

# **Experiment 6**

## **To Launch an AWS EC2 instance and connect to it using PuTTY.**

### ***Elaboration of the terms used:-***

#### **AWS EC2 Instance:-**

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides with complete control of your computing resources and lets you run on Amazon's proven computing environment. There are 5 types of instances:- General purpose, Compute Optimized, Memory optimized, Accelerated Computing and Storage Optimized. We will be using General purpose instance as it provides a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads. These instances are ideal for applications that use these resources in equal proportions such as web servers and code repositories.

#### **PuTTY:-**

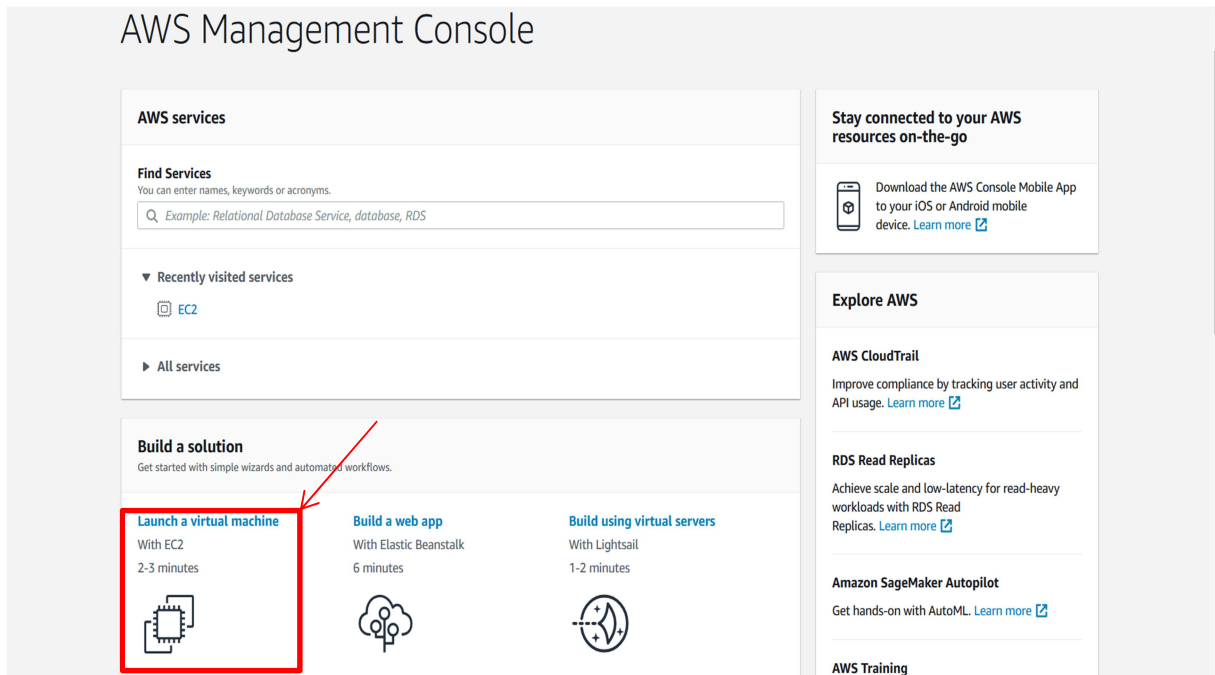
PuTTY is a free and open-source terminal emulator, serial console and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection. It can also connect to a serial port. PuTTY supports many variations on the secure remote terminal, and provides user control over the SSH encryption key and protocol version, alternate ciphers such as AES, 3DES, RC4, Blowfish, DES, and Public-key authentication. PuTTY uses own format of key files – PPK.

#### **SSH:-**

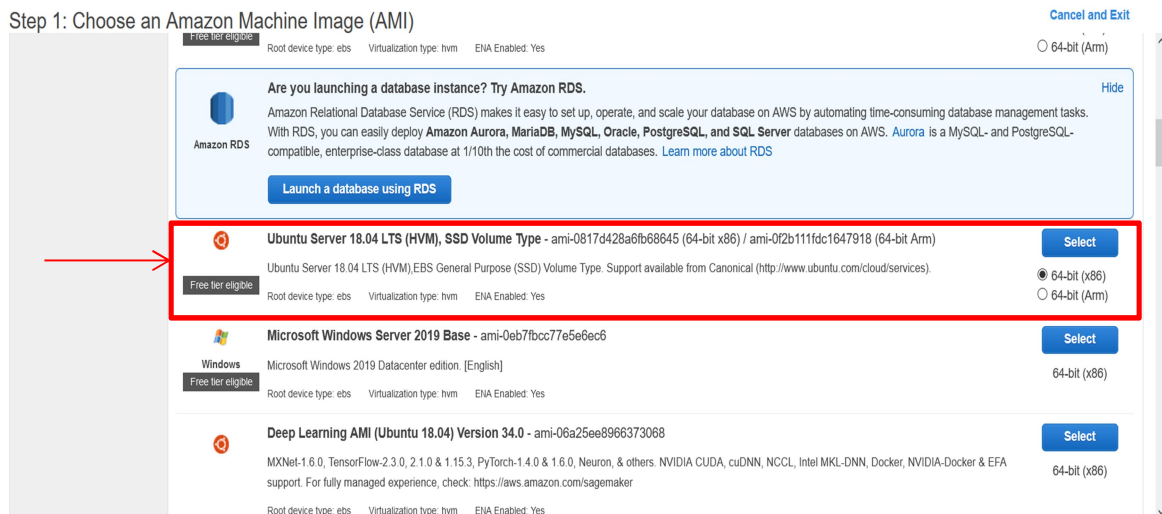
Secure Shell is a cryptographic network protocol for operating network services securely over an unsecured network. Typical applications include remote command-line, login, and remote command execution, but any network service can be secured with SSH. The SSH protocol uses encryption to secure the connection between a client and a server. All user authentication, commands, output, and file transfers are encrypted to protect against attacks in the network.

## Steps to launch and connect to EC2 instance:-

- Go to AWS educate and open AWS Management console. Now, click on Launch a Virtual Machine with EC2.



- Now, choose an Amazon machine language(AMI). It is a template that contains the software configuration (operating system, application server, and applications) required to launch instance. Here we choose **Ubuntu Server 18.04 LTS**



- Choose the Instance type that you require according to your need. Here we select general purpose t2.micro instance which consists of 1 CPU with 1GiB Memory and EBS instance storage. Then click Review and Launch. After reviewing, Launch the instance.

## Step 2: Choose an Instance Type

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes

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

- On launching, create a new key pair. This key pair will allow you to connect to the instance securely. Type a key pair name and download the key pair. Keep it in a safe and accessible location. After this click on Launch Instance

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair name

RavideepS15

Download Key Pair

You have to download the **private key file** (\*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel
Launch Instances

- Click on view instance. You would be able to see the instance that you created as shown below

Launch Instance **Connect** Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6
Ravi15	i-03d3cc11d64295a1e	t2.micro	us-east-1b	running	2/2 checks ...	None	ec2-3-94-253-124.com...	3.94.253.124	-

Note the value of IPv4 public IP as it will be used to connect to the instance. After this click on Connect.

Now choose the connection method as standalone SSH client.

## Connect to your instance

Connection method

☒ A standalone SSH client ⓘ  
☐ Session Manager ⓘ  
☐ EC2 Instance Connect (browser-based SSH connection) ⓘ

To access your instance:

1. Open an SSH client. (find out how to [connect using PuTTY](#))

2. Locate your private key file (RavideepS15.pem). The wizard automatically detects the key you used to launch the instance.

3. Your key must not be publicly viewable for SSH to work. Use this command if needed:  

```
chmod 400 RavideepS15.pem
```

4. Connect to your instance using its Public DNS:  

```
ec2-3-94-253-124.compute-1.amazonaws.com
```

Example:

```
ssh -i "RavideepS15.pem" ubuntu@ec2-3-94-253-124.compute-1.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our [connection documentation](#).

Close

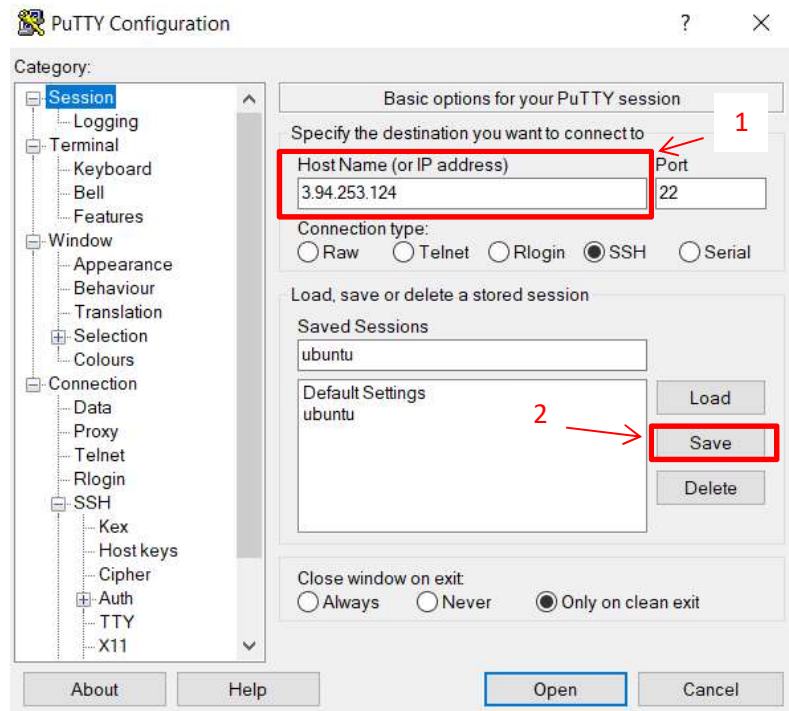
- Now download PuTTY from the internet and install it. After installation open PuTTYgen to generate a key. After opening, select the private key file that you downloaded before and load it. Also put a key passphrase. This is a password that you would need to enter for accessing the instance you created. After this select parameter as RSA and click on save private key and download the key.

The screenshot shows the PuTTY Key Generator window. Red annotations highlight specific areas:

- 1**: Points to the 'Generate' button.
- 2**: Points to the 'Key passphrase' and 'Confirm passphrase' fields.
- 3**: Points to the 'Type of key to generate' section, specifically the 'RSA' radio button.
- 4**: Points to the 'Load' and 'Save private key' buttons.

The 'Key' section shows a public key for pasting into the OpenSSH authorized\_keys file, a key fingerprint, and a key comment. The 'Actions' section includes buttons for 'Generate', 'Load', 'Save public key', and 'Save private key'. The 'Parameters' section shows 'Type of key to generate' with 'RSA' selected, and 'Number of bits in a generated key' set to 2048.

- Now open PuTTY. Under sessions category, put host name as the IPv4 Public address that you noted in the beginning and save the session by giving a name (Ubuntu here).



Under Auth section, browse the putty file that you generated thorough PuTTYgen in the previous step. Click open. A command window will open. Type the name of the session in login as and password in passphrase section.

3.94.253.124 - PuTTY

```
login as: ubuntu
Authenticating with public key "imported-openssh-key"
Passphrase for key "imported-openssh-key":
```

After the correct input of username and password, the command line of the Ubuntu operating system will open

```
ubuntu@ip-172-31-45-160: /
Swap usage: 0%
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-172-31-45-160:~$ cd /
ubuntu@ip-172-31-45-160:/$ ls
bin      home      lib64      opt       sbin      tmp        vmlinuz.old
boot     initrd.img lost+found  proc      snap      usr
dev      initrd.img.old media      root      srv       var
etc      lib       mnt       run       sys       vmlinuz
ubuntu@ip-172-31-45-160:/$
```