of basic configurations, called Amazon Machine Images (AMIs), that serve as templates for your instance. And after that **Select an HVM version of Amazon Linux 2**. We will find AMIs are marked "Free tier eligible." See fig 3.

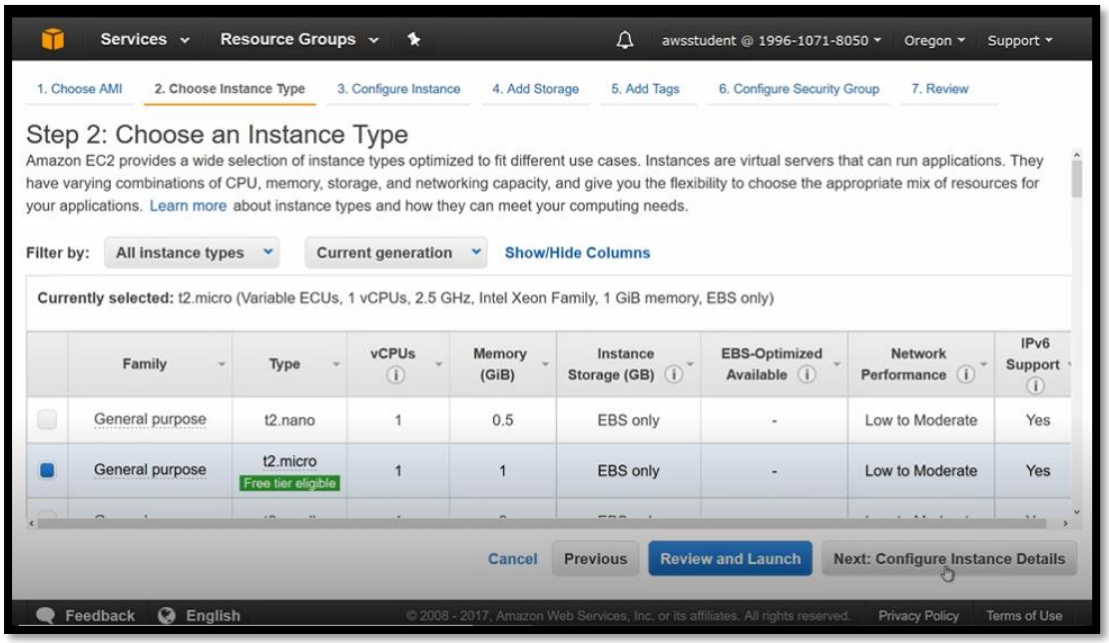
Fig 3: Amazon Machine Image(AMI)



After that, the Choose an Instance Type page disclose, we can select the hardware configuration of our own instance. Please **select the t2. micro instance type**, which is selected by default. Further, the t2. micro instance type is eligible for the free tier. In addition, we can use a t3.

micro instance under the free tier when where t2. micro is unavailable in your regions. Finally, we can **click next configure instance** detail further after completion (fig 4)

Fig 4: Instance Type page



On the Choose an Instance Type page, click **Review and Launch** to let the wizard complete the other configuration settings for you. Fix number of instances. Stay with default network, Subnet and auto sign Public IP(fig5)

Fig 5: Configure instance details

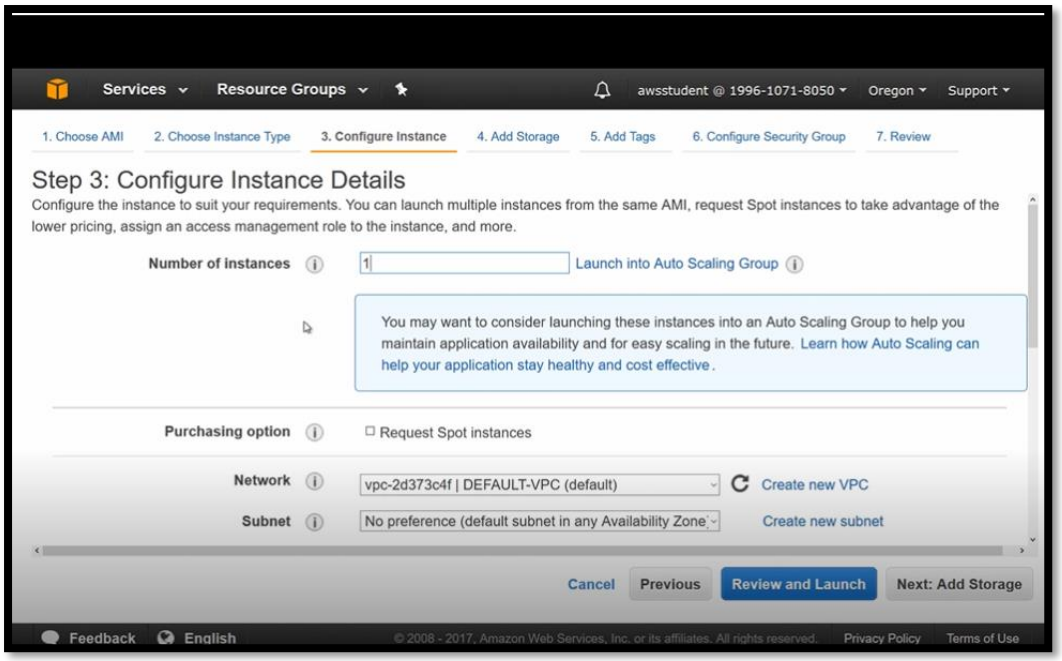
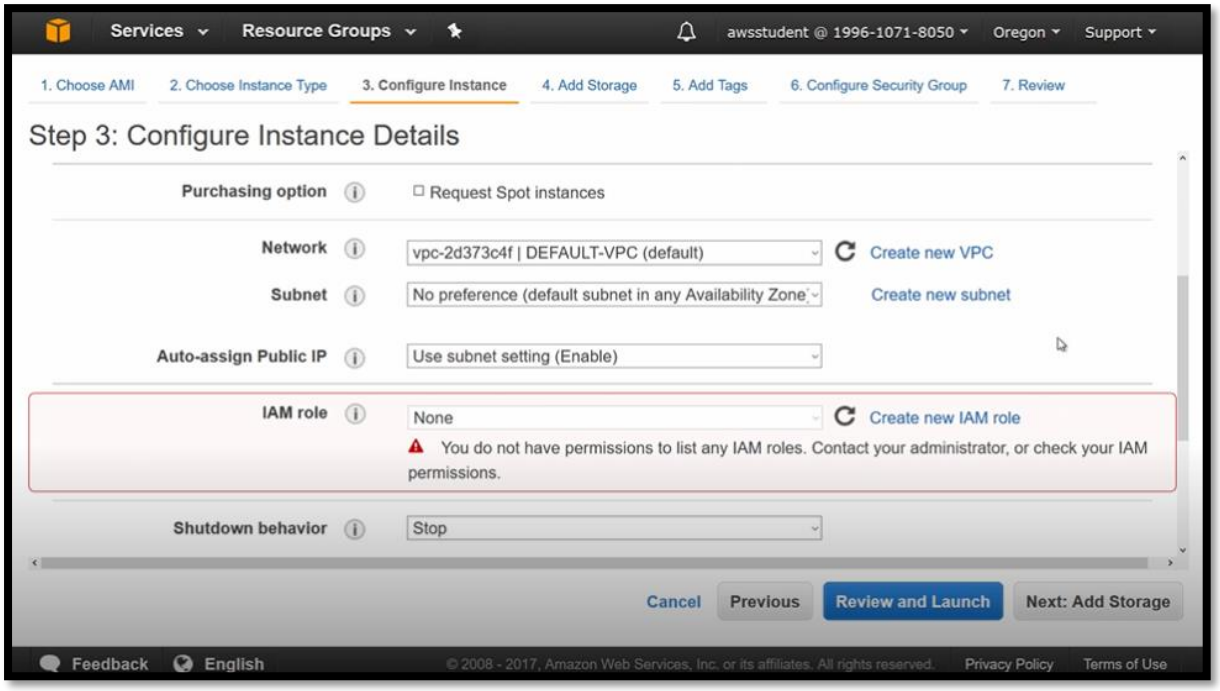
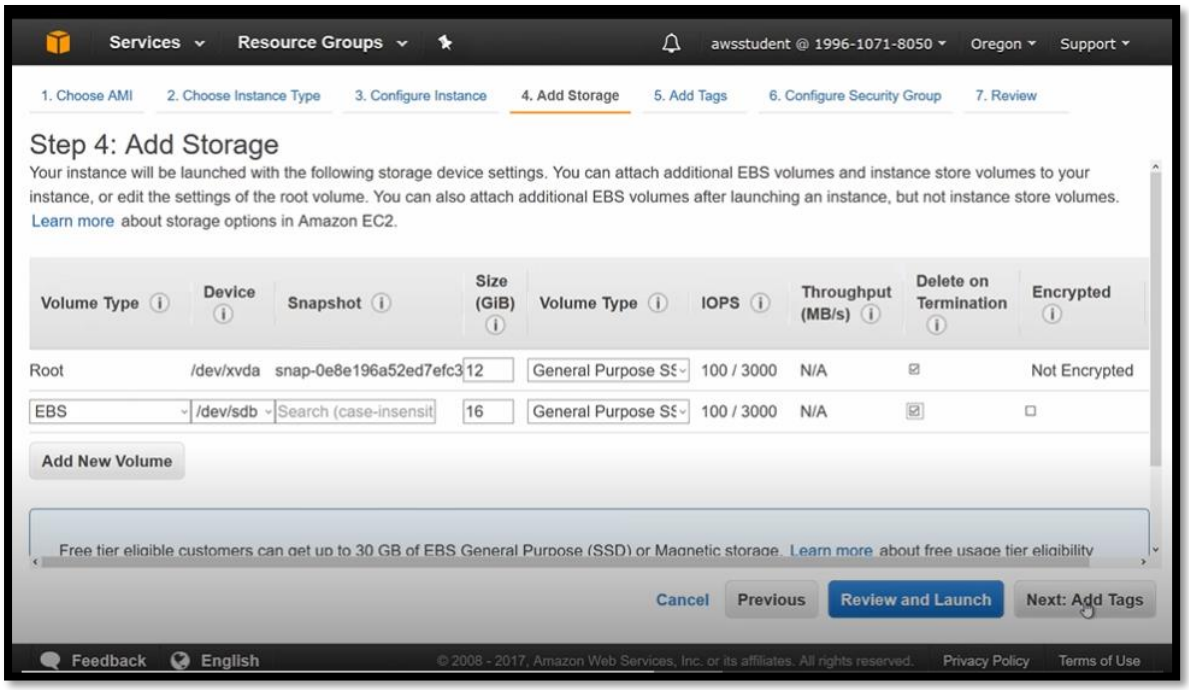


Fig: 6: Configure instance details



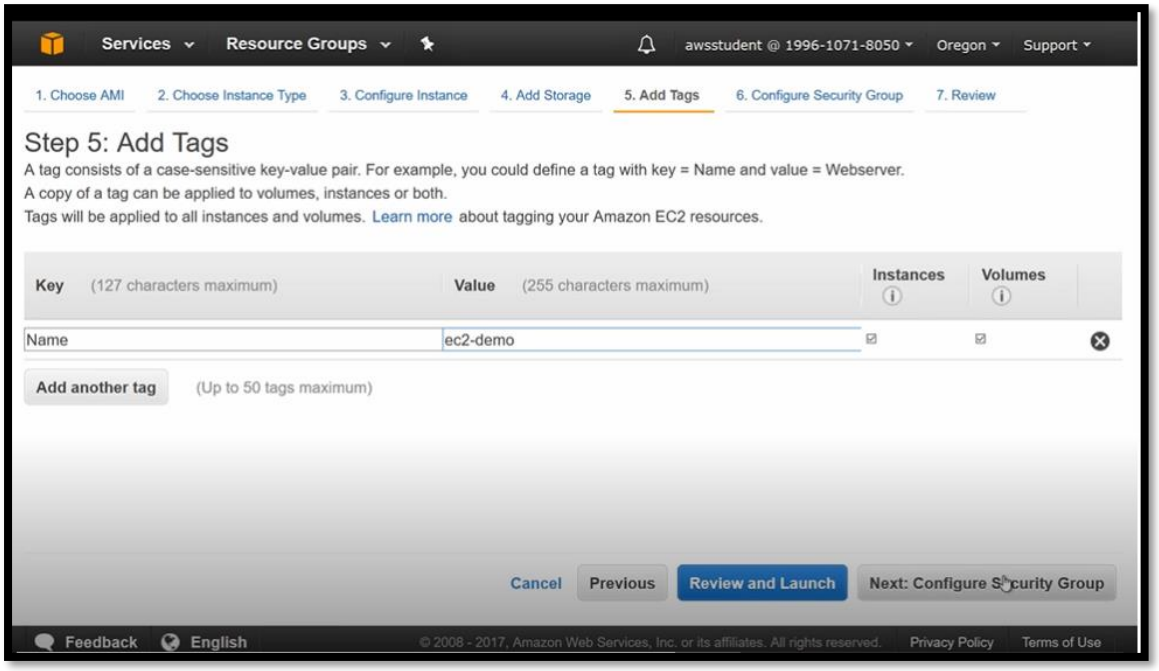
After Configure Instance completion, click **Add Storage**. On add storage we can increase root volume from 8 to 12 and also, we can change volume type such as General Purpose SSD(GP2), magnetic etc. and we can add new volume. Further we can delete instance as, mentioned by selecting delete on termination. Lastly, please click **Add Tags** as mentioned below(fig 7)

Fig 7: Add Storage



Here, we can **add tag such as key and value** on right side of instance. We named ec2-demo. Please click on **Next configure Security Group** as shown in figure 8.

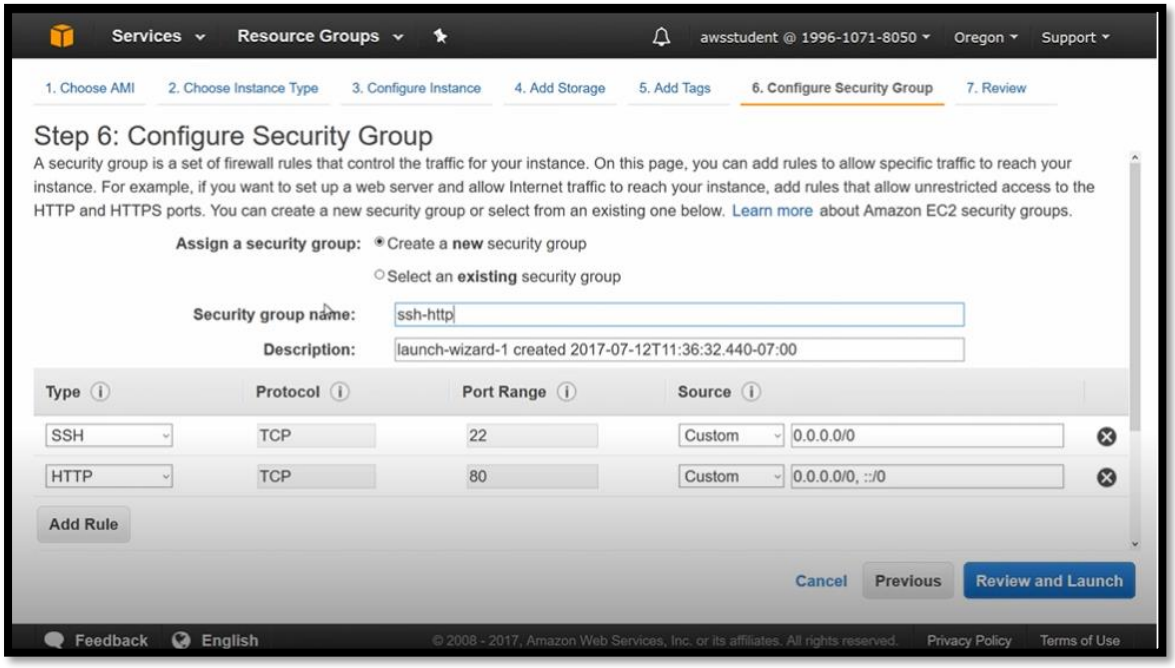
Fig 8:Add Tags on Instance



Under **Configure Security Groups** on review instance launch page, we will see the wizard created and selected a security group for us. We can **use either exiting security group, or we can select the security group** that we created when getting set up by choose Edit security groups. Please ensure that select an existing security group is selected.

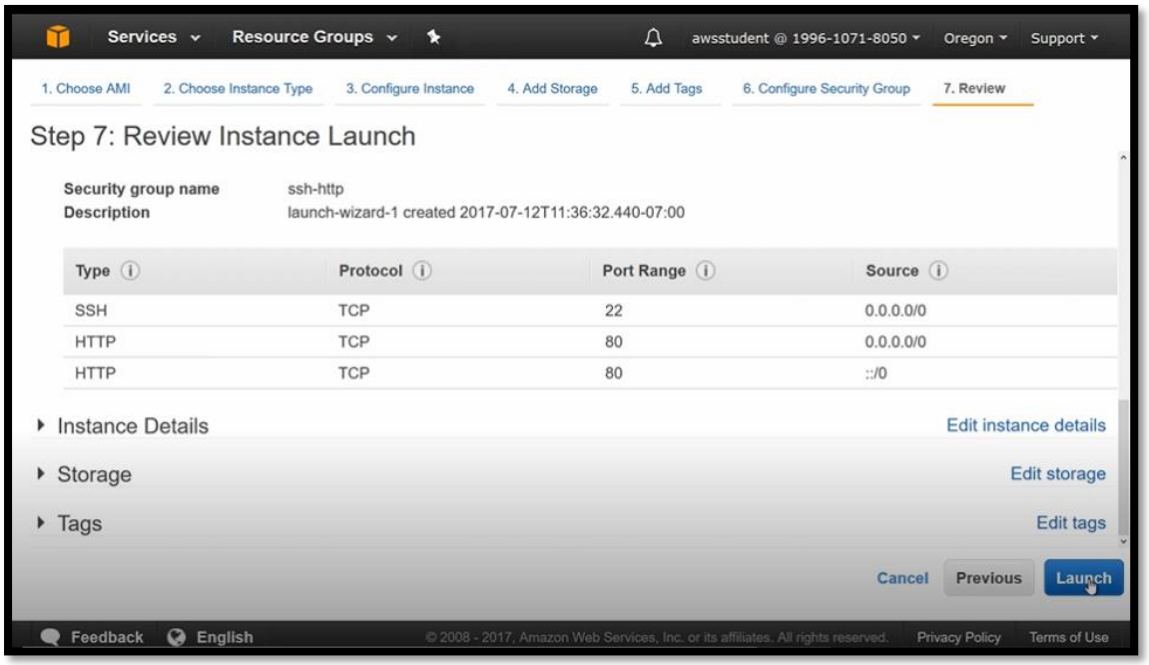
Further we select our security group from the list of existing security groups, and then we choose **Review and launch** at the corner of Instance Page launch. See fig 9.

Fig 9: Configure Security Group



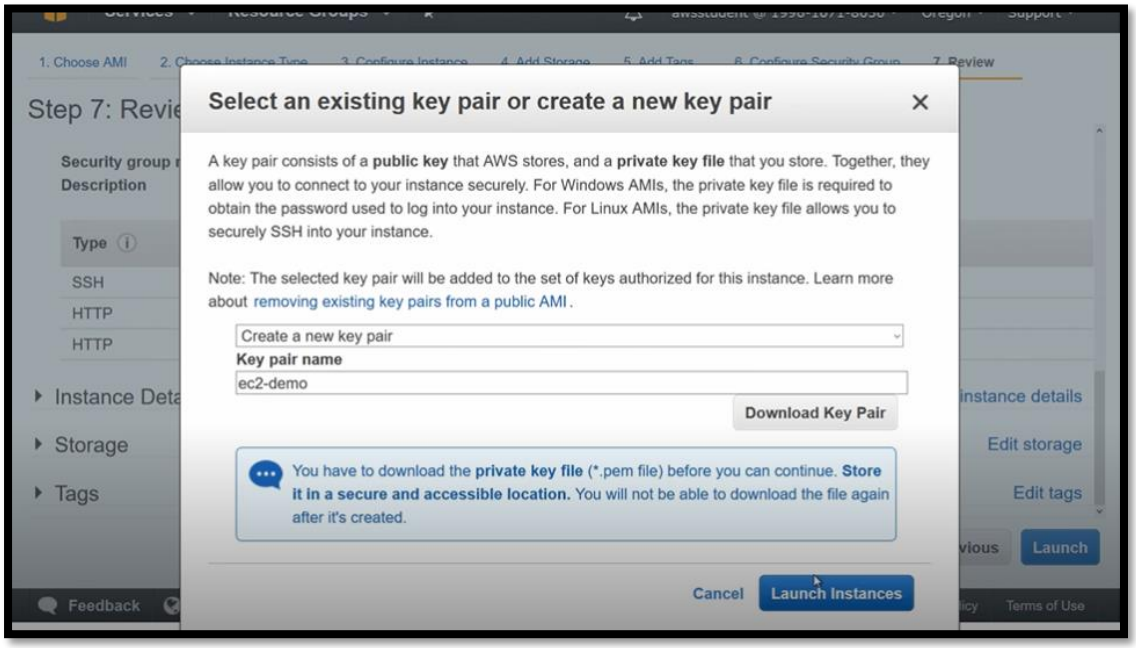
Finally, **Review Instance Launch Page** opens which remind us about our selection and then choose Launch as mentioned below Fig 10.

Fig 10: Review Instance launch



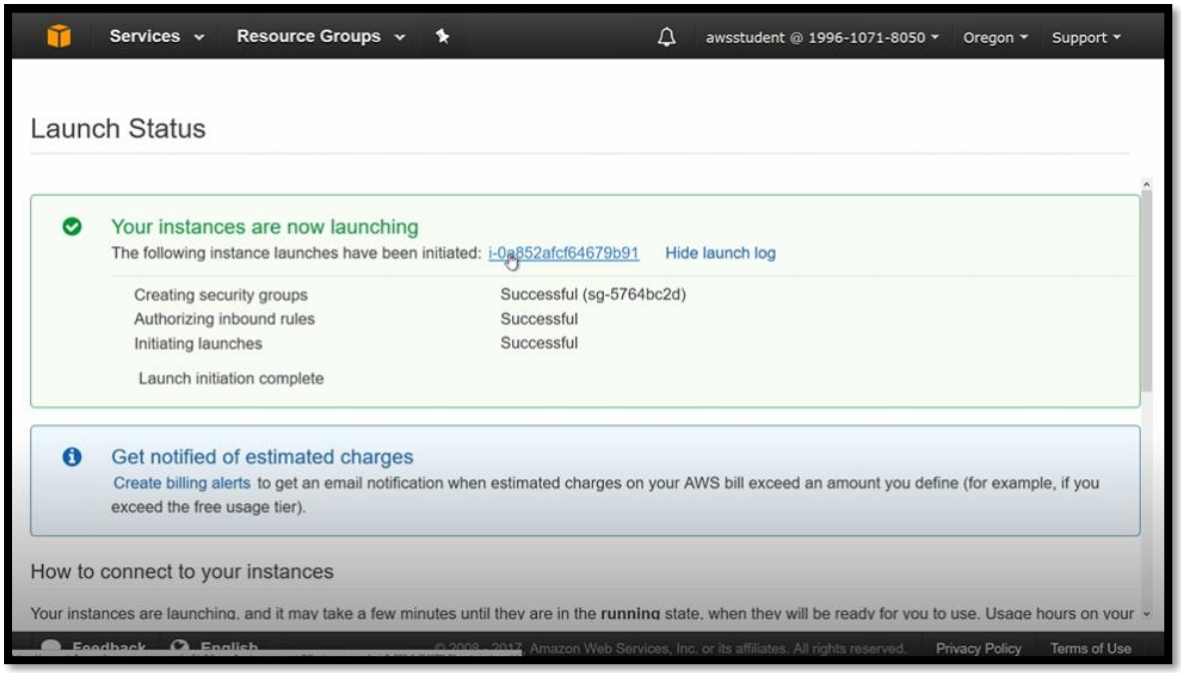
When prompted for a key pair, select **choose an existing key pair**, then select the key pair that you created when getting set up. Please remember that don't select proceed without a key pair. If we launch our instance without a key pair, then we fail to connect to it. Here we wil generate a new key pair. and then choose **Launch Instances aa the buttom of left instance**. Select the acknowledgement check box, if appear.(fig 11)

Fig 11: Create or exit key pair page



A confirmation page lets you know that your instance is launching. As shown in fig 12, view Instances to close the confirmation page and return to the console.

Fig 12 Launch status Instance



We will see our instance pending and please **click refresh bottom** above the pending as shown in figure 13 and 14.

Fig 13 Pending instance

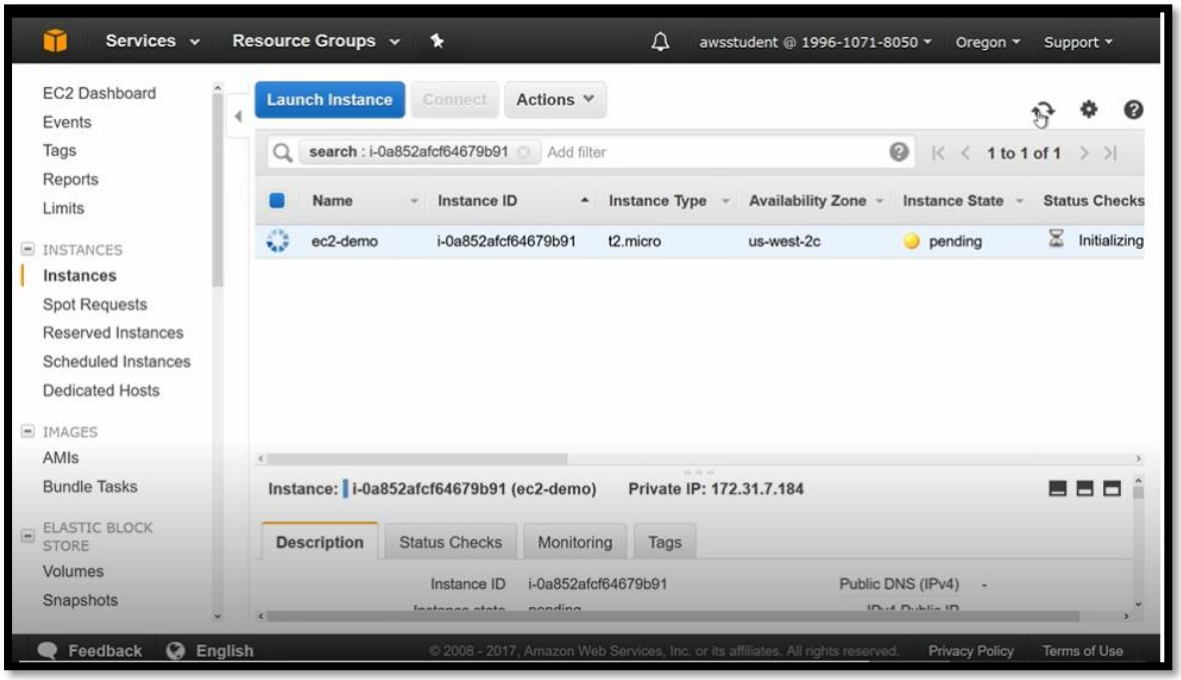
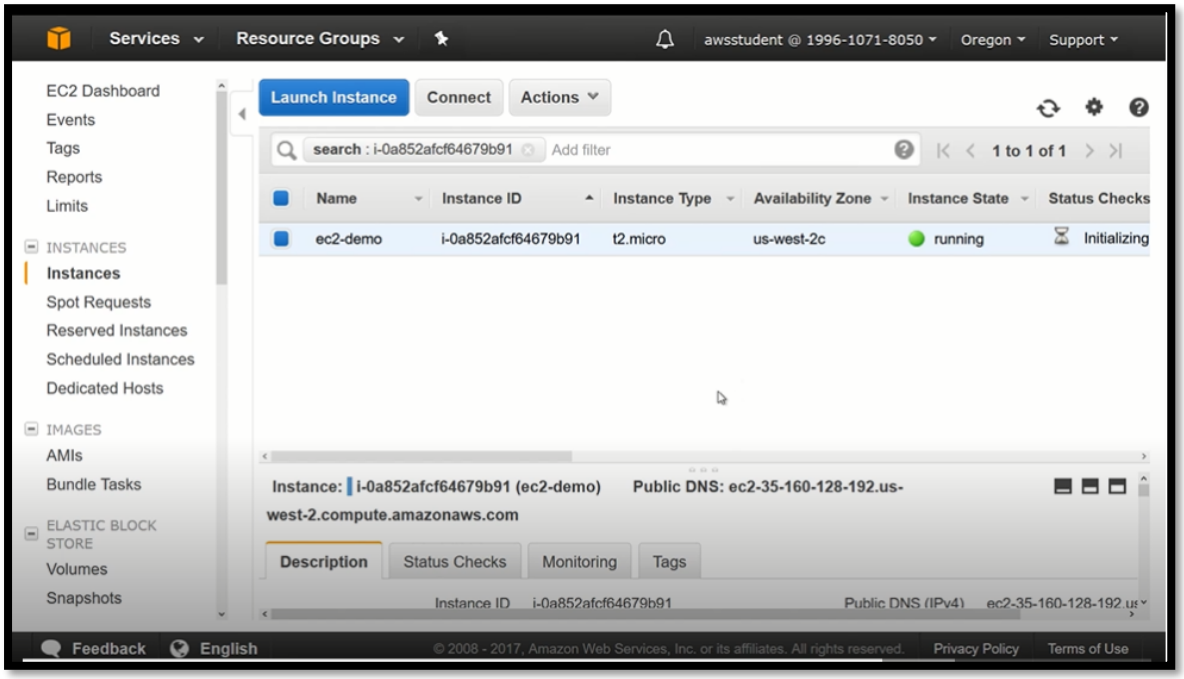
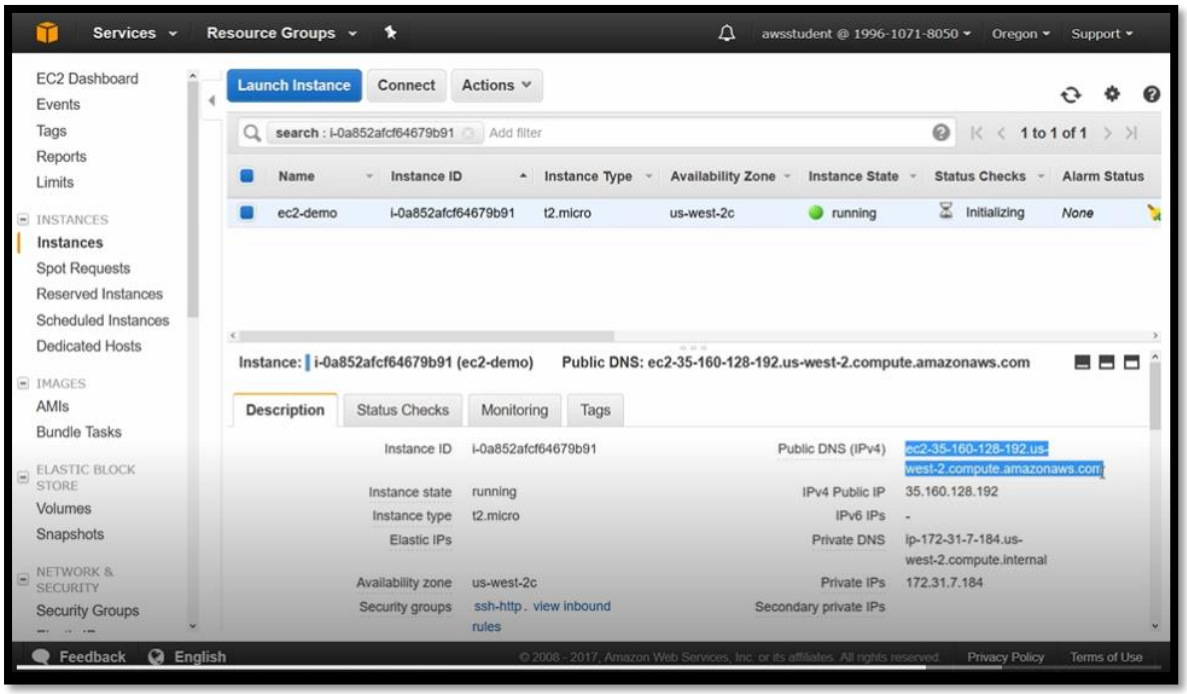


Fig 14: Running instance



After that **highlight instance** and copy **Public DNS** shown in figure 15.

Fig 15: Running instance



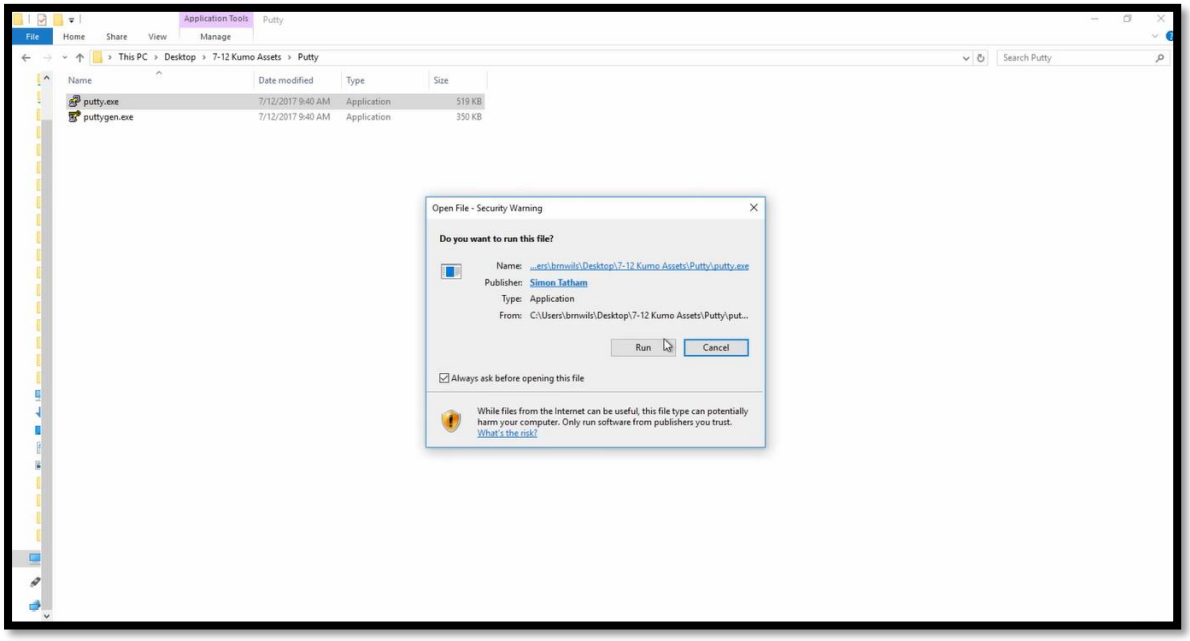
II. Connect to an AWS EC2 Instance – Windows & PuTTY

After creation of AWS Linux EC2 instance, we can connect to it from our computer using the SSH protocol. **PuTTY** is a free SSH client that allows us to do this from a local computer running Windows. Then, we work within the EC2 instance as our linux.

Initially, **Please download Putty files** from Putty.org website.

Then. Lunch putty.exe on Pc an open it.(fig16)

Fig 16: PUTTY launch



Then PuTTy configuration dialog box will appear. Type your **default user and paste** in DNS and click on Open buttom.(fig17)

Fig 17: PUTTY configuration dialog box

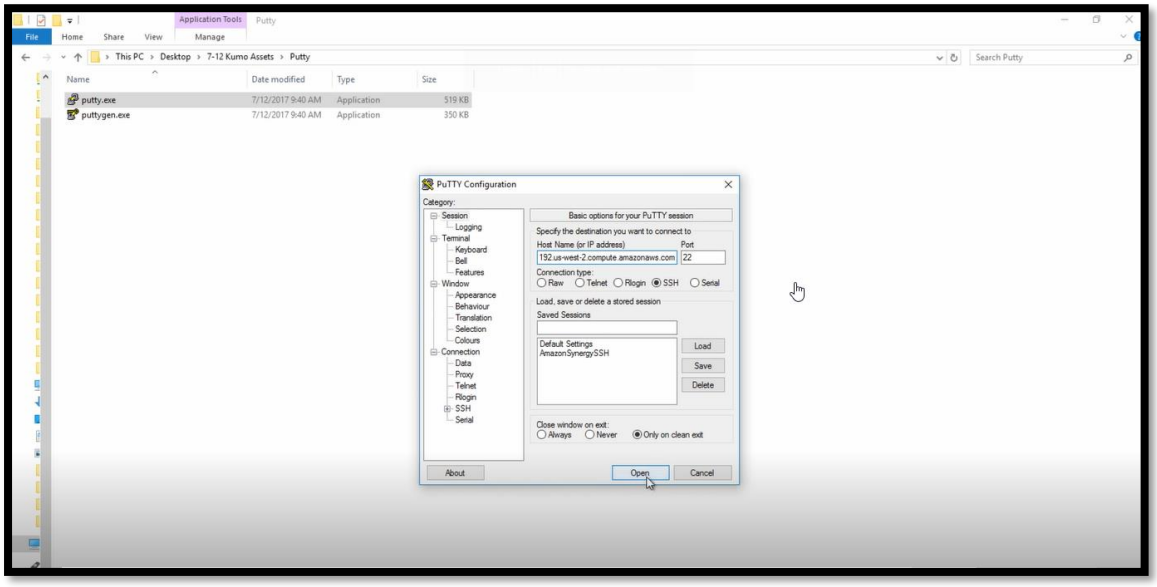
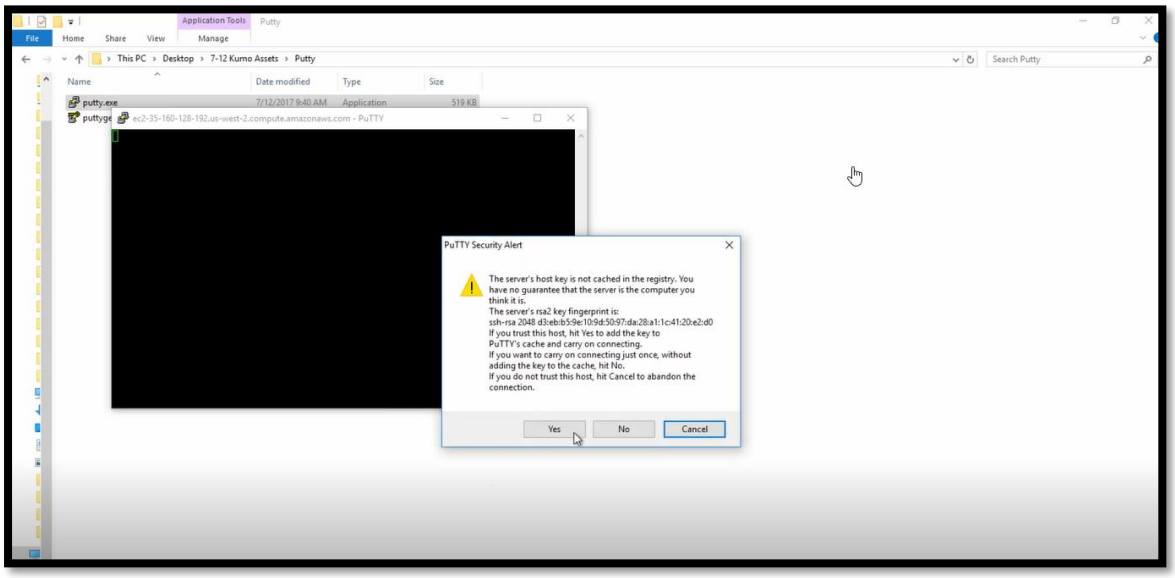
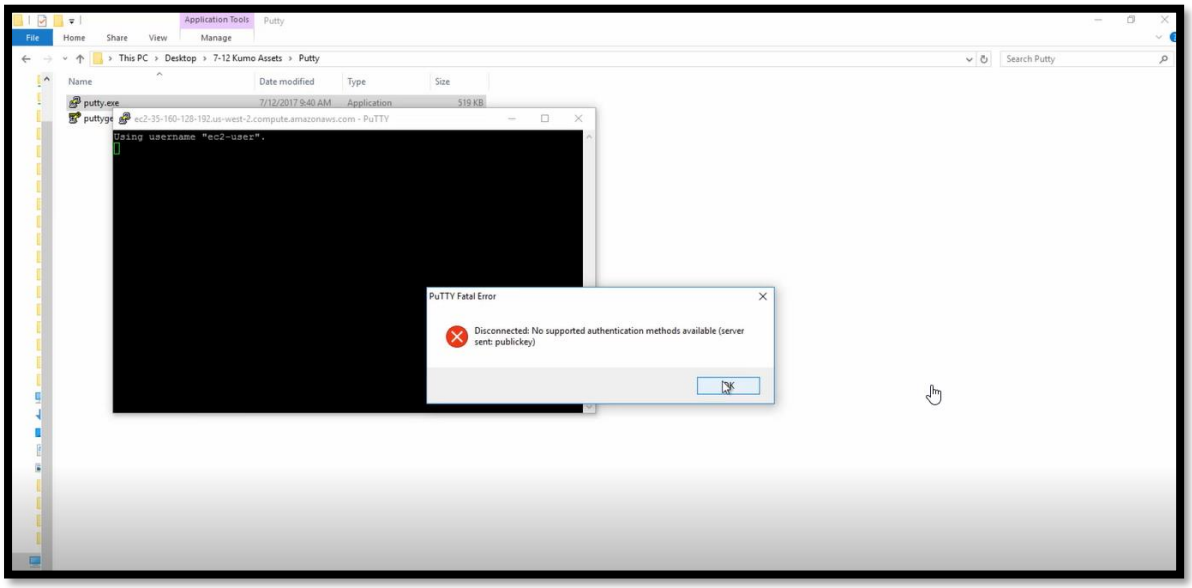


Fig 18: PuTTY security Alert



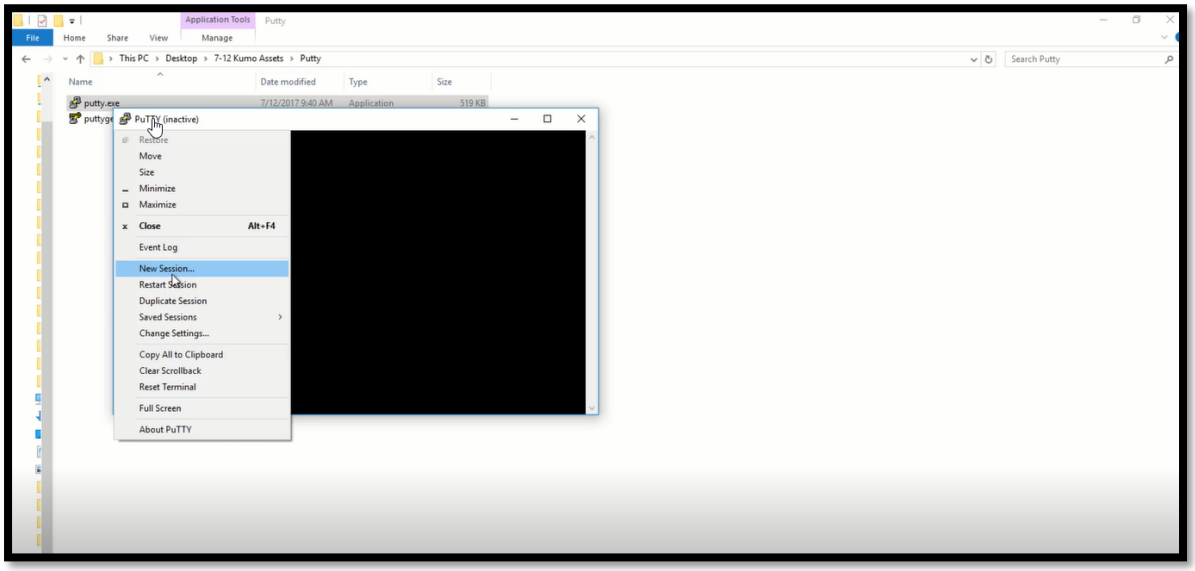
Here it does not work. Private key has not been configured.(fig19)

Fig 19: PuTTY Fatal Error.



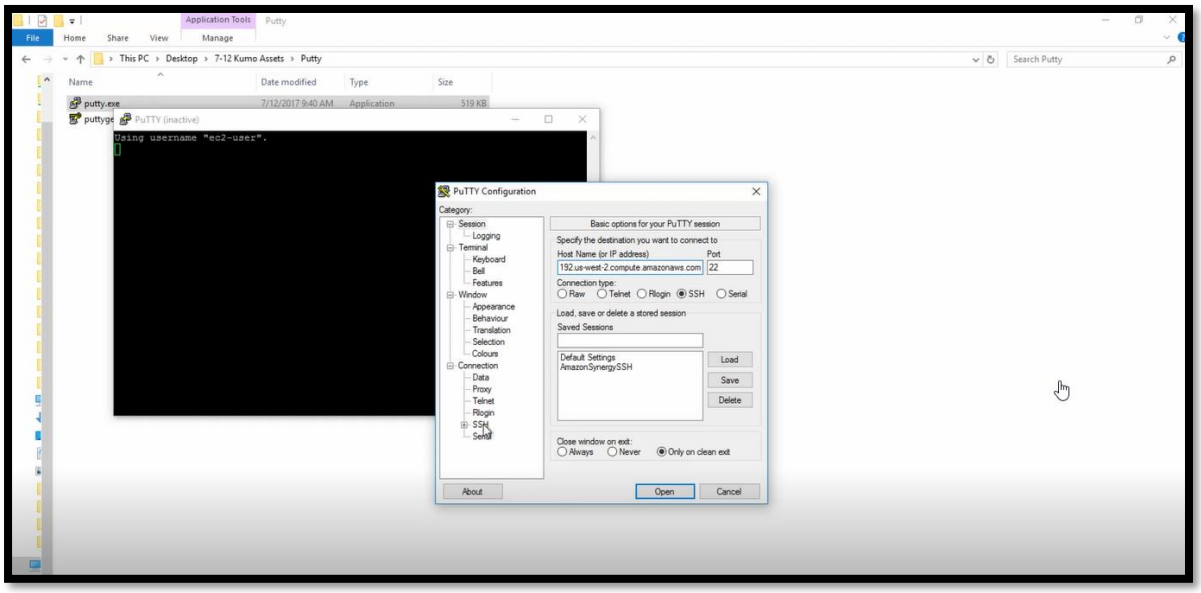
Go **PuTTy-New session(fig20)**

Fig 20: PuTTy New session



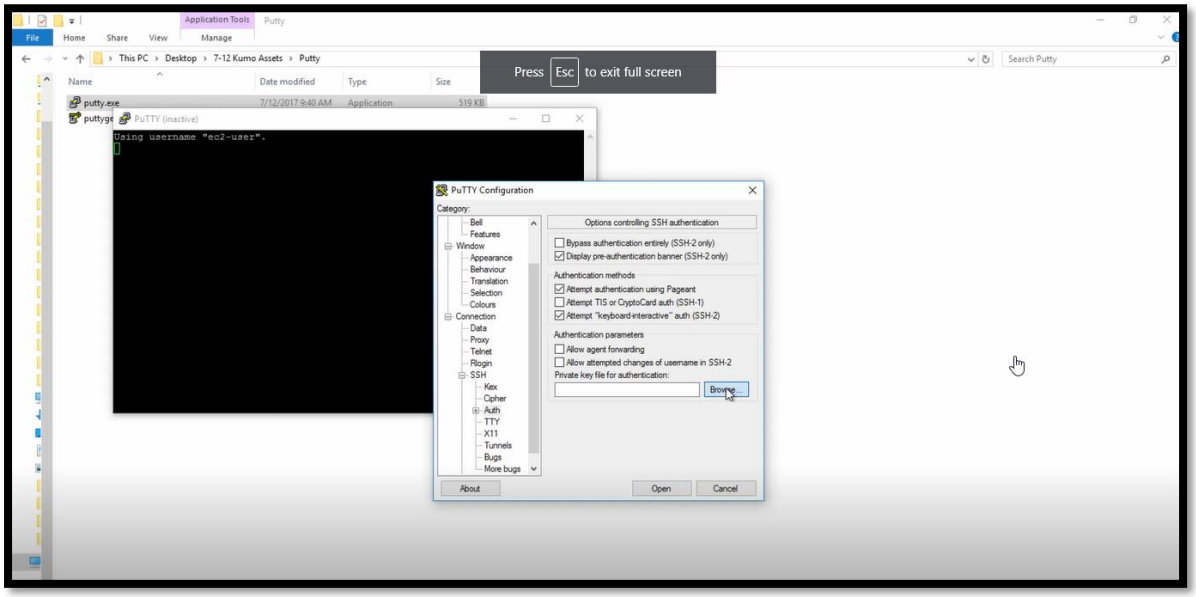
Type default instance and **click SSH** and under SSH we **select Auth.(fig21)**

Fig 21: PuTTY Configuration



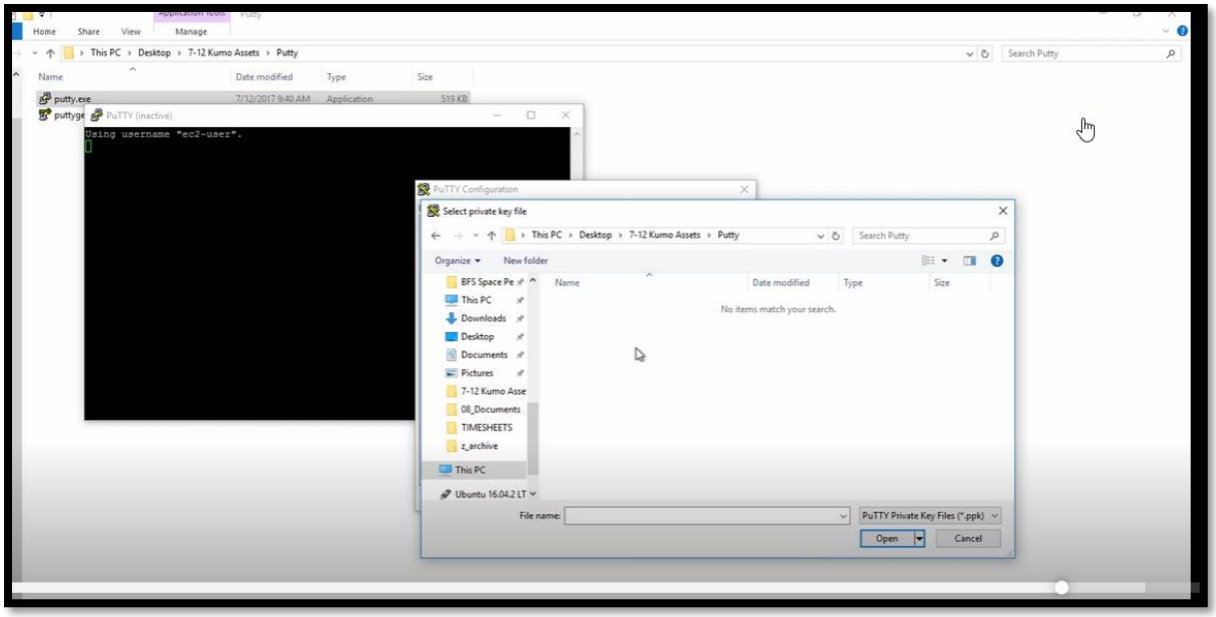
Then **browse for private Key(fig22)**

Fig 22: PuTTY Configuration



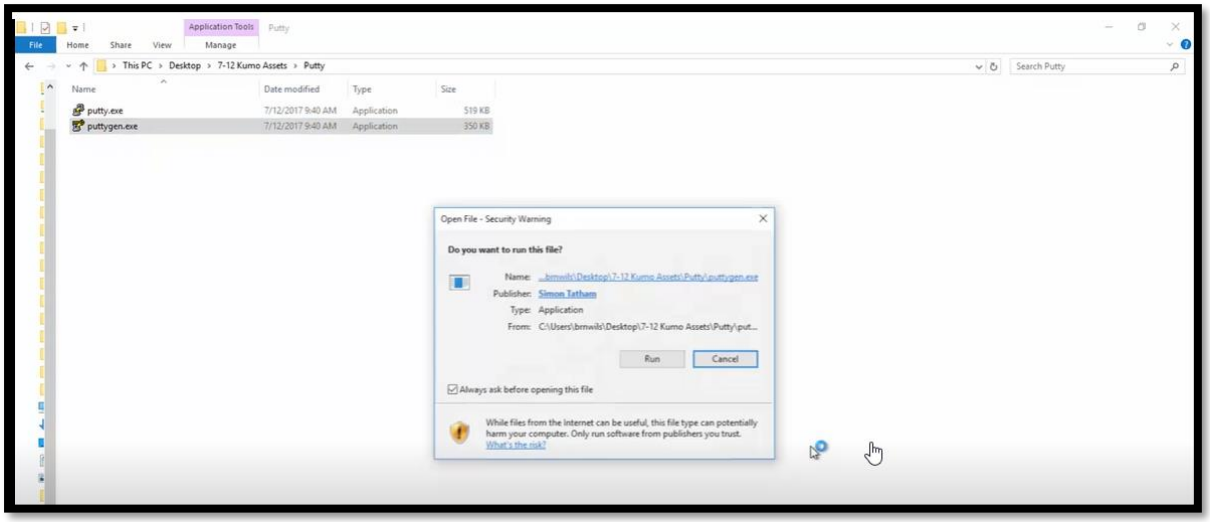
No folder found actually PuTTY requires PPK files(fig23)

Fig 23: Searching PuTTy File



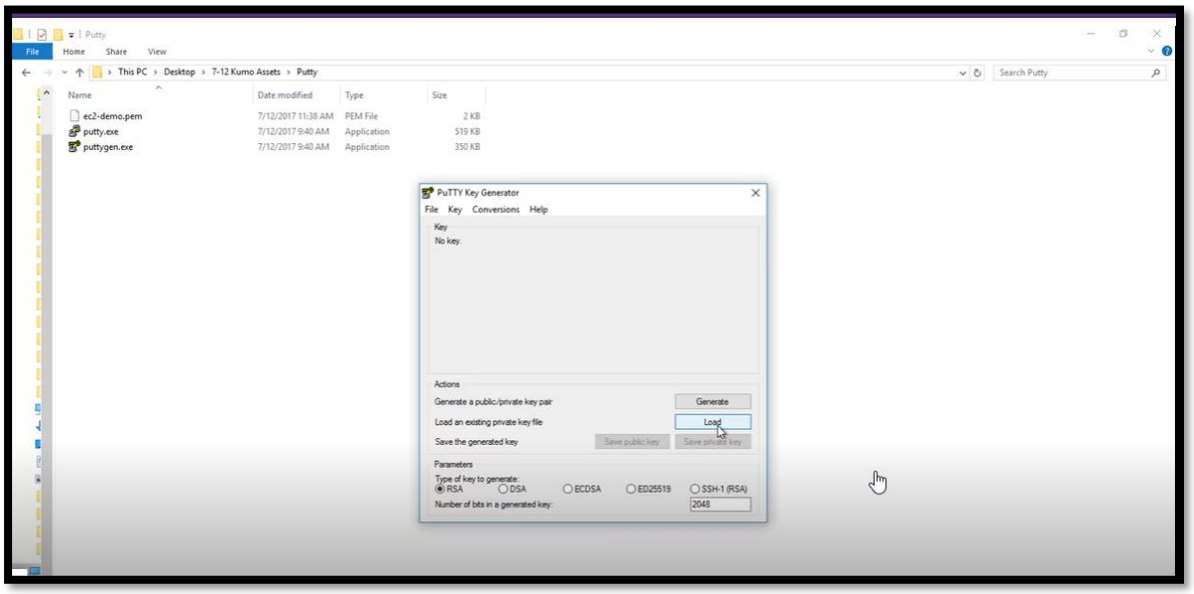
Open puttygen.exe by double clicking and run it.(fig24)

Fig 24: Opening Puttygen file



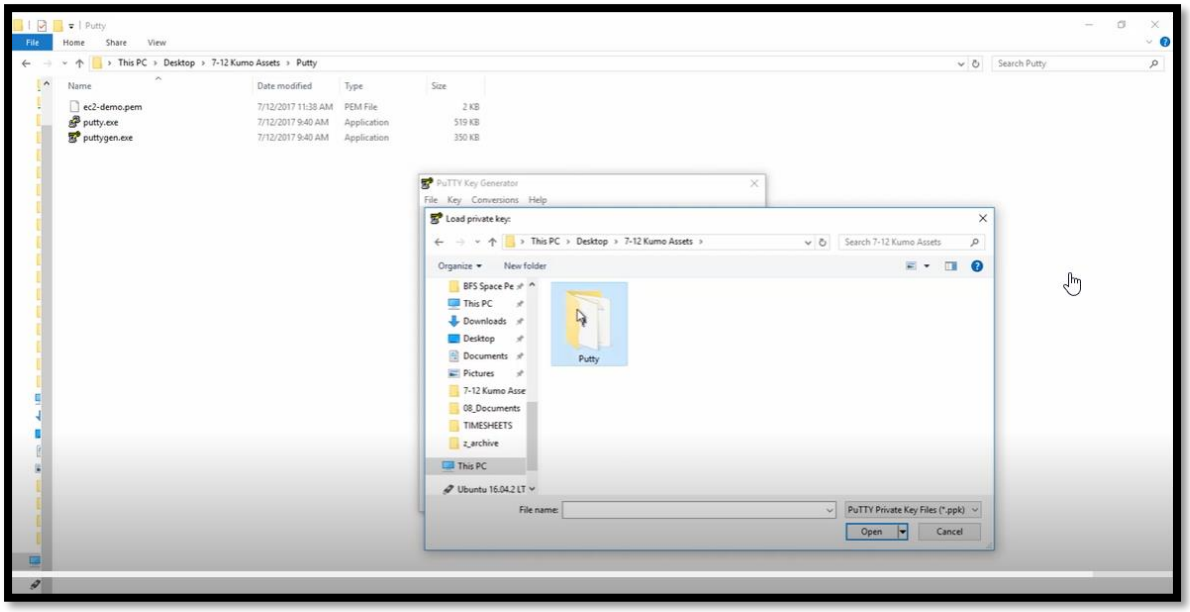
Dialogue box appear and **click load and navigate** to the folder that contains the private key file (.pem) created during the EC2 configuration process.(fig25)

Fig 25: Putty Key Generator



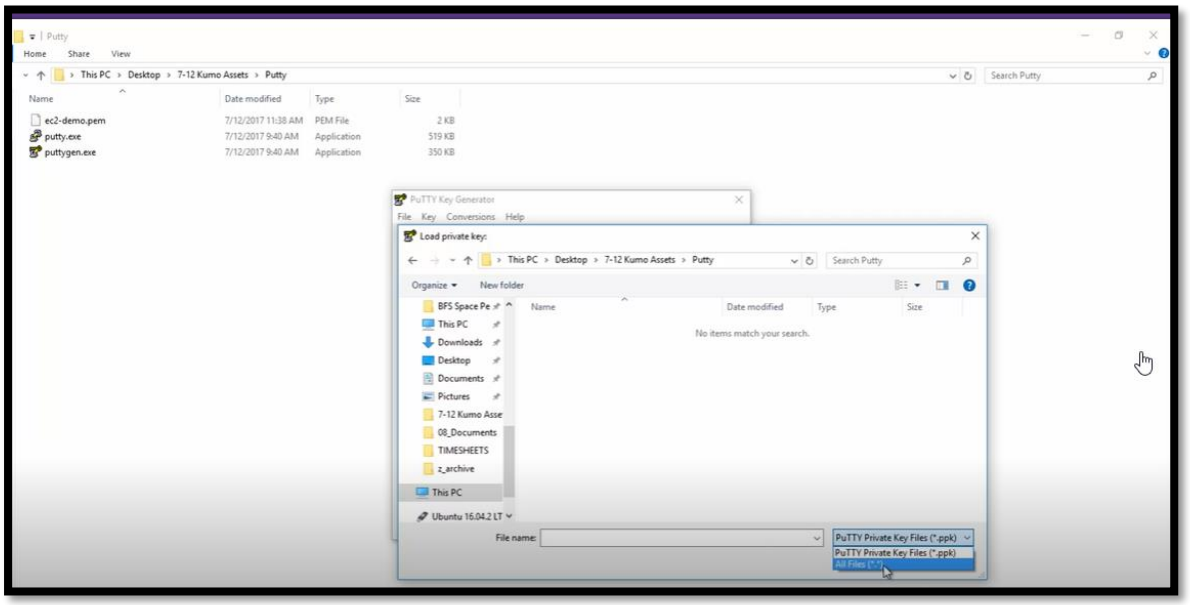
Again, visit same folder Putty.(fig 26)

Fig 26: Opening Putty folder



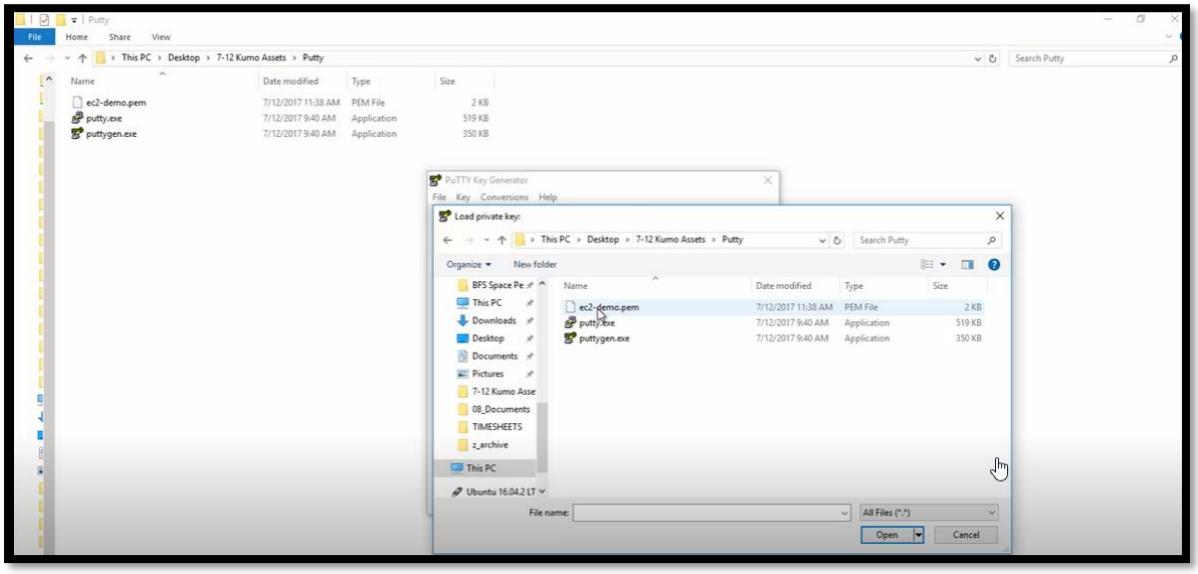
Change file names. **Click on the PuTTY Private Key Files button** in the lower right corner of the window (Fig. 27) and select All Files (*.*).

Fig 27: Fixing File setting



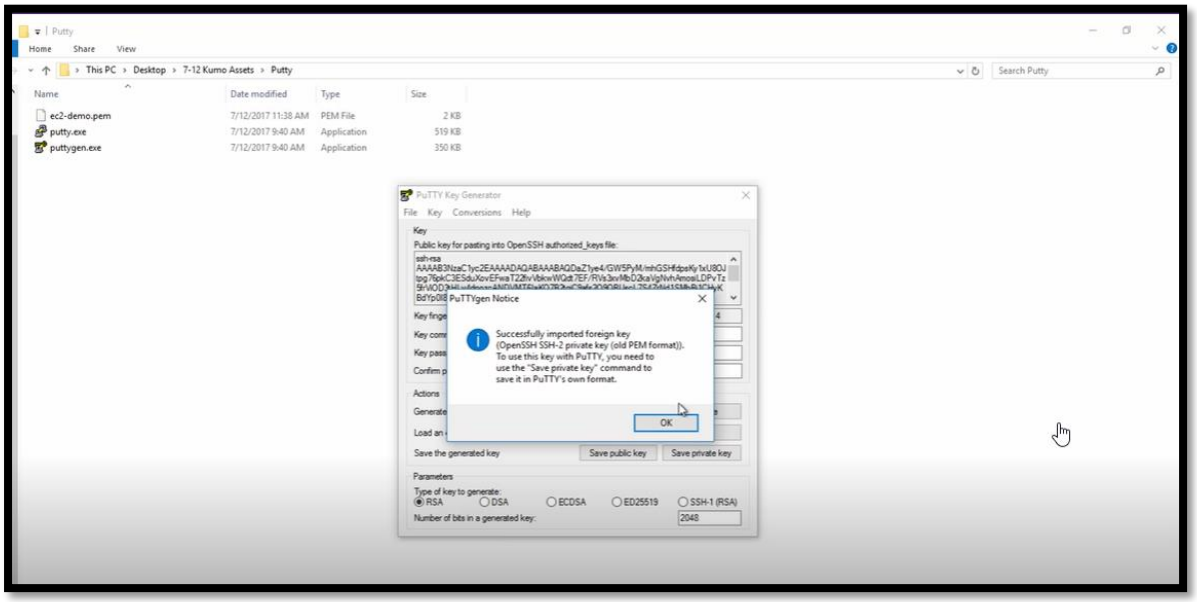
Select your private key file (.pem) and click Open(Fig28)

Fig 28: Opening ec2 demo pem file.



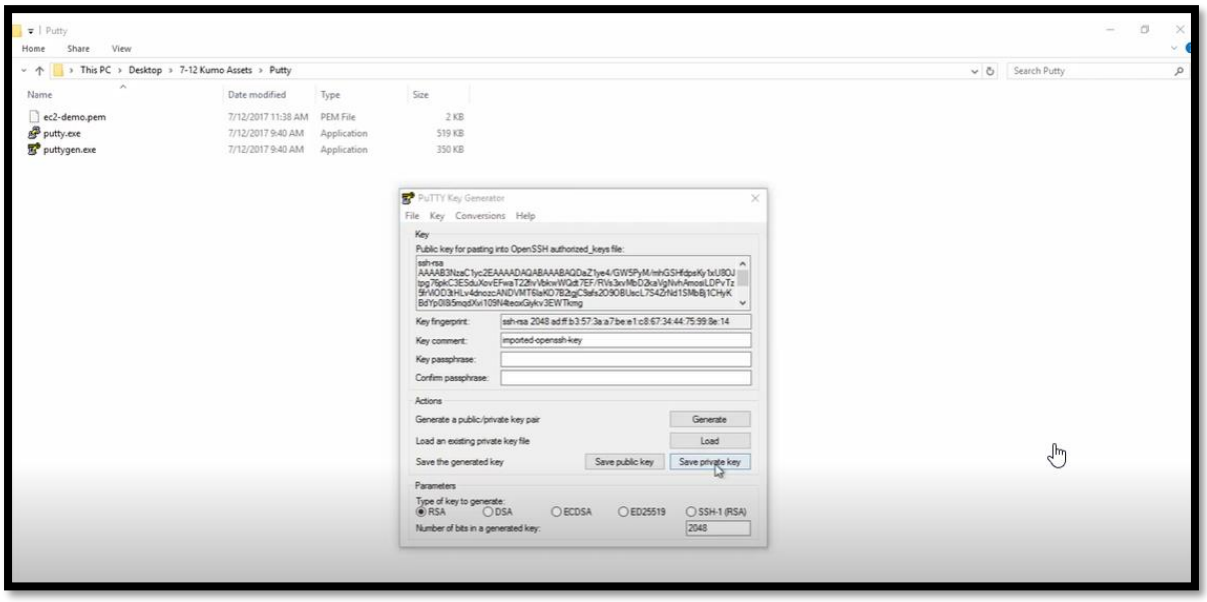
Click OK to close the PuTTYgen Notice pop-up window. In PuTTY Key Generator, make sure Type of key to **generate value is set to RSA** (Fig. 29).

Fig 29: Putty key Generator box



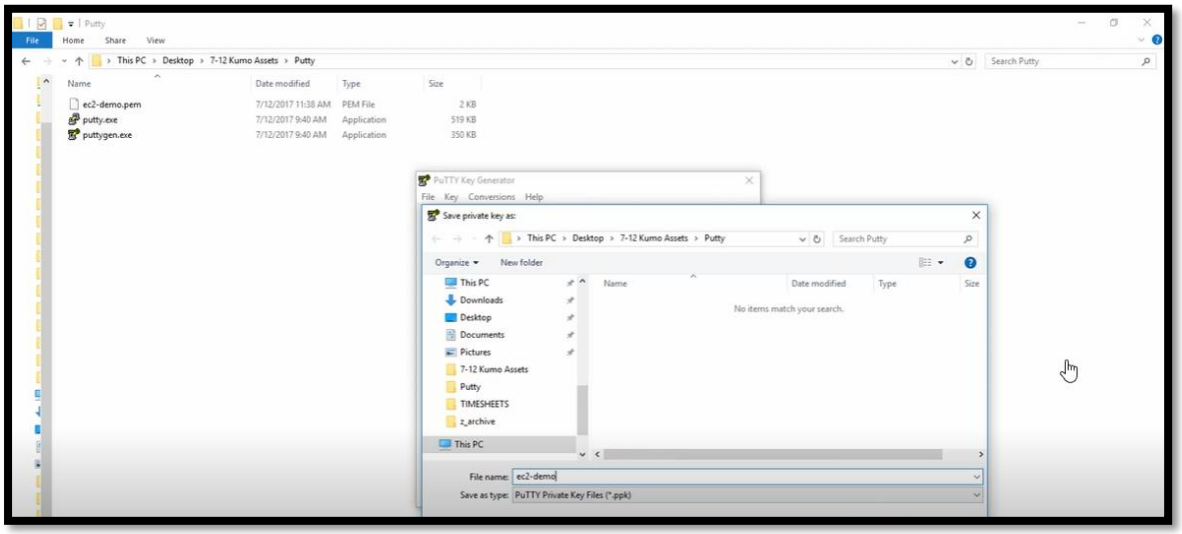
Click **Save private key** and then Yes to close the PuTTYgen Warning pop-up window.(fig30)

Fig 30: Saving private Key



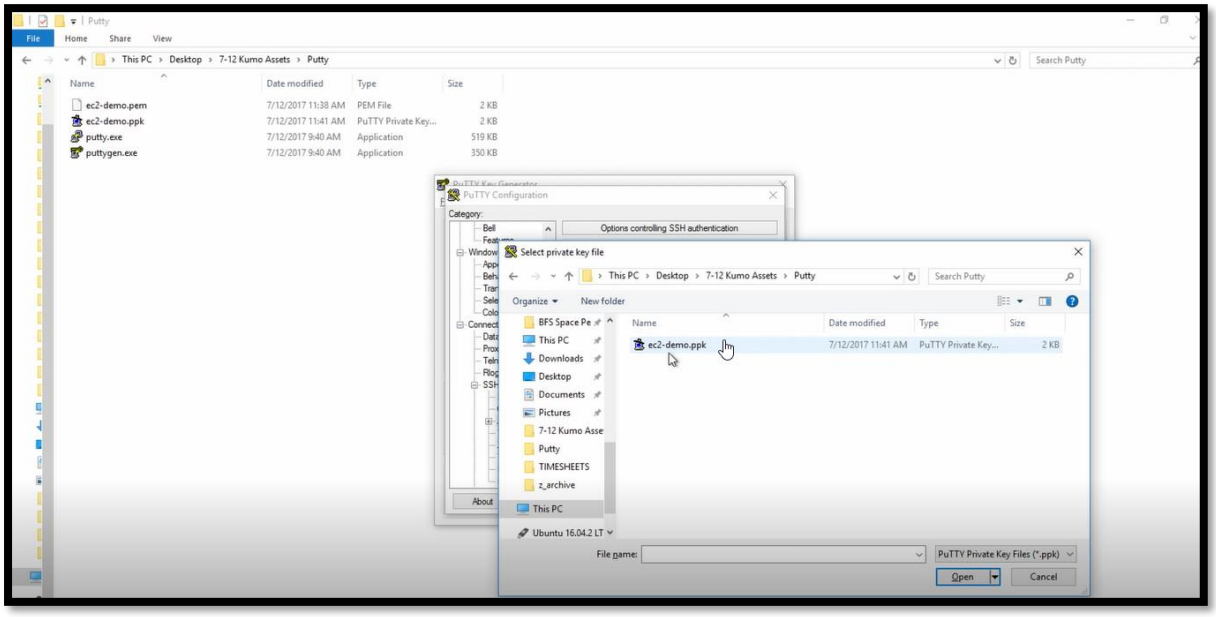
Navigate to the location we want to store your PuTTY Private Key file (.ppk) and give it a name. Click Save. Close the PuTTY Key Generator window.(fig 31)

Fig 31: saving PuTTY private key file



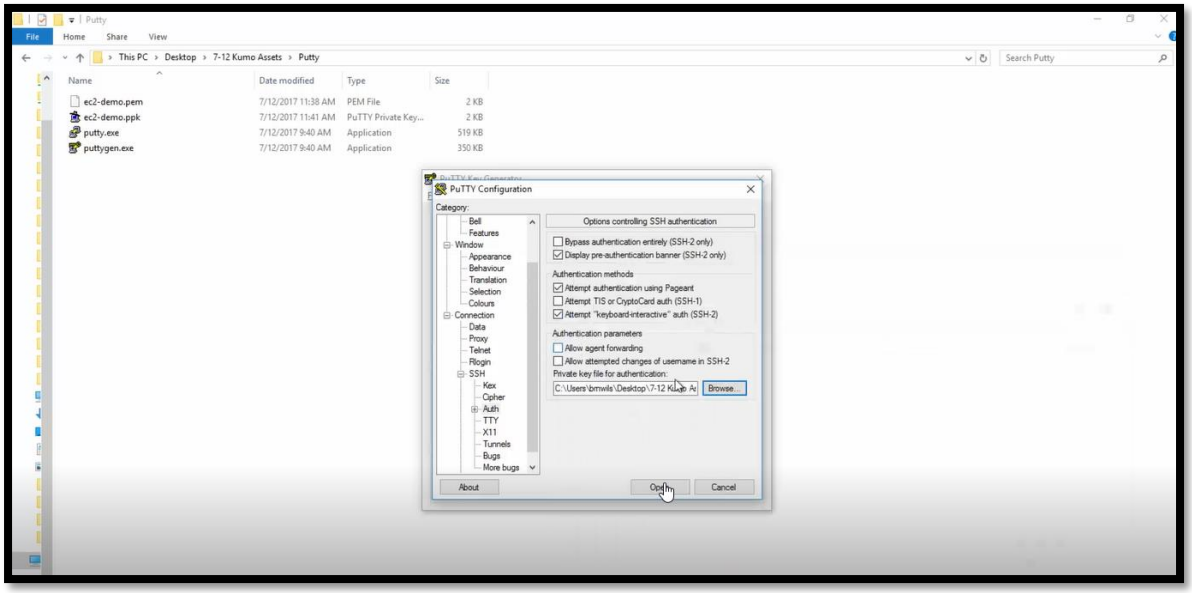
Click on open.(fig32)

Fig 32: Opening ec2 demo-ppk



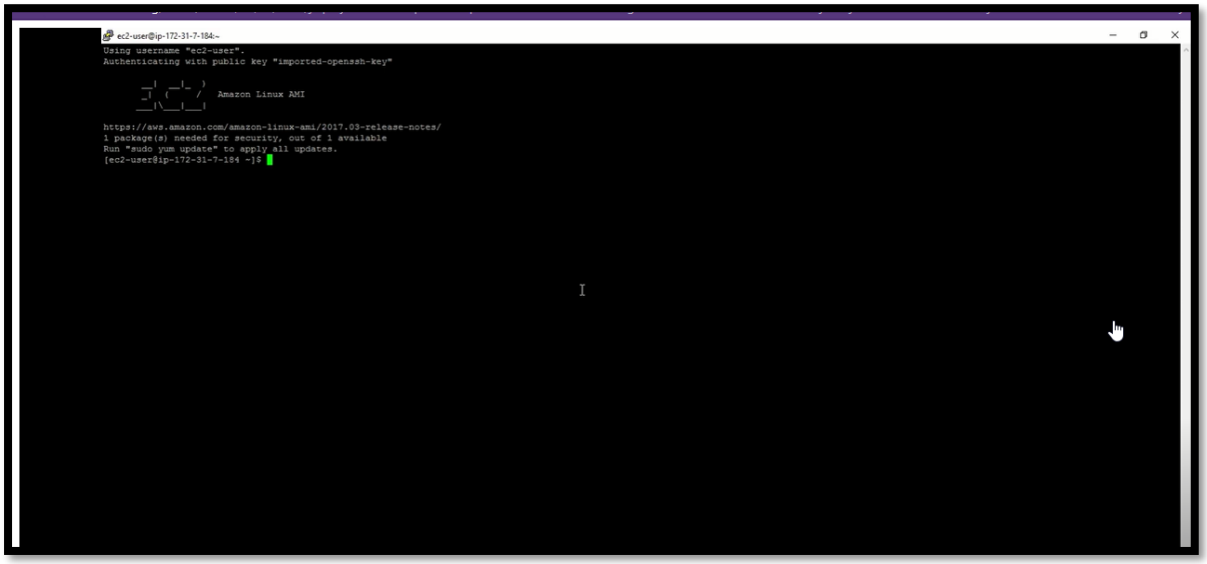
Click on open(Fig32)

Fig 33: Putty Key configuration



You are now connected to your EC2 instance as shown in figure 33.

Fig 33: Successful EC2 Instance connection



Reference

https://www.youtube.com/watch?v=kMMybKqC2Y0

https://aws.amazon.com/ec2/instance-types/

https://www.youtube.com/watch?v=bi7ow5NGC-U&list=RDCMUClGShptNEuvTWGAAfpa2Etw

https://asf.alaska.edu/how-to/data-recipes/connect-to-your-ec2-instance-using-putty-v1-1/

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-ug.pdf#concepts

https://www.aws.training/Details/Video?id=16382

https://www.youtube.com/watch?v=xkXZGvuerRU