
Creating a Linux Instance Using EC2

[Edition 13]

[Last Update 221117]

Contents

1	Introduction	3
2	Documentation Links	4
3	Pre-requisite	5
4	Launching a Linux Instance Using EC2	6
4.1	Creating an EC2 Instance.....	6
4.2	Accessing EC2 Instance Using Browser SSH Connection.....	15
4.3	Accessing EC2 Instance Using Putty	18
4.4	Accessing EC2 Instance Using MobaXterm	26
5	Deleting/Cleanup	30
5.1	Stopping the Linux Instance.....	30
5.2	Terminating the Linux Instance	31
6	Troubleshooting	33
7	Summary.....	34

1 INTRODUCTION

Amazon EC2 presents a true virtual computing environment, allowing you to use web service interfaces to launch instances with a variety of operating systems, load them with your custom application environment, manage your network's access permissions, and run your image using as many or few systems as you desire.

To use Amazon EC2, you simply:

- Select a pre-configured, templated Amazon Machine Image (AMI) to get up and running immediately, or create an AMI containing your applications, libraries, data, and associated configuration settings.
- Configure security and network access on your Amazon EC2 instance.
- Choose which instance type(s) you want, then start, terminate, and monitor as many instances of your AMI as needed, using the web service APIs or the variety of management tools provided.
- Determine whether you want to run in multiple locations, utilize static IP endpoints, or attach persistent block storage to your instances.
- Pay only for the resources that you actually consume, like instance-hours or data transfer.

This activity guide cover steps for:

1. Launching a Linux Instance using EC2
 - a. Creating an EC2 Instance
 - b. Accessing EC2 Instance Using Browser SSH Connection
 - c. Accessing EC2 Instance Using Putty
2. Deleting/Cleanup
 - a. Stopping the Linux Instance
 - b. Terminating the Linux Instance

2 DOCUMENTATION LINKS

1. Amazon EC2
<https://aws.amazon.com/ec2/>
2. Features of Amazon EC2
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>
3. Amazon EC2 FAQs
<https://aws.amazon.com/ec2/faqs/>
4. Getting Started with Amazon EC2 linux Instances
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html

3 PRE-REQUISITE

1. An AWS Account (Free or Paid).

K21Academy

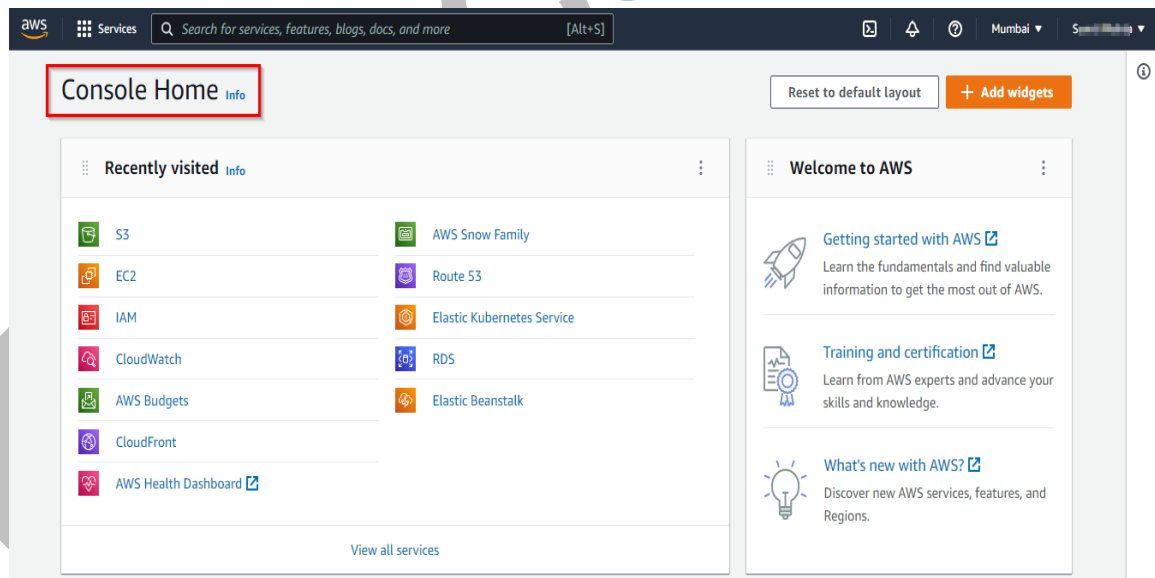
4 LAUNCHING A LINUX INSTANCE USING EC2

Note: Launching a Linux Instance means you will create a Virtual Machine and access it using tools like Putty, MobaXterm etc. You shall be accessing this Virtual machine using internet, accessing on internet means these machines are physically present in the datacenters that lie in different Regions in the world. A datacenter is a place used to house computer systems and associated components, such as routers, switches, firewalls, storage systems, servers etc. So, you are basically accessing the machine present in such datacenters from anywhere in the world via internet. The launching of an instance does not depend on the underlying infrastructure whether your Laptop is having Windows or macOS.

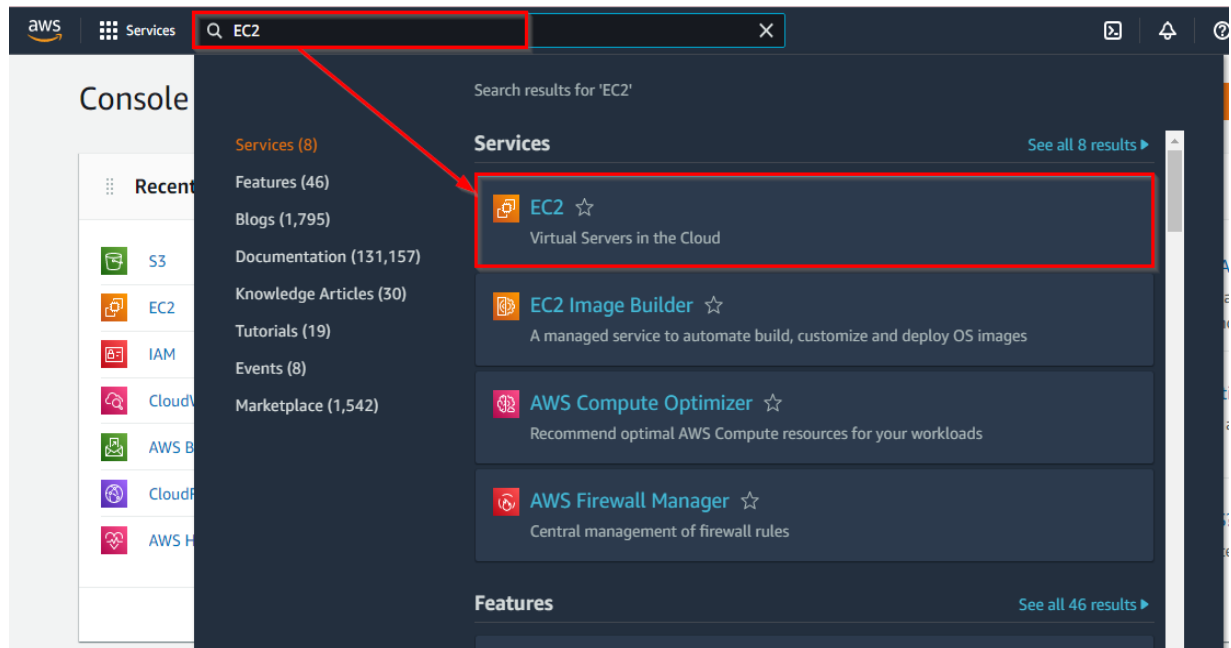
4.1 Creating an EC2 Instance

Note: Amazon EC2 basically provides a **Virtual Computing environment**, where they have pre-configured AMI (Amazon Machine Image) from which you can launch Virtual machines. In this section, we shall see the steps to create and configure the Linux Ec2 instance from a pre-configured AMI named Linux-2 AMI.

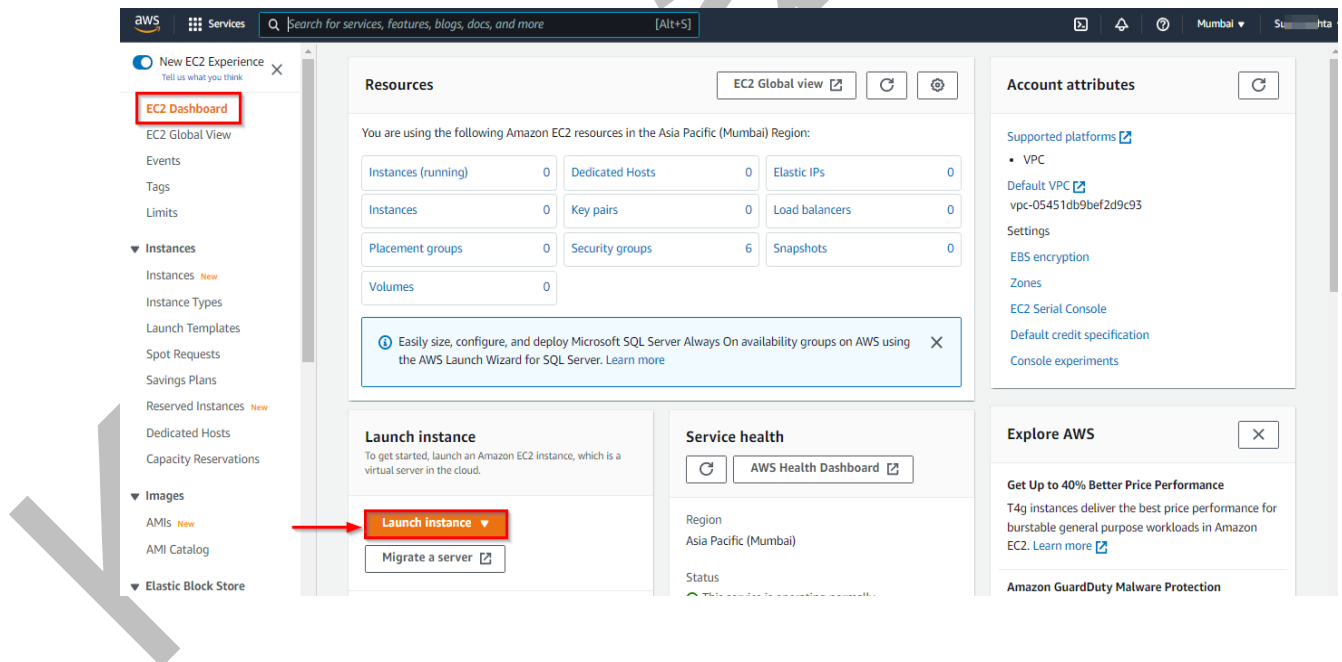
1. Log in to your AWS console.



2. Search for **EC2** in the search bar and click on it.



3. Once you are at the EC2 Dashboard, click on **Launch** to launch an instance.



4. In the Name and Tags step you can add tags to an instance, here tags help you to enable categorize AWS resources in different ways, for example, by owner, environment, or purpose.

EC2 > Instances > Launch an instance

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

k21-linux-server [Add additional tags](#)

5. Choose **Amazon Linux**, select **Amazon Linux 2 AMI (HVM)** architecture and Click on **Select**.

Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents **Quick Start**

Amazon Linux
macOS
Ubuntu
Windows
Red Hat

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
ami-05fa00d4c63e32376 (64-bit (x86)) / ami-05f3141013eebdc12 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs Free tier eligible

Description
Amazon Linux 2 Kernel 5.10 AMI 2.0.20220805.0 x86_64 HVM gp2

Architecture AMI ID
64-bit (x86) ami-05fa00d4c63e32376 Verified provider

Summary

Number of instances Info
1

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-05fa00d4c63e32376

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of

Cancel Launch Instance

Note: Here, in the above step we are selecting the Amazon Linux 2 AMI which is basically preconfigured with some tools installed on it like Amazon Linux 2 AMI comes with AWS CLI pre-installed. AWS CLI is basically the Command Line Interface provided by AWS to access any of their services using CLI.

6. For Select the **t2.micro** instance type, if you want you may select another instance type but they are chargeable so we choose the t2.micro instance type which is eligible for the free tier and limited resources.

▼ Instance type Info

Instance type

t2.micro Free tier eligible
Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0124 USD per Hour
On-Demand Windows pricing: 0.017 USD per Hour

Q

t2.nano
Family: t2 1 vCPU 0.5 GiB Memory
On-Demand Linux pricing: 0.0062 USD per Hour
On-Demand Windows pricing: 0.0085 USD per Hour

t2.micro Free tier eligible
Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0124 USD per Hour
On-Demand Windows pricing: 0.017 USD per Hour

t2.small
Family: t2 1 vCPU 2 GiB Memory
On-Demand Linux pricing: 0.0248 USD per Hour
On-Demand Windows pricing: 0.034 USD per Hour

t2.medium
Family: t2 2 vCPU 4 GiB Memory
On-Demand Linux pricing: 0.0496 USD per Hour
On-Demand Windows pricing: 0.0676 USD per Hour

t2.large
Family: t2 2 vCPU 8 GiB Memory
On-Demand Linux pricing: 0.0992 USD per Hour
On-Demand Windows pricing: 0.1272 USD per Hour

Note: Here we must choose the instance type as **t2.micro**, because it is free tier eligible, i.e. you won't be charge for running the instance with this instance type as we need to perform the labs being in free tier. So, please make sure you select the t2.micro as the instance type else you shall be charged for running instances from Paid AMI's.

7. Select an existing key pair or create a new one, we will Create a new one, enter the name of the Key-pair as Linux-Key, select **.ppk** and Create the Key Pair.

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Select

Create new key pair

Create key pair

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#)

Key pair name

linux-key

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA
RSA encrypted private and public key pair

☐ ED25519
ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format

☐ .pem
For use with OpenSSH

☒ .ppk
For use with PuTTY

Cancel Create key pair

8. Now select HTTP Port under Network Setting.

Network settings Info Edit

Network Info
vpc-05451db9bef2d9c93

Subnet Info
No preference (Default subnet in any availability zone)

Auto-assign public IP Info
Enable

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

We'll create a new security group called 'launch-wizard-3' with the following rules:

☒ Allow SSH traffic from
Helps you connect to your instance Anywhere
0.0.0.0/0

☐ Allow HTTPs traffic from the internet
To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

Note: We add http rule here because we need to access our Webpage. As the **communication** for request and response between Webserver and Browser **happens on port 80** which is **http**.

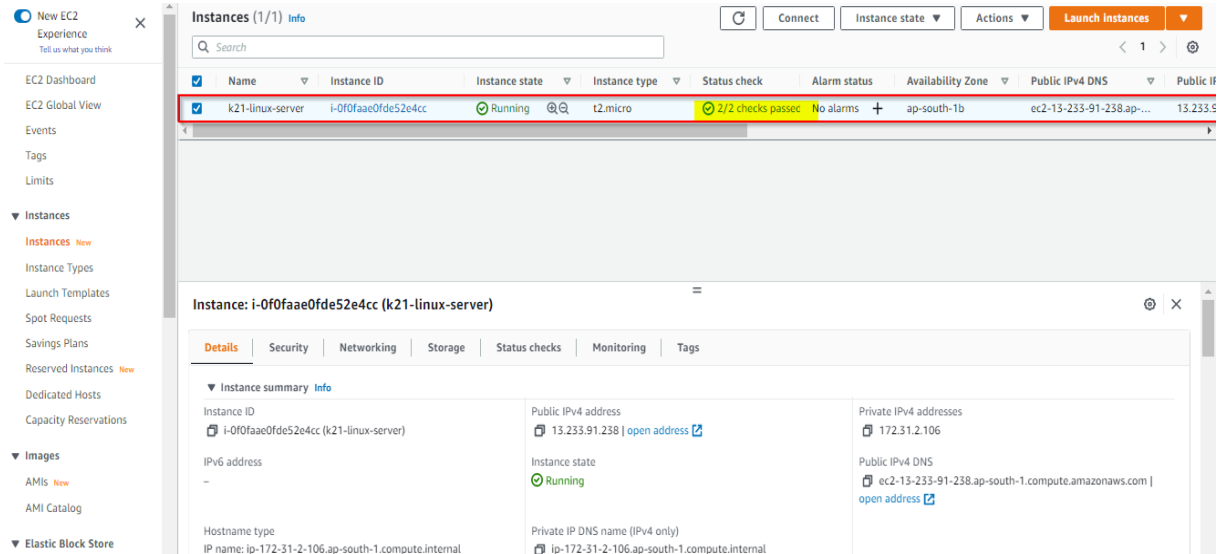
9. Now review all the things you have Configured and Click on **Launch Instance**.

10. Now Click on **View all Instances**.

11. Here, you shall see your instance is launching and the **Status check** is **Initializing**, wait for some time.

Instances (1) Info									
Search									
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
<input type="checkbox"/>	k21-linux-server	i-0f0faae0fde52e4cc	Running	t2.micro	Initializing	No alarms +	ap-south-1b	ec2-13-233-91-238.ap-...	13.233.9

12. Refresh and you shall see your instance are Up and Running, and **Status check** has changed to **2/2 checks**.



The screenshot displays the AWS Management Console interface for EC2 instances. The top navigation bar shows the 'Instances (1/1)' tab selected. Below the navigation bar, a table lists the instances. The instance 'k21-linux-server' with ID 'i-0f0faae0fde52e4cc' is shown in the 'Running' state. The 'Status check' column indicates '2/2 checks passed'. The details pane on the right shows the instance summary for 'k21-linux-server', including the public IP address '13.233.91.238' and the private IP address '172.31.2.106'.

Now, we have successfully created our first Linux Instance using EC2.

Note: Now, wait for the Status check of the instance to change to 2/2 checks pass as the instance is launching from an AMI and it needs time to install all the preconfigured things like AWS CLI and other tools. So, please wait till the Status check becomes 2/2 checks pass and then only proceed with further steps, and if status check has not passed then you shall get error in the further steps so please wait.

Note: Now we have launched the instance successfully, further we need to access the instance so, we have two ways for accessing an instance-

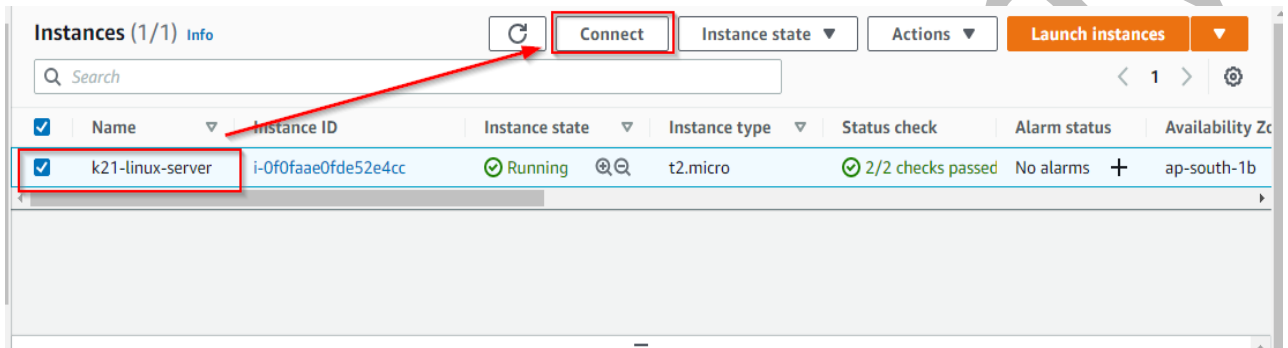
- Accessing via Browser using EC2 instance Connect.
- Accessing using Putty (a Secure Shell SSH client).

In further steps we shall see both the ways to access Linux EC2 instance

4.2 Accessing EC2 Instance Using Browser SSH Connection

Note: Till now, we have launched an instance successfully, but we need to access it to run commands so for accessing our instance the easy way is you can access the CLI directly from the browser using EC2 instance connect that we shall see in further steps.

1. Select your Instance and Click on **Connect**.



2. Under EC2 Instance Connect, let everything default and Click on Connect.

EC2 > Instances > i-008a9c8d0fa6f45bb > Connect to instance

Connect to instance [Info](#)

Connect to your instance i-008a9c8d0fa6f45bb (k21-linux-server) using any of these options

EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Instance ID
i-008a9c8d0fa6f45bb (k21-linux-server)

Public IP address
100.26.112.205

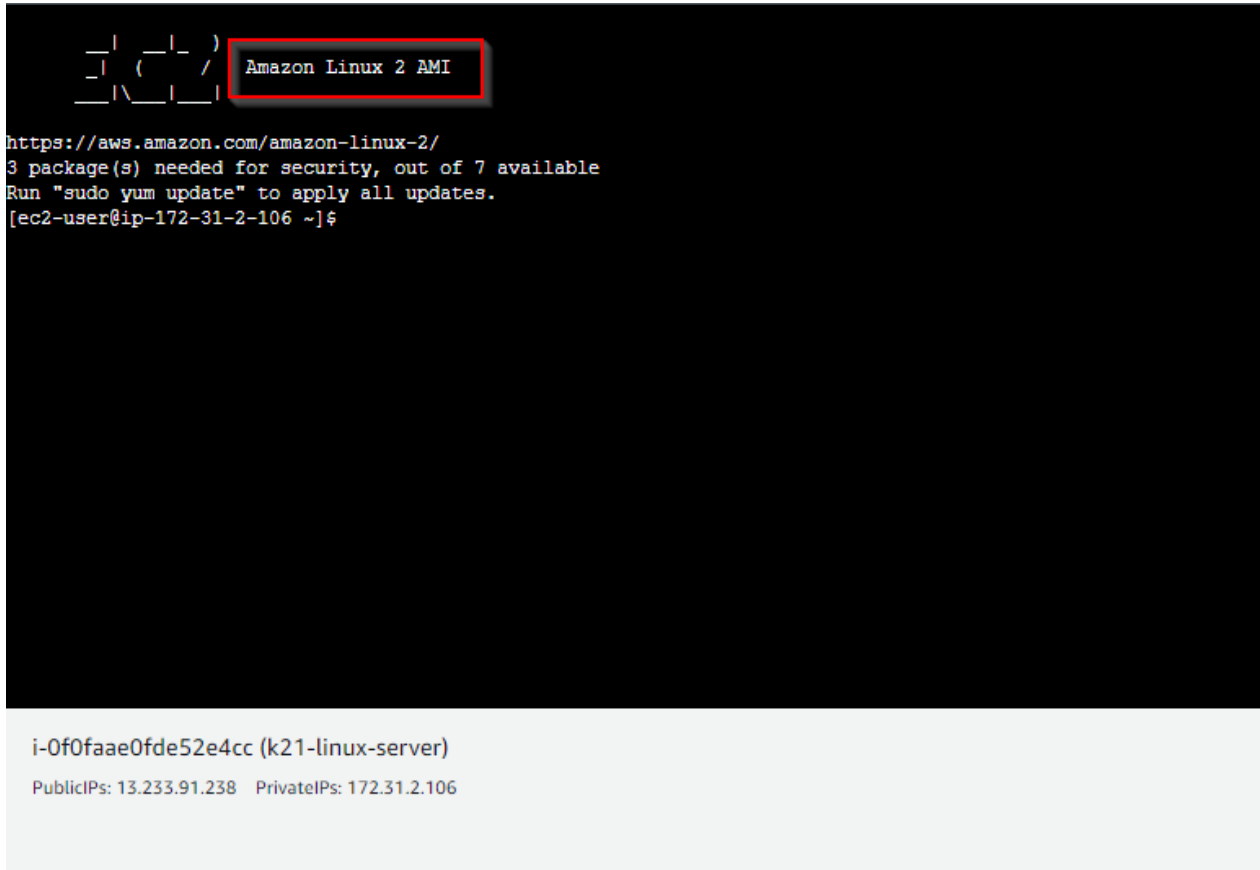
User name

Connect using a custom user name, or use the default user name ec2-user for the AMI used to launch the instance.

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel **Connect**

3. Now you shall be redirected to new window and you shall see the Linux CLI.



```

 _ _ _ _ _
 _ | ( _ | /
 _ | \ _ | _ |

Amazon Linux 2 AMI

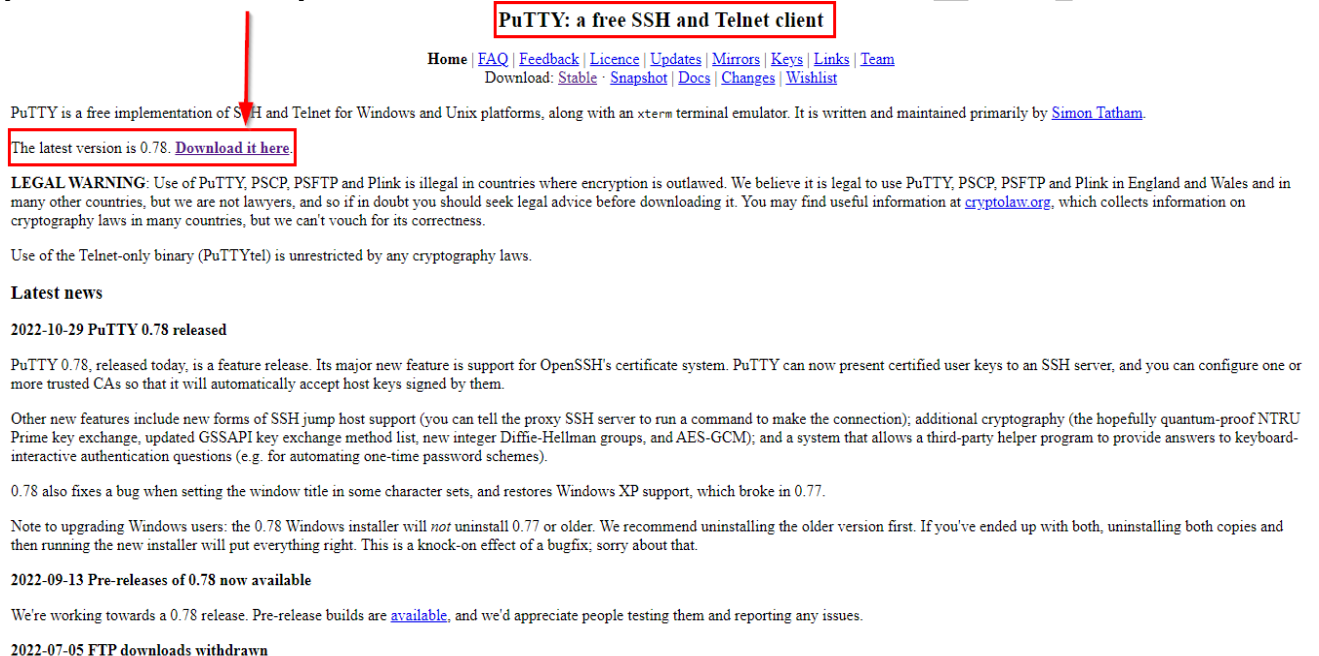
https://aws.amazon.com/amazon-linux-2/
3 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-2-106 ~]$
```

i-0f0faae0fde52e4cc (k21-linux-server)
PublicIPs: 13.233.91.238 PrivateIPs: 172.31.2.106

4.3 Accessing EC2 Instance Using Putty

Note: PuTTY is an SSH and telnet client. 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. In this section, we will **download and install Putty** and will try to SSH (Secure Shell) into our instance using it.

1. Download and Install Putty from <http://www.chiark.greenend.org.uk/~sgtatham/putty/> Once you Click on the link you shall see this screen, Click on **Download it here**.



PuTTY: a free SSH and Telnet client

Home | [FAQ](#) | [Feedback](#) | [Licence](#) | [Updates](#) | [Mirrors](#) | [Keys](#) | [Links](#) | [Team](#)
Download: [Stable](#) · [Snapshot](#) | [Docs](#) | [Changes](#) | [Wishlist](#)

PuTTY is a free implementation of SSH and Telnet for Windows and Unix platforms, along with an xterm terminal emulator. It is written and maintained primarily by [Simon Tatham](#).

The latest version is 0.78. [Download it here](#).

LEGAL WARNING: Use of PuTTY, PSCP, PSFTP and Plink is illegal in countries where encryption is outlawed. We believe it is legal to use PuTTY, PSCP, PSFTP and Plink in England and Wales and in many other countries, but we are not lawyers, and so if in doubt you should seek legal advice before downloading it. You may find useful information at [cryptolaw.org](#), which collects information on cryptography laws in many countries, but we can't vouch for its correctness.

Use of the Telnet-only binary (PuTTYtel) is unrestricted by any cryptography laws.

Latest news

2022-10-29 PuTTY 0.78 released

PuTTY 0.78, released today, is a feature release. Its major new feature is support for OpenSSH's certificate system. PuTTY can now present certified user keys to an SSH server, and you can configure one or more trusted CAs so that it will automatically accept host keys signed by them.

Other new features include new forms of SSH jump host support (you can tell the proxy SSH server to run a command to make the connection); additional cryptography (the hopefully quantum-proof NTRU Prime key exchange, updated GSSAPI key exchange method list, new integer Diffie-Hellman groups, and AES-GCM); and a system that allows a third-party helper program to provide answers to keyboard-interactive authentication questions (e.g. for automating one-time password schemes).

0.78 also fixes a bug when setting the window title in some character sets, and restores Windows XP support, which broke in 0.77.

Note to upgrading Windows users: the 0.78 Windows installer will *not* uninstall 0.77 or older. We recommend uninstalling the older version first. If you've ended up with both, uninstalling both copies and then running the new installer will put everything right. This is a knock-on effect of a bugfix; sorry about that.

2022-09-13 Pre-releases of 0.78 now available

We're working towards a 0.78 release. Pre-release builds are [available](#), and we'd appreciate people testing them and reporting any issues.

2022-07-05 FTP downloads withdrawn

2. Now Click on **64-bit x86** and install it once downloaded.

Download PuTTY: latest release (0.78)

[Home](#) | [FAQ](#) | [Feedback](#) | [Licence](#) | [Updates](#) | [Mirrors](#) | [Keys](#) | [Links](#) | [Team](#)
Download: [Stable](#) | [Snapshot](#) | [Docs](#) | [Changes](#) | [Wishlist](#)

This page contains download links for the latest released version of PuTTY. Currently this is 0.78, released on 2022-10-29.

When new releases come out, this page will update to contain the latest, so this is a good page to bookmark or link to. Alternatively, here is a [permanent link to the 0.78 release](#).

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date version of the code available. If you have a problem with this release, then it might be worth trying out the [development snapshots](#), to see if the problem has already been fixed in those versions.

Package files

You probably want one of these. They include versions of all the PuTTY utilities (except the new and slightly experimental Windows pterm).

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

We also publish the latest PuTTY installers for all Windows architectures as a free-of-charge download at the [Microsoft Store](#); they usually take a few days to appear there after we release them.

MSI ("Windows Installer")

64-bit x86:	putty-64bit-0.78-installer.msi	(signature)
64-bit Arm:	putty-arm64-0.78-installer.msi	(signature)
32-bit x86:	putty-0.78-installer.msi	(signature)

Unix source archive

.tar.gz:	putty-0.78.tar.gz	(signature)
----------	-----------------------------------	-----------------------------

3. Now Select your instance and Copy the **public IPv4 address** under Details as shown below.

Instances (1/1) Info

Search

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
k21-linux-server	i-0f0faae0fde52e4cc	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-13-233-91-238.ap-south-1.compute.amazonaws.com	13.233.91.238

Instance: i-0f0faae0fde52e4cc (k21-linux-server)

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

Instance ID
i-0f0faae0fde52e4cc (k21-linux-server)

IPv6 address
-

Hostname type
IP name: ip-172-31-2-106.ap-south-1.compute.internal

Answer private resource DNS name

Public IPv4 address
13.233.91.238 | [open address](#)

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-2-106.ap-south-1.compute.internal

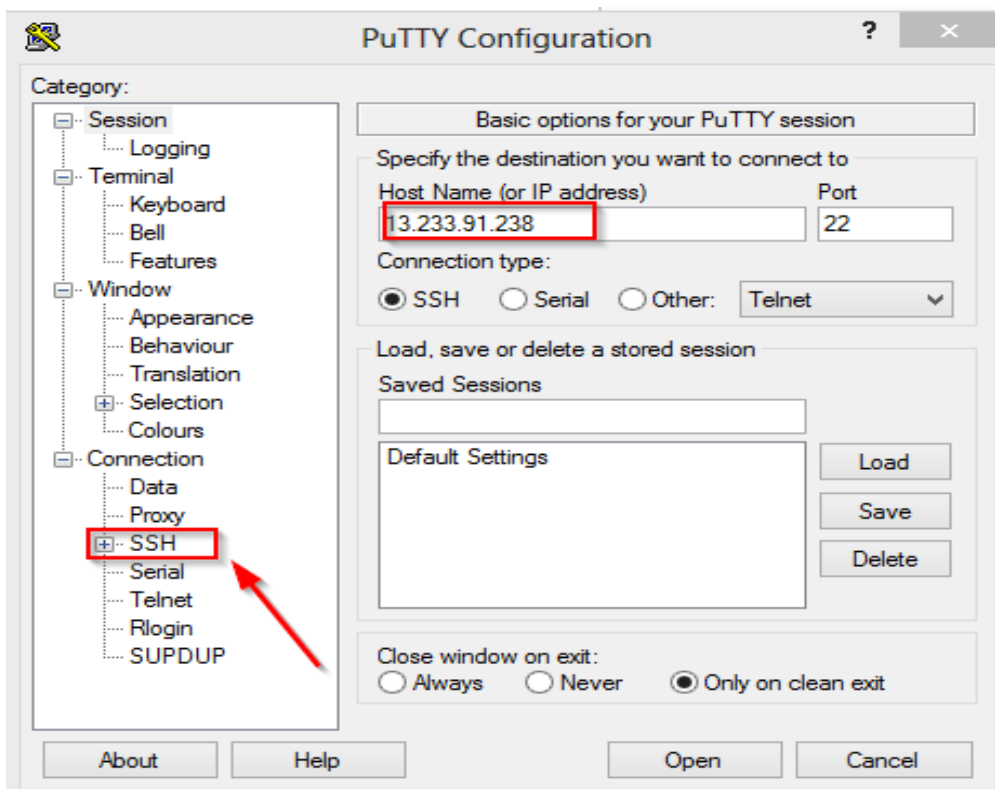
Instance type

Private IPv4 addresses
172.31.2.106

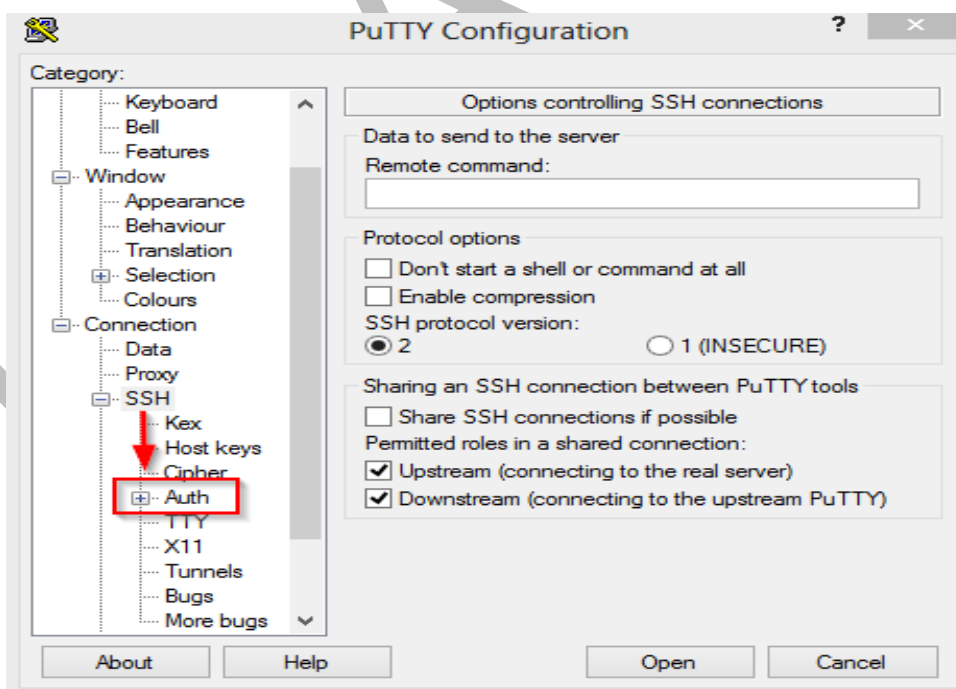
Public IPv4 DNS
ec2-13-233-91-238.ap-south-1.compute.amazonaws.com | [open address](#)

Elastic IP addresses

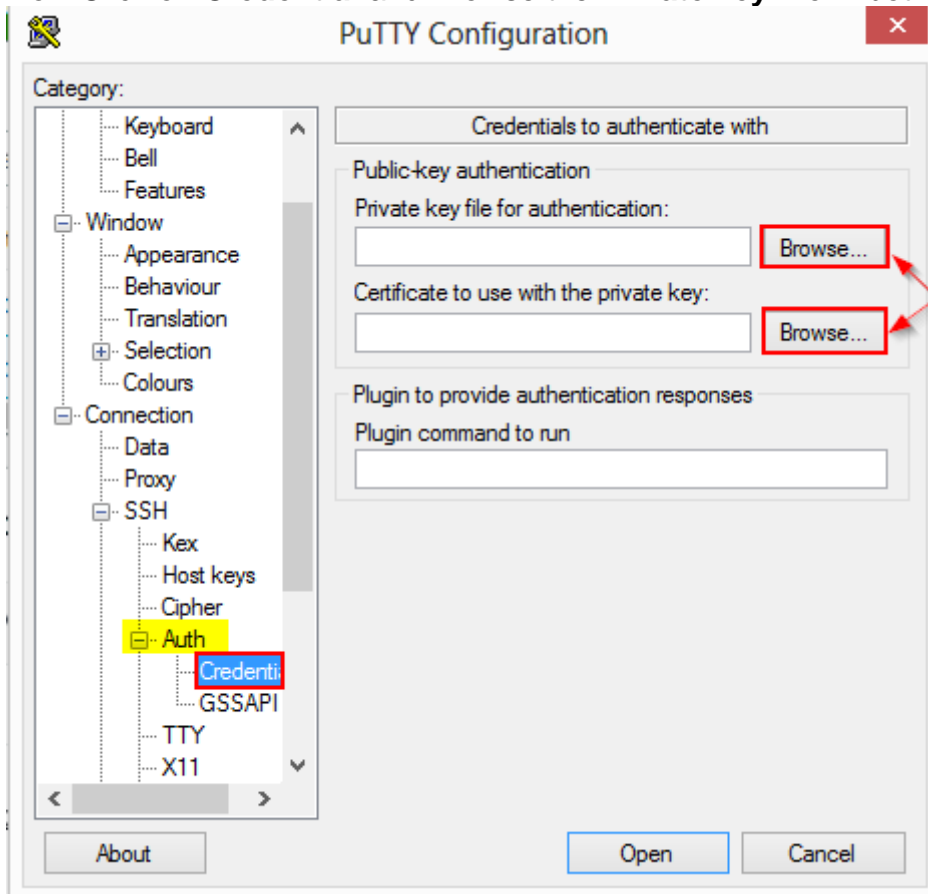
- Now Open **PuTTY** and paste the Public-ip address under Host Name (or IP address) you copied and click on **plus mark on SSH**.



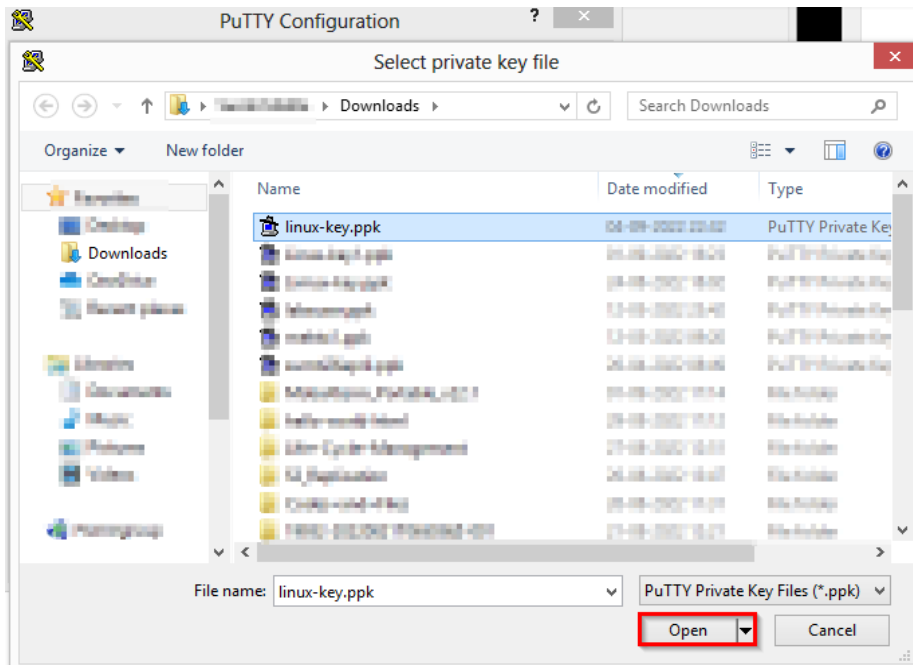
- Under it now Click on “+” and expand **Auth**.



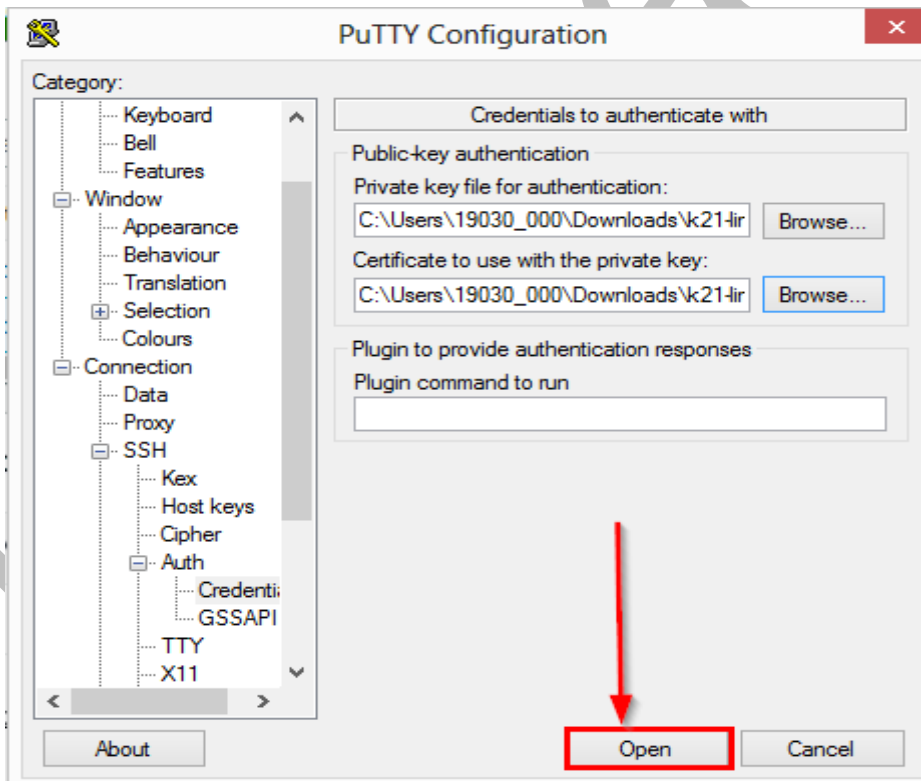
6. Now Click on **Credential** and Browse the **Private key file** in both the sections.



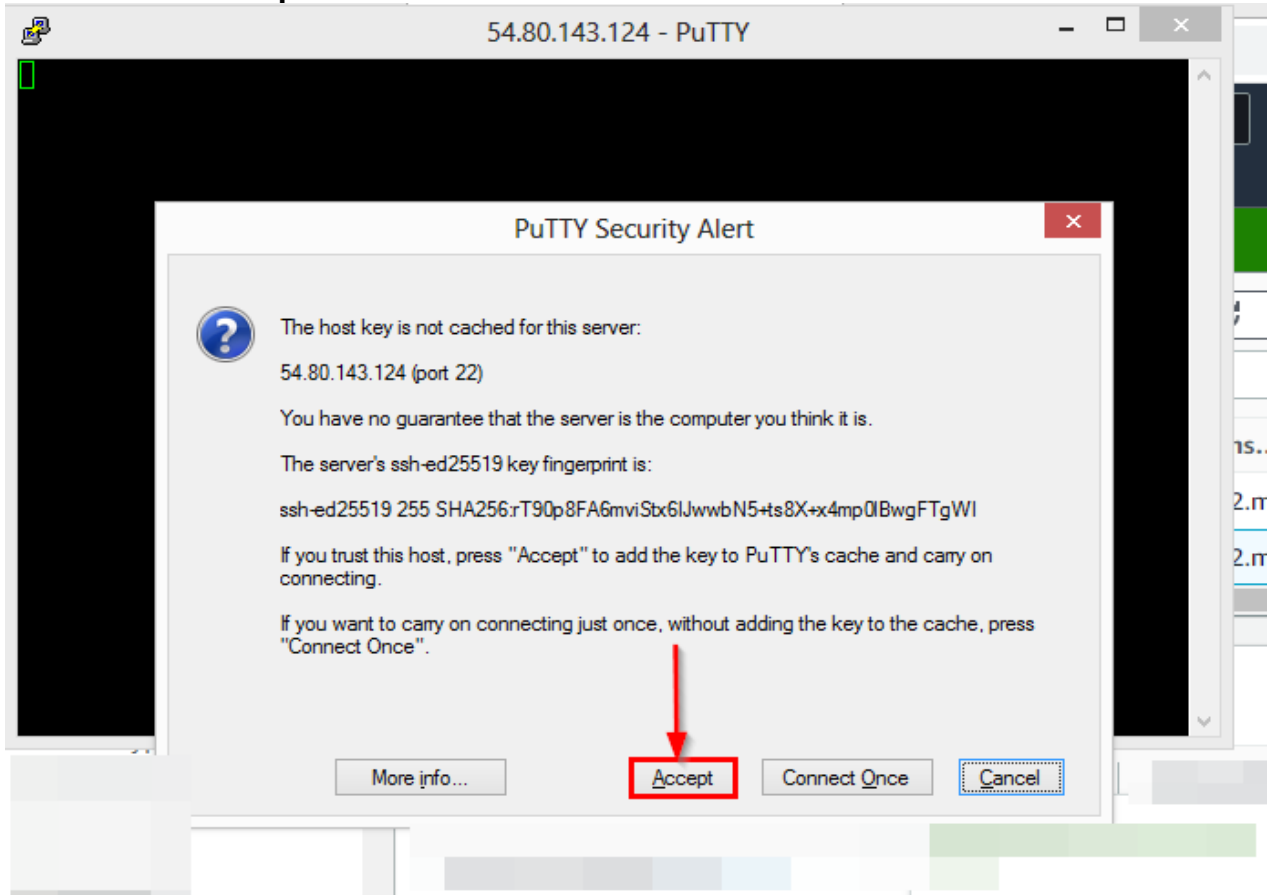
7. Now Select our **.ppk file** and Click on **Open**.



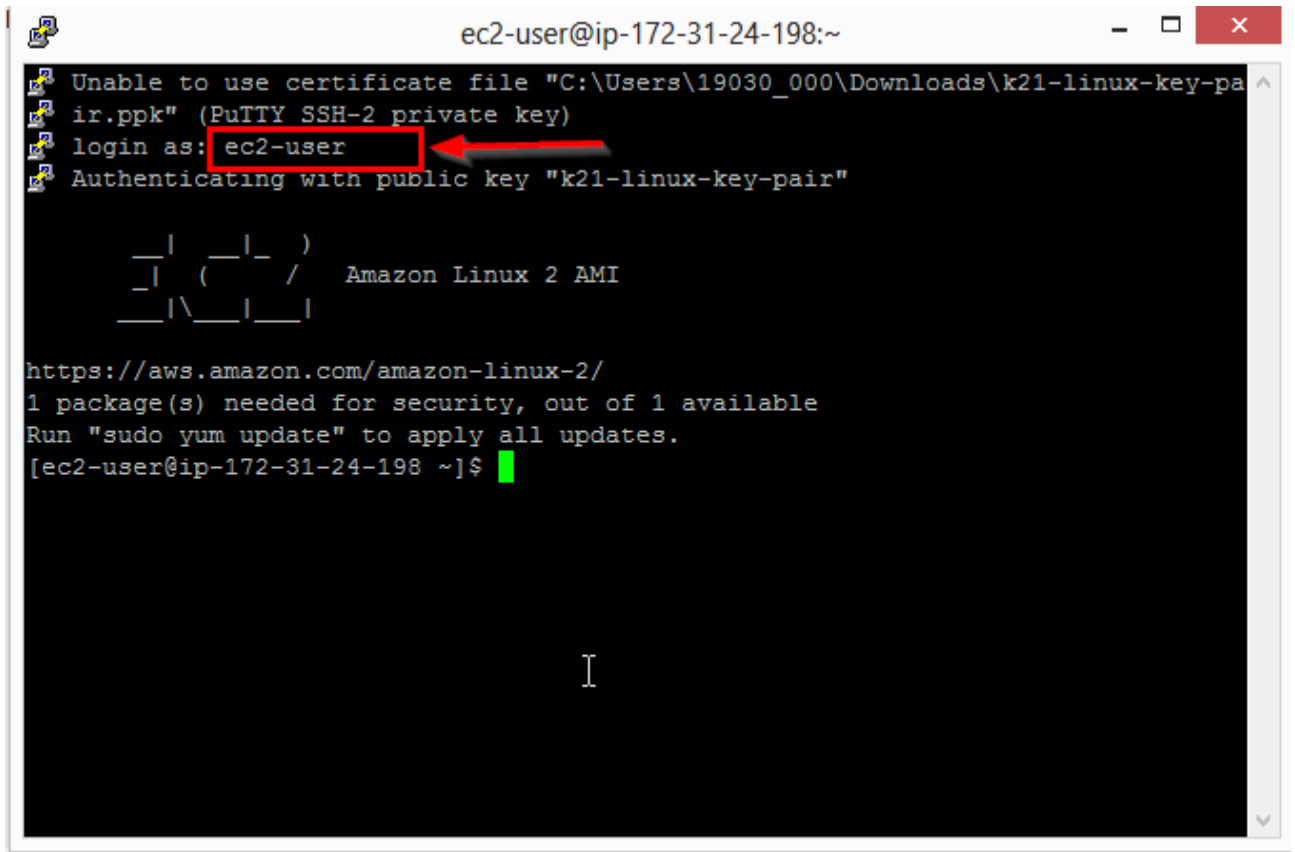
8. Now click n the **Open**



9. Now Click on **Accept**.



10. Now type **ec2-user** for login as and press Enter.



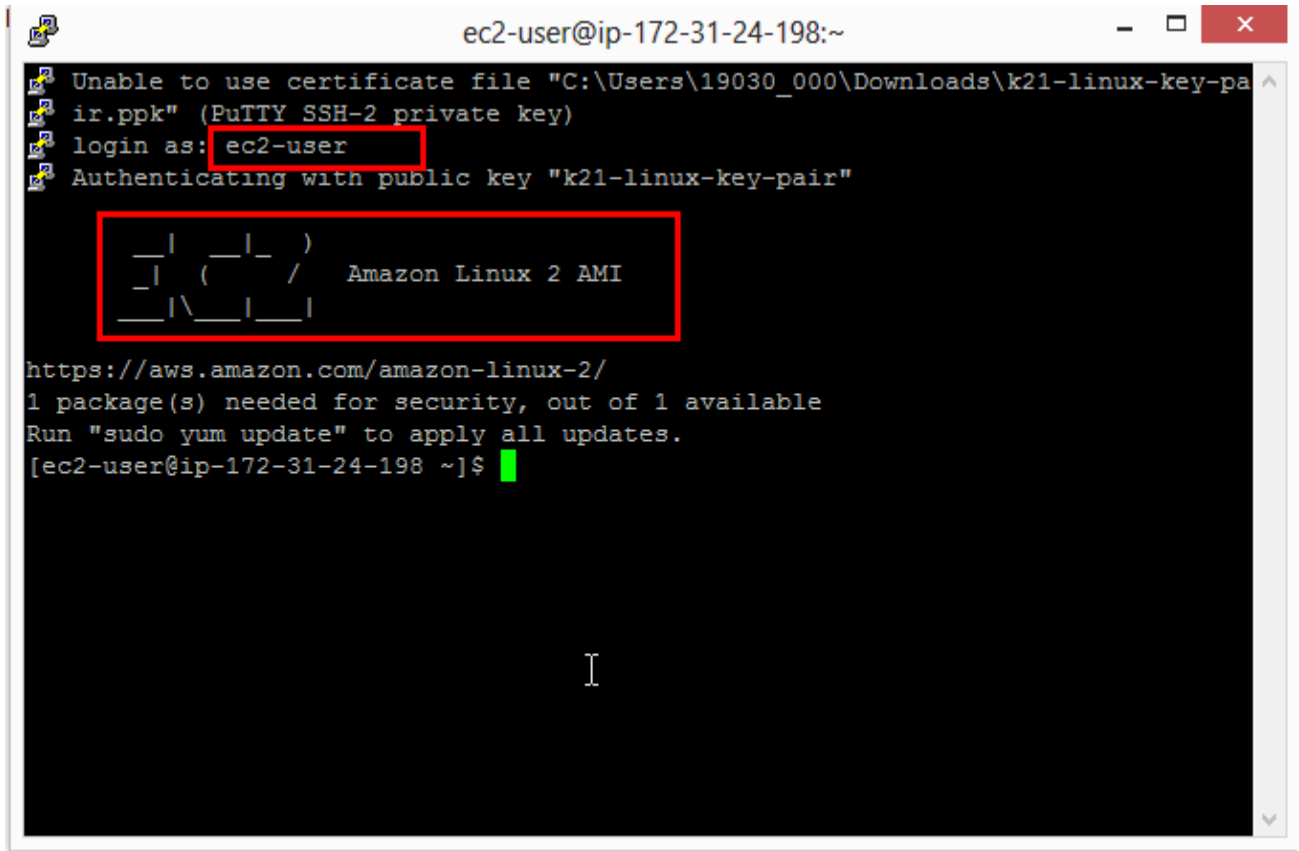
```
ec2-user@ip-172-31-24-198:~
Unable to use certificate file "C:\Users\19030_000\Downloads\k21-linux-key-pa
ir.ppk" (PuTTY SSH-2 private key)
login as: ec2-user
Authenticating with public key "k21-linux-key-pair"

  _ | _ | _ )
  _ | ( _ - /   Amazon Linux 2 AMI
  _ |\ _ | _ |

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 1 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-24-198 ~]$
```

Note: Don't forget to type **ec2-user** to login as it is the default user by which you login into a Linux EC2 instance.

11. Now you have successfully connected to your Linux instance using putty.



The screenshot shows a PuTTY terminal window titled "ec2-user@ip-172-31-24-198:~". The terminal output is as follows:

```
Unable to use certificate file "C:\Users\19030_000\Downloads\k21-linux-key-pa
ir.ppk" (PuTTY SSH-2 private key)
login as: ec2-user
Authenticating with public key "k21-linux-key-pair"

 _ | _ | _ )
 _ | ( _ | /   Amazon Linux 2 AMI
 _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 1 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-24-198 ~]$
```

The text "login as: ec2-user" and the ASCII art logo for Amazon Linux 2 AMI are highlighted with red boxes in the original image.

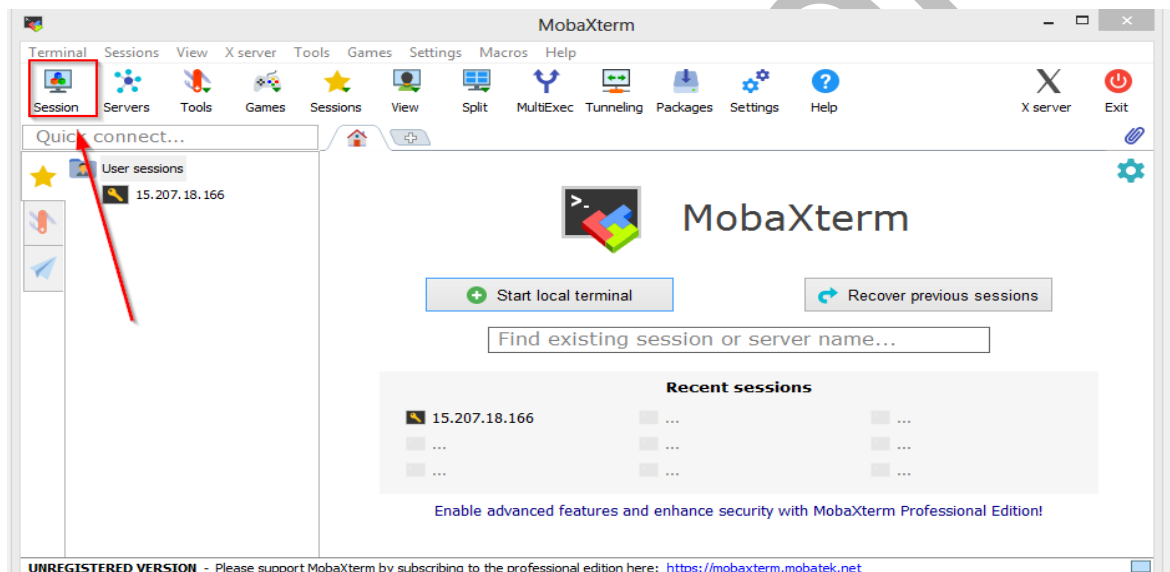
4.4 Accessing EC2 Instance Using MobaXterm

MobaXterm provides all the important remote network tools (SSH, RDP, X11, SFTP, FTP, Telnet, Rlogin) to Windows desktop, in a single portable exe file which works out of the box. Some plugins can be **used** to add functions to **MobaXterm** such as Unix commands (bash, ls, cat, sed, grep, awk, rsync).

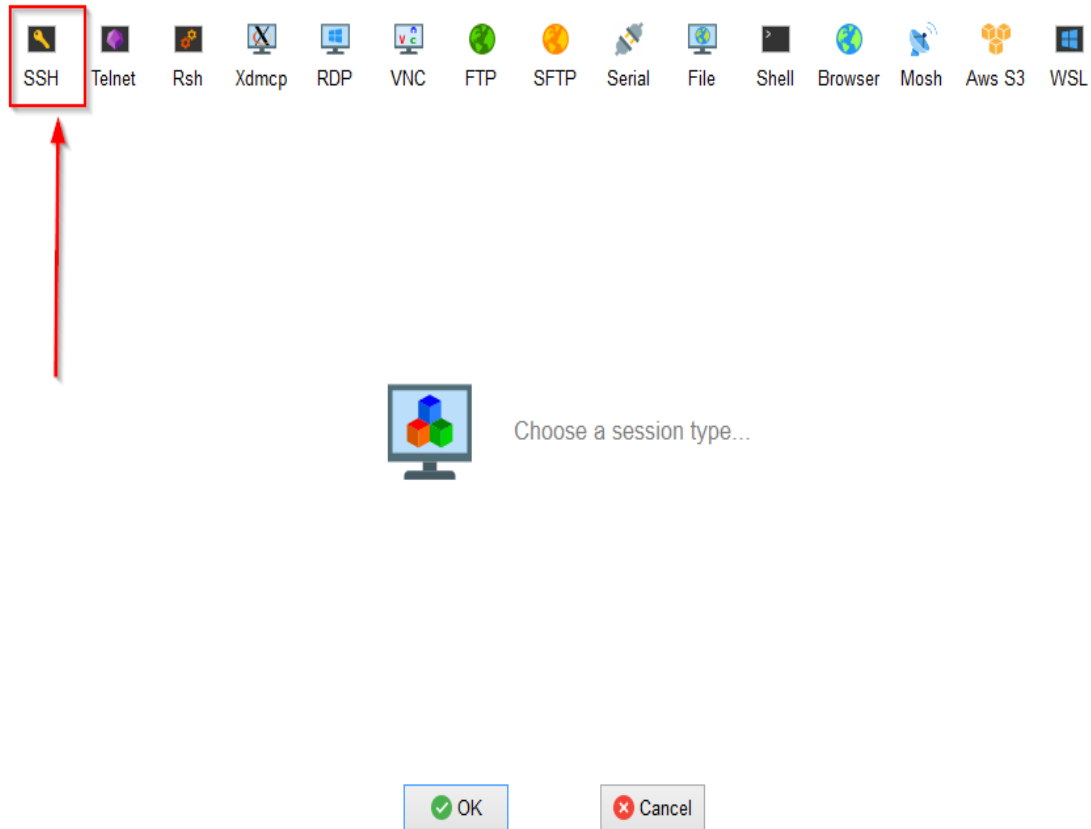
Download link for MobaXterm (Windows):

<https://mobaxterm.mobatek.net/download-home-edition.html>

1. Open **MobaXterm** and click on **Session**

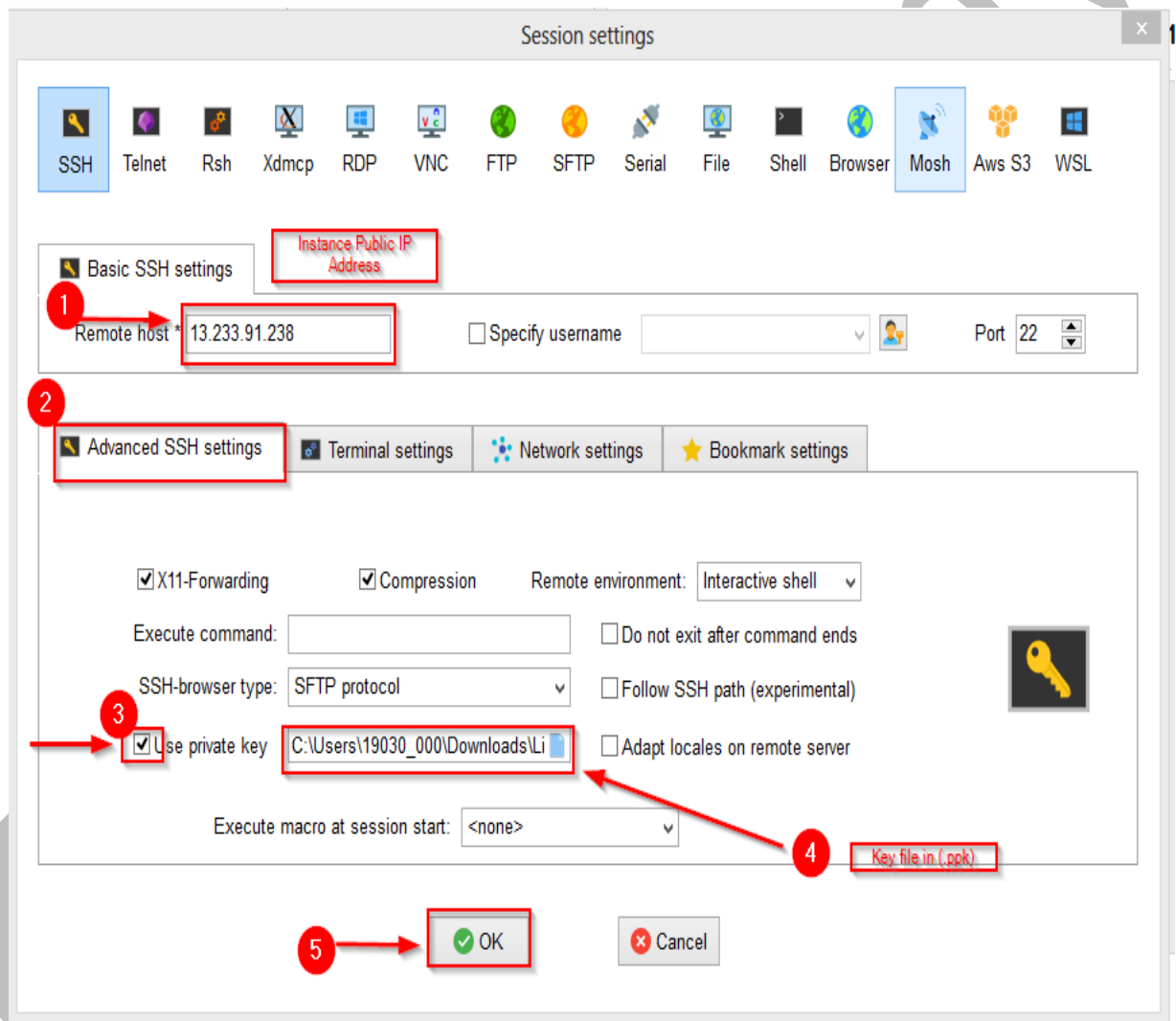


2. A dialog box will open and click on **SSH** at the top right corner.

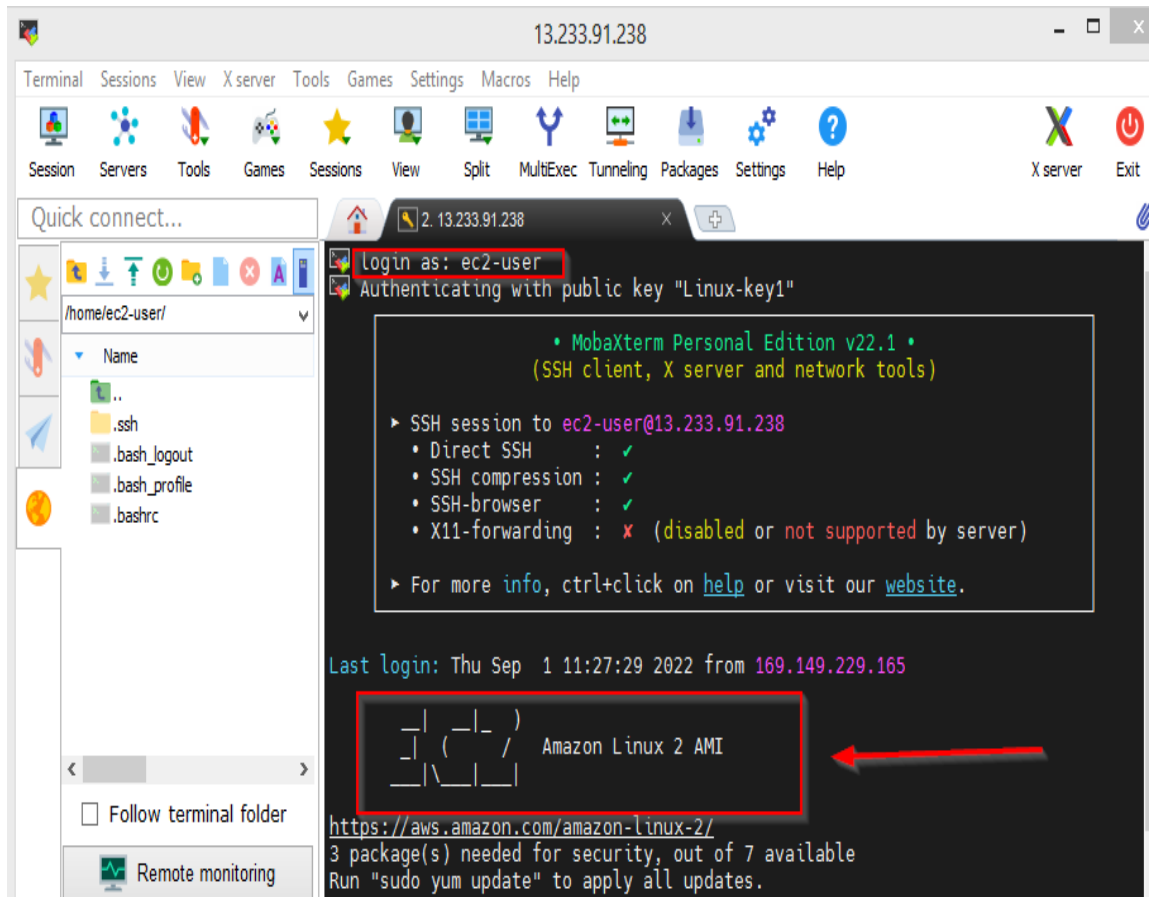


3. Now configure basic **SSH Settings**

- In the **Remote host** : Type or Paste the **Public IP** of Linux Instance
- Click on Advanced SSH settings
- Click on **Use private key** checkbox and provide your **private (ppk)**
- Finally click on **OK**



4. You will be logged in to the Linux Instance as shown below.



Thus we have successfully logged in to the Linux Instance using MobaXterm

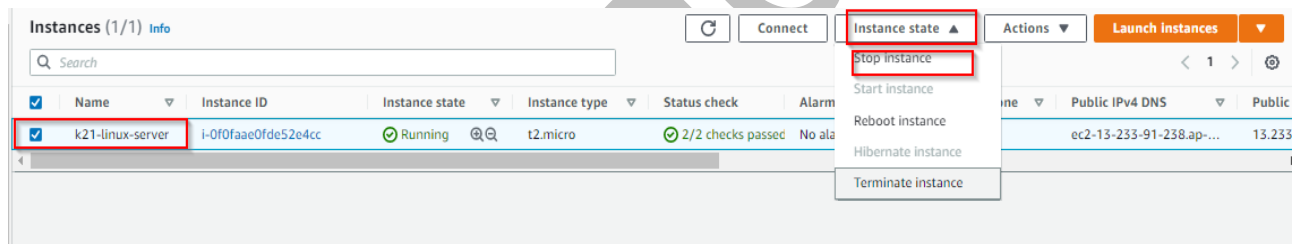
5 DELETING/CLEANUP

Note: In this section we will be deleting all the resources we have used for this lab in order to be in free tier limit, as keeping these resources running will be chargeable.

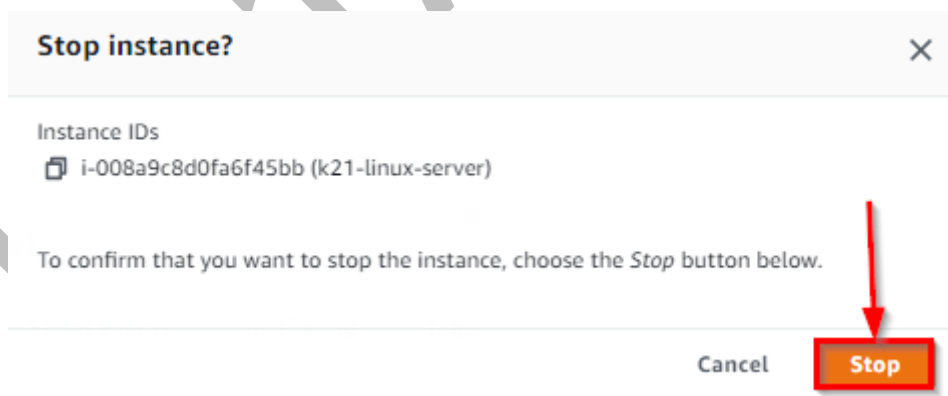
5.1 Stopping the Linux Instance

Note: We would always recommend to Terminate the instance if there is no requirement of it. But many times, it may happen that you shall require the instance after some time to continue with your work, then you shall stop the instance. Stopping the instance will not charge you but you shall be charged for the volume/storage attached to it. So, the charges shall be minimum as compared to running instance, but you will still get charged.

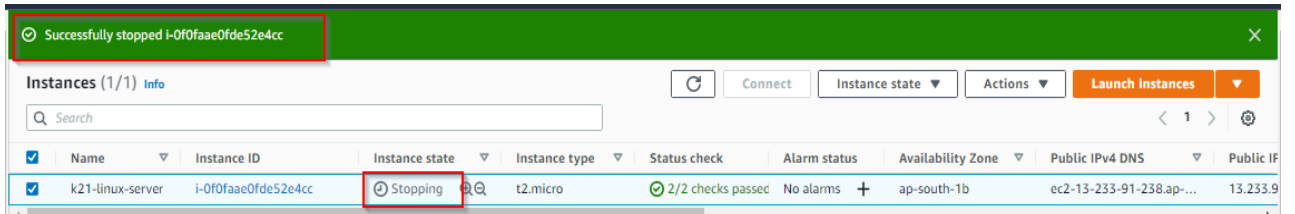
1. Back on EC2 Console, select our Instance, under **instance state** Click on **Stop instance**.



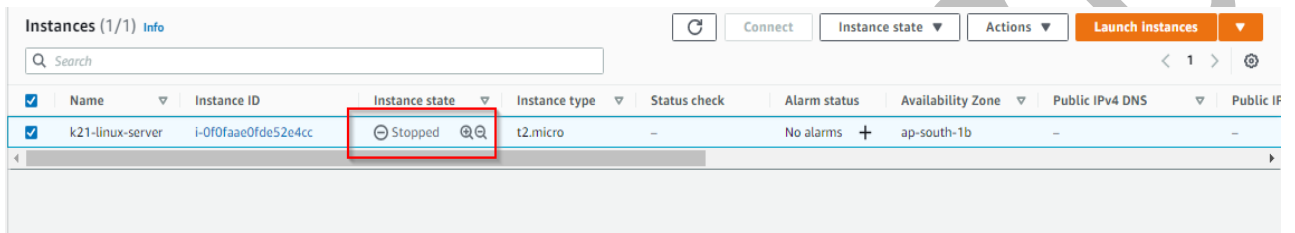
2. Click on **Stop** to **Confirm**.



- Now you shall see our instance is in stopping state, wait for some time.



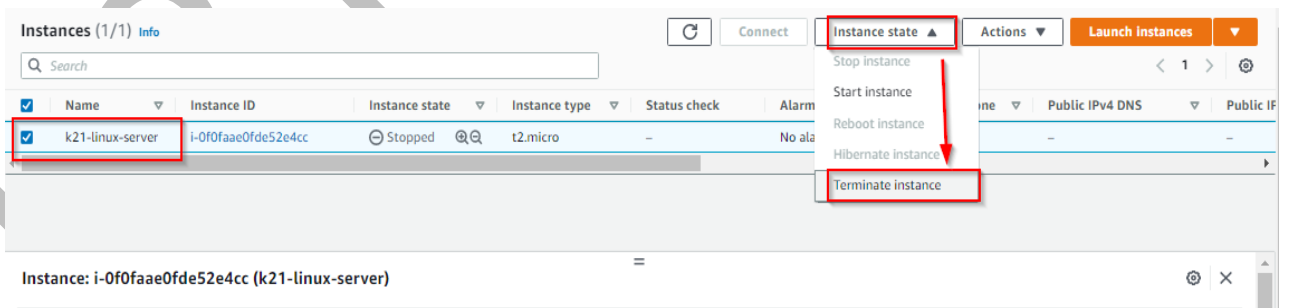
- Refresh and you shall see our instance have stopped.



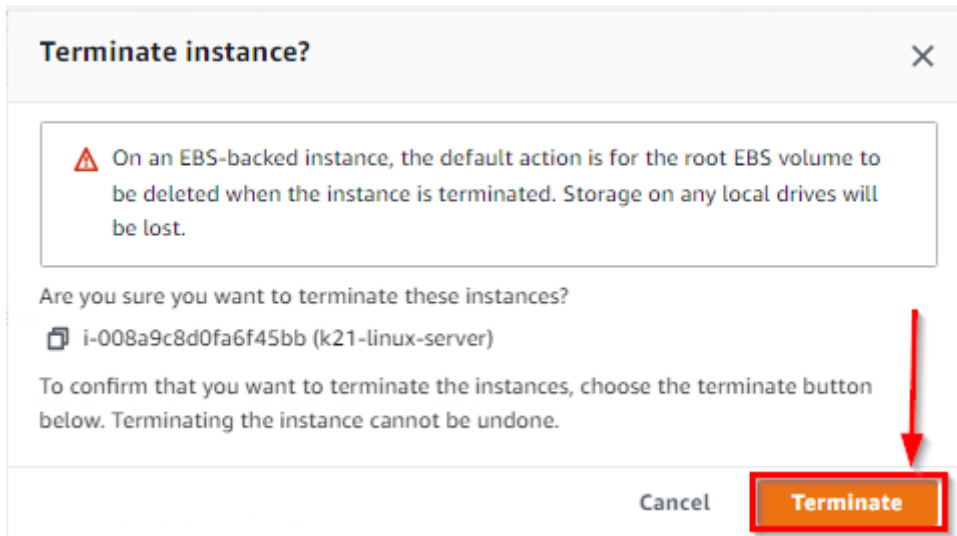
5.2 Terminating the Linux Instance

Note: In this section, we shall see the steps to Terminate the EC2 instance, as if the instances are not terminated once there is no requirement of it then you shall be charged for using those resources. So, to be in free tier and complete all the labs please terminate the resources here the EC2 instance when in no use.

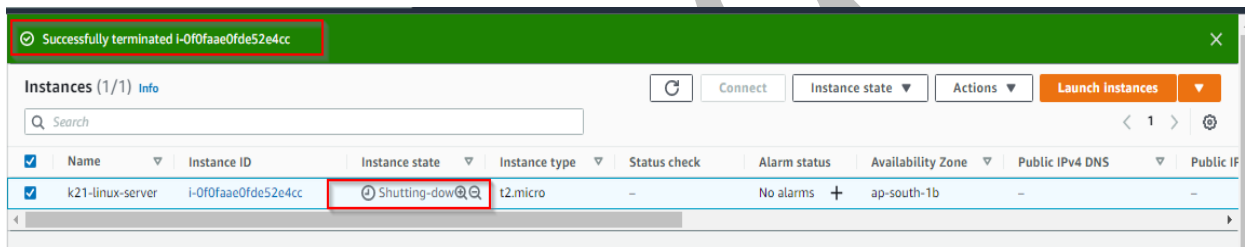
- Select our instance, under **Instance state** Click on **Terminate instance**.



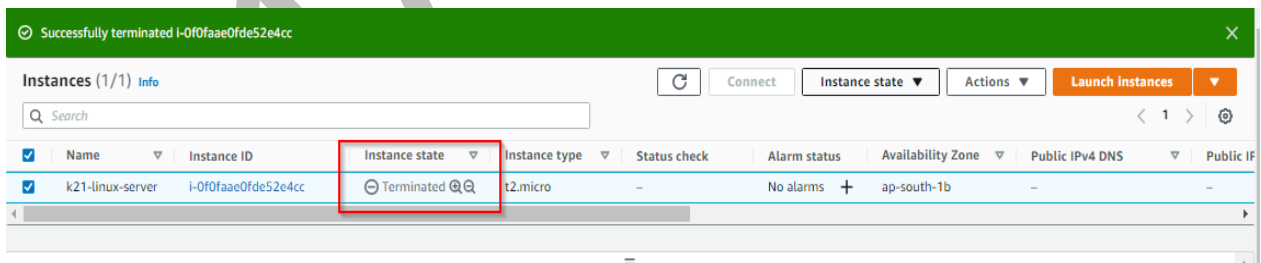
- Now Click on **Terminate** to Confirm.



- Your instance is in **shutting down** state, wait for some time.



- Refresh and you shall see our instance is **Terminated**.

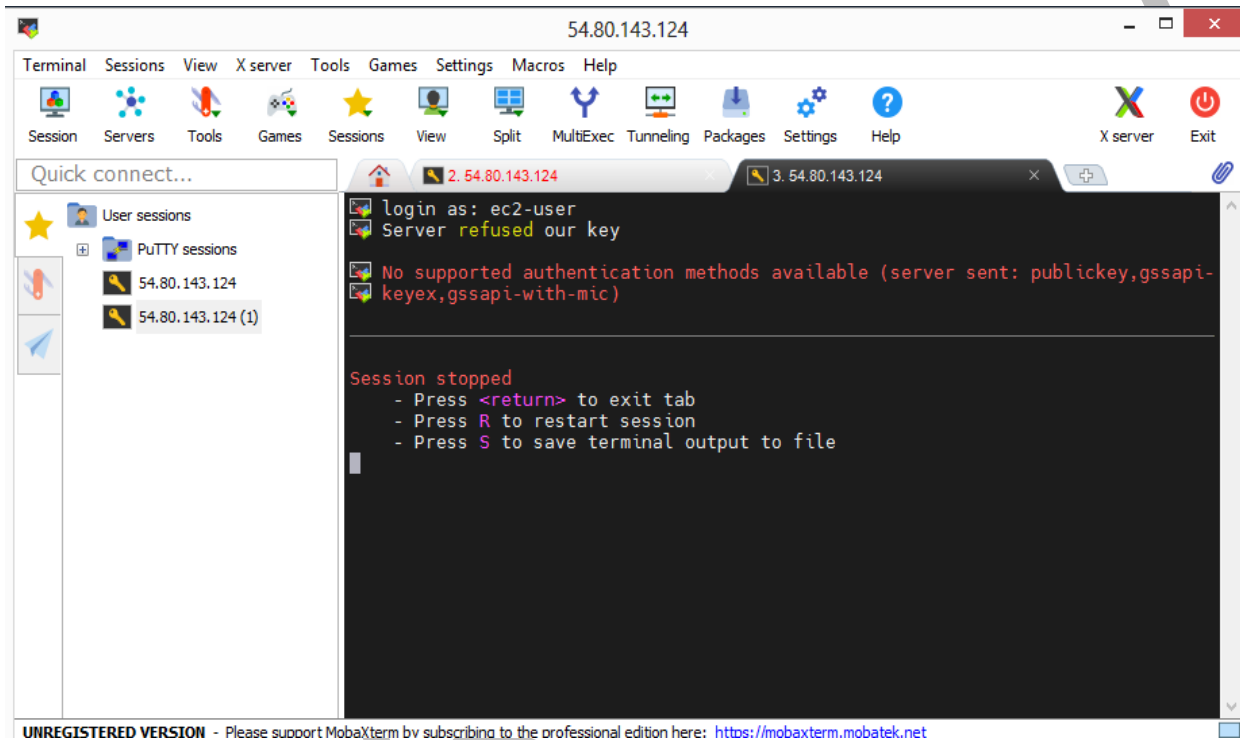


Thus, we have successfully Terminated the Linux EC2 instance.

6 TROUBLESHOOTING

Sometimes you can get this error message while using **MobaXterm**.

Error: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).



Fix: Check your key, if you are using the correct key file or not and initiate the SSH session again.

7 SUMMARY

This activity guide covered steps to:

1. Launching a Linux Instance Using EC2
 - a. Creating an EC2 Instance
 - b. Accessing EC2 Instance Using Browser SSH Connection
 - c. Accessing EC2 Instance Using Putty
2. Deleting/Cleanup
 - a. Stopping the Linux Instance
 - b. Terminating the Linux Instance