

Demo: Launch and Connect to EC2 Ubuntu Instance

Goto EC2 Dashboard and in the left pane click on Instances.

Resources EC2 Global view Refresh Settings

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	0	Auto Scaling Groups	0
Dedicated Hosts	0	Elastic IPs	2
Instances	0	Key pairs	12
Load balancers	0	Placement groups	0
Security groups	12	Snapshots	2
Volumes	0		

Account attributes Refresh

Supported platforms External link

- VPC

Default VPC External link
vpc-018e1dad574748a56

Settings
EBS encryption
Zones
EC2 Serial Console
Default credit specification
Console experiments

Explore AWS Close

Notification: Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. [Learn more](#) Close

Now provide a name to your instance.

[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

ec2-assignment1 [Add additional tags](#)

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

Select the Ubuntu AMI.

▼ 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

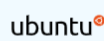
Amazon
Linux



macOS



Ubuntu



Windows



Red Hat



SUSE



[Browse more AMIs](#)

Including AMIs from
AWS, Marketplace and
the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

ami-007855ac798b5175e (64-bit (x86)) / ami-0c6c29c5125214c77 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Choose Instance Type as **t2.micro** as it is the instance type which comes under **Free Tier**.

Now, select one of the existing key pairs, as it will allow us to connect to our instance using an ssh client such as PuTTY later.

▼ Instance type [Info](#)

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory

On-Demand Windows pricing: 0.0162 USD per Hour

On-Demand SUSE pricing: 0.0116 USD per Hour

On-Demand RHEL pricing: 0.0716 USD per Hour

On-Demand Linux pricing: 0.0116 USD per Hour

[Compare instance types](#)

▼ 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*

vishwas-nv



[Create new key pair](#)

In case you are not having any existing key pairs, you can also create a new key pair by clicking on **Create new key pair** option.

Provide a name and create a key with **RSA** and **.pem** extension.

Create key pair [X]

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

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](#)

Once done, now in network settings, create a new security group with ssh and http protocols allowed.

Note: You can also use existing security groups.

The screenshot shows the 'Network settings' page of the AWS EC2 instance configuration wizard. The page is divided into two main sections: 'Network settings' and 'Summary'.

Network settings:

- Network:** vpc-018e1dad574748a56 | Default
- Subnet:** No preference (Default subnet in any availability zone)
- Auto-assign public IP:** Enable
- Firewall (security groups):** 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-35' 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.
- A warning message: Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Summary:

- Number of instances:** 1
- Software Image (AMI):** Canonical, Ubuntu, 22.04 LTS, ...read more (ami-007855ac798b5175e)
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** New security group
- Storage (volumes):** 1 volume(s) - 8 GiB
- A warning message: Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is available).
- Buttons:** Cancel, Launch instance, Review commands

Configure the storage, 8gb storage is sufficient for our purpose.

Now, click on **Launch Instance**.

The screenshot shows the 'Configure storage' page of the AWS EC2 instance configuration wizard. The page is divided into two main sections: 'Configure storage' and 'Summary'.

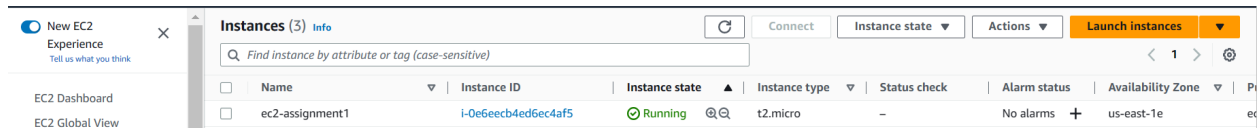
Configure storage:

- Storage configuration:** 1x 8 GiB gp2 Root volume (Not encrypted)
- A warning message: Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage.
- Buttons:** Add new volume
- Text:** The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance
- File systems:** 0 x File systems
- Buttons:** Edit
- Advanced details:** Info

Summary:

- Software image (AMI):** Canonical, Ubuntu, 22.04 LTS, ...read more (ami-007855ac798b5175e)
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** New security group
- Storage (volumes):** 1 volume(s) - 8 GiB
- A warning message: Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is available).
- Buttons:** Cancel, Launch instance, Review commands

After launching it successfully, you can go back to instances.
After waiting for a few seconds, the instance will be in running state and we can login to it.



To login, now open Puttygen in your system.

You can download Putty **64-bit x86: putty-64bit-0.78-installer.msi** from here:
<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 (except the n

Bug: this installer was built differently to other versions, in a way that causes trouble for upgr trouble, when moving between 0.78 and other versions, we recommend completely uninstalling this installer) by installing 0.78 with a special command-line invocation like:

```
msiexec.exe /i path\to\putty-64bit-0.78-installer.msi ALLUSERS=1
```

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

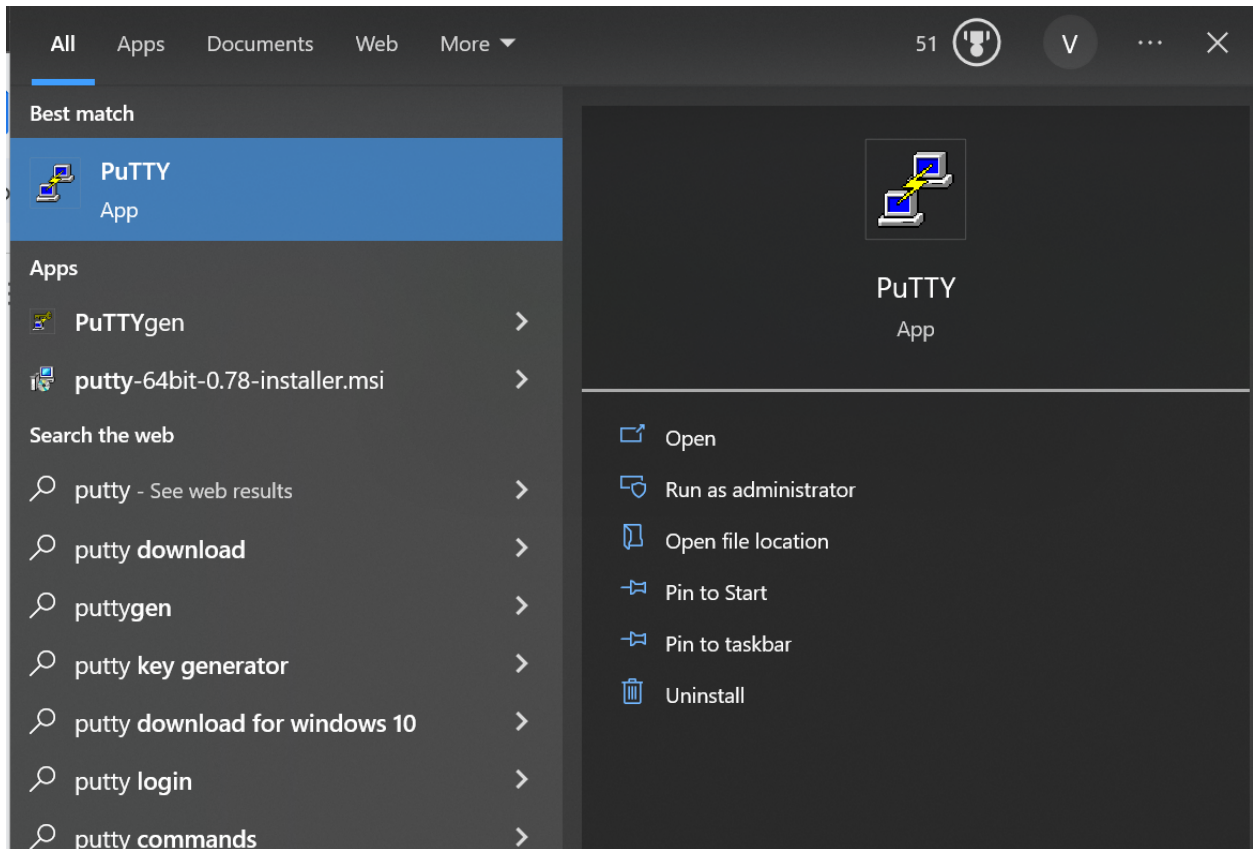
We also publish the latest PuTTY installers for all Windows architectures as a free-of-charge d we release them.

MSI ('Windows Installer')

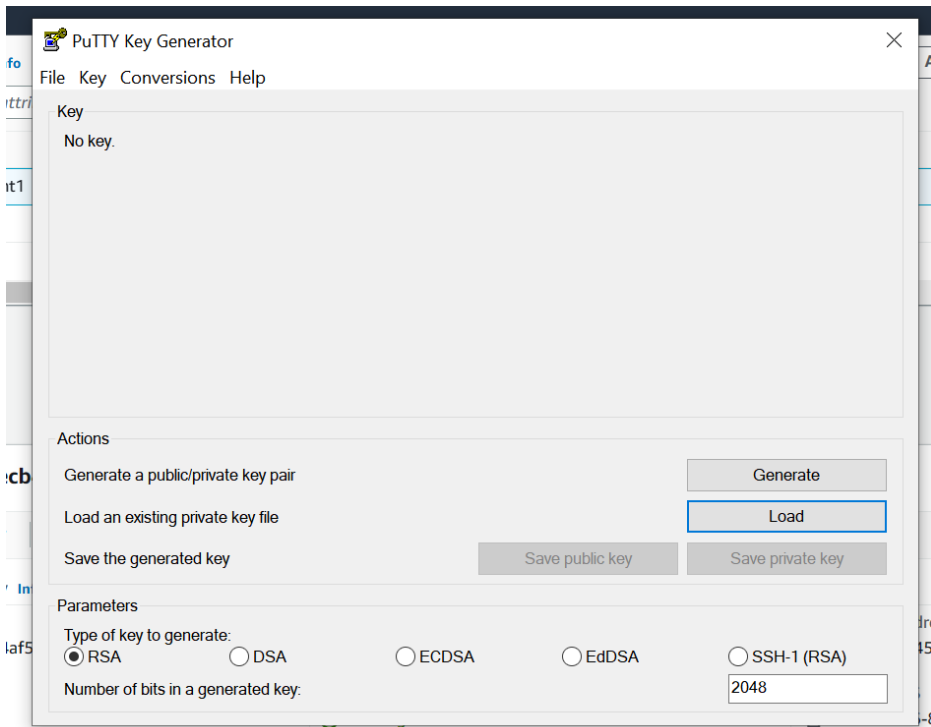
64-bit x86: [putty-64bit-0.78-installer.msi](#) ([signature](#))

After Downloading, you need to install it by all default configurations.

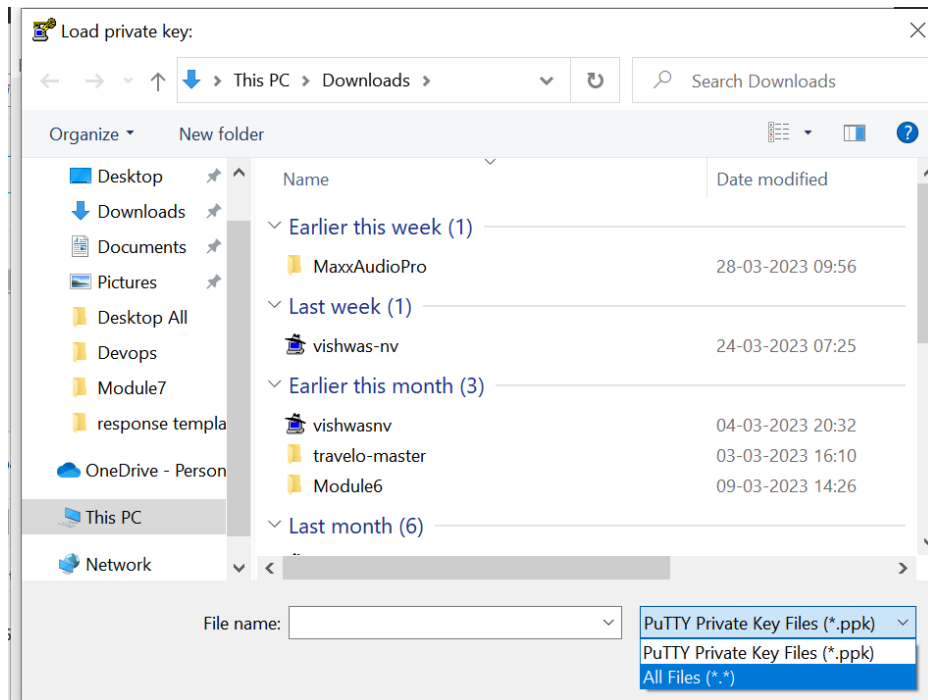
After Installation is done, just search putty in your windows search bar.



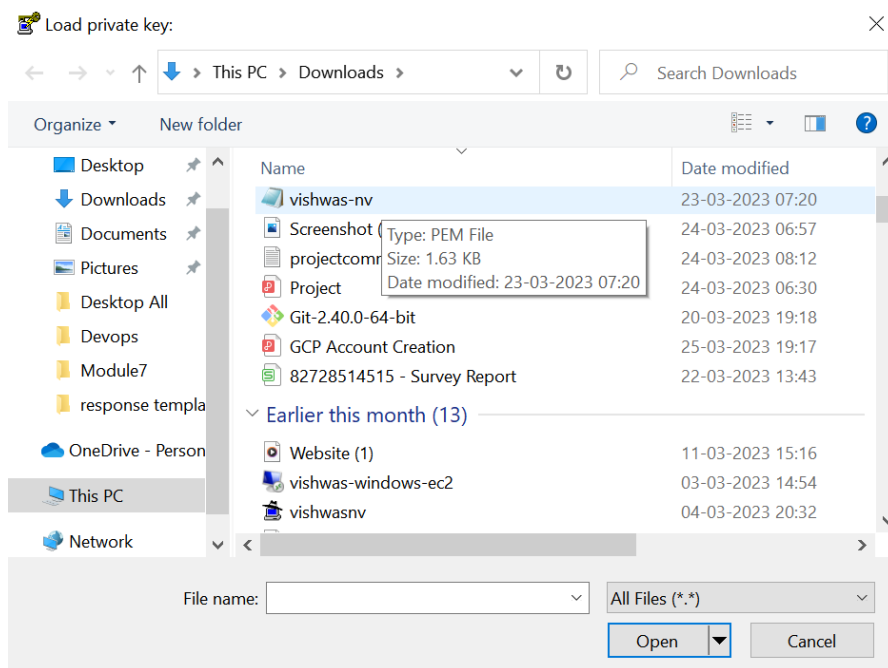
Open Puttygen.



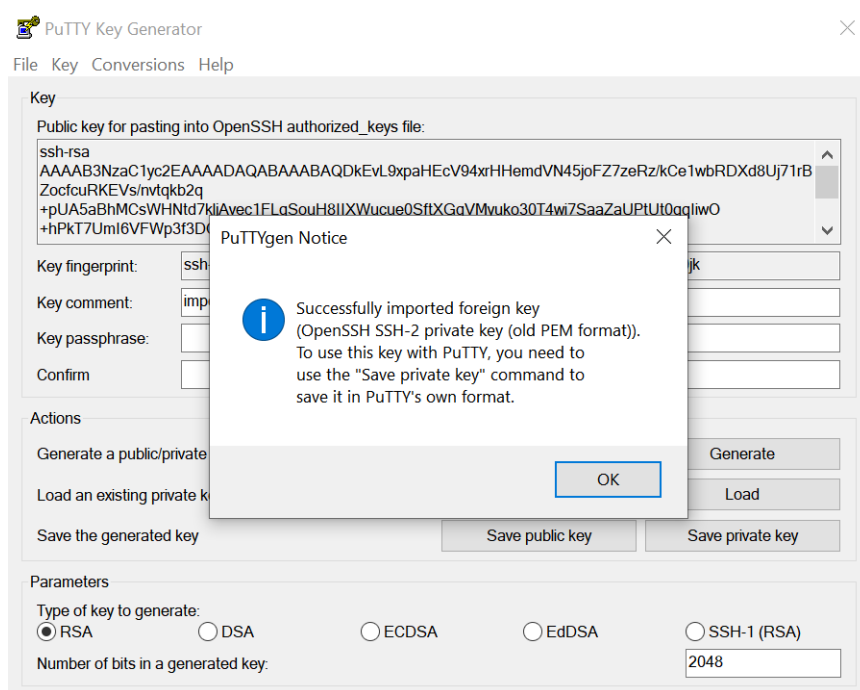
Click on Load:



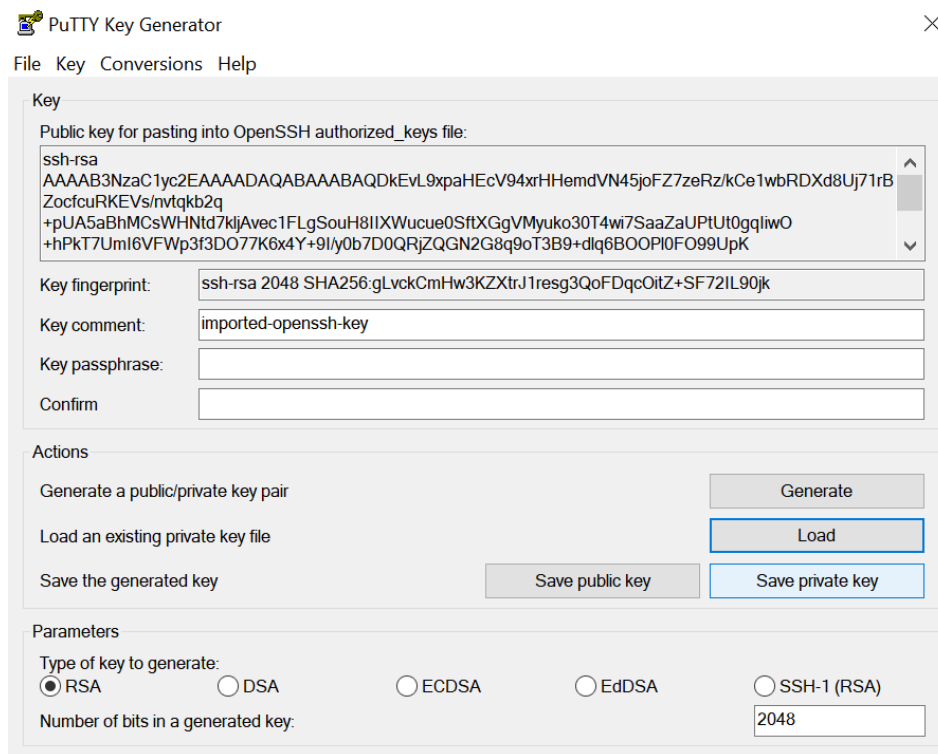
Select your .pem key

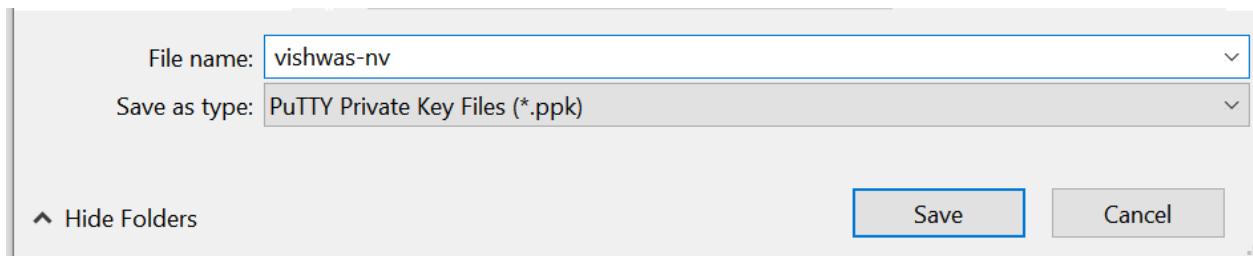
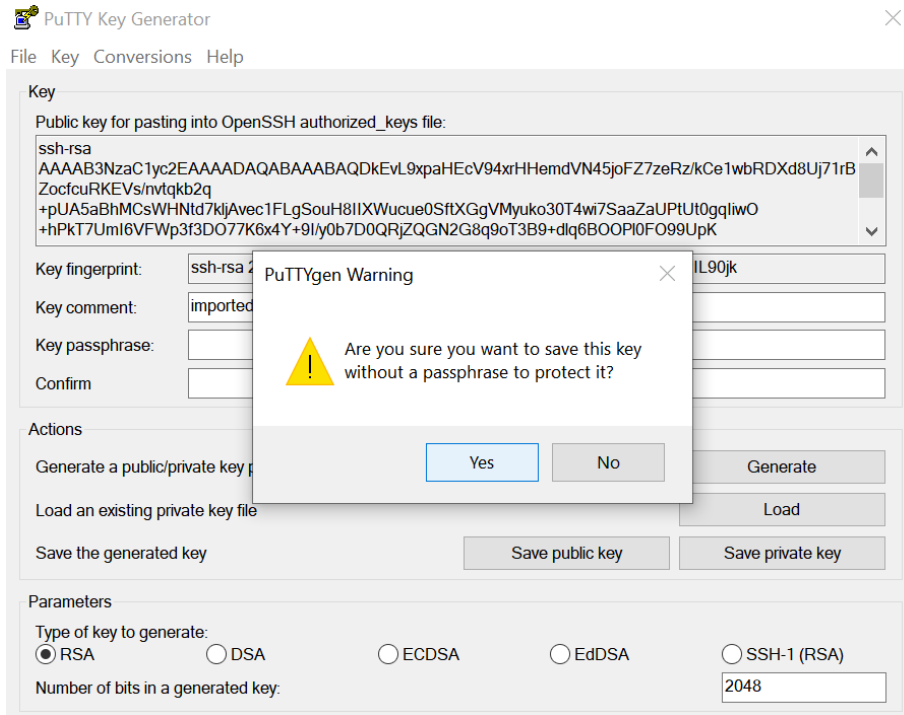


Key is successfully imported.

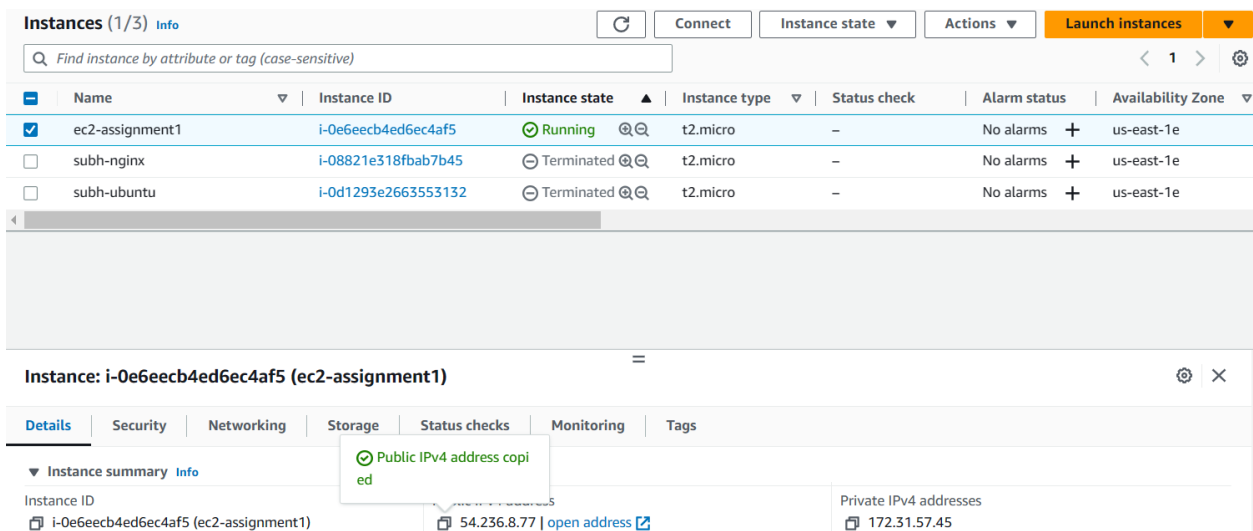


Now, click on save private key.

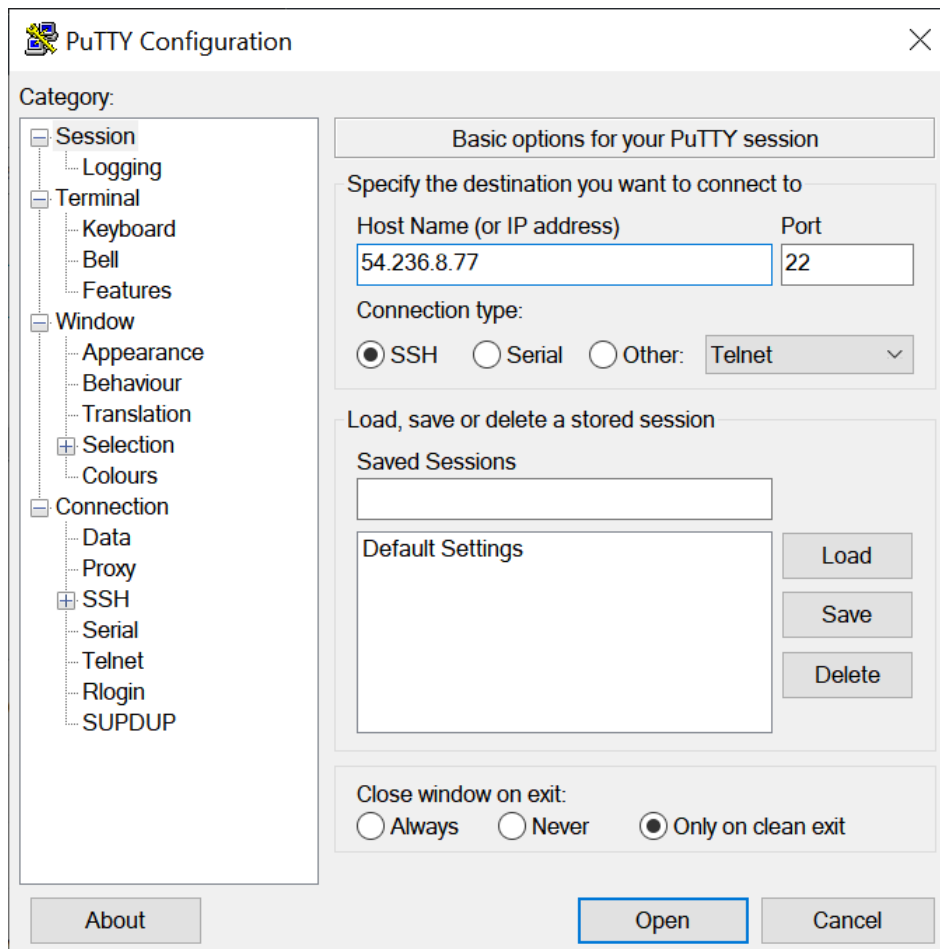




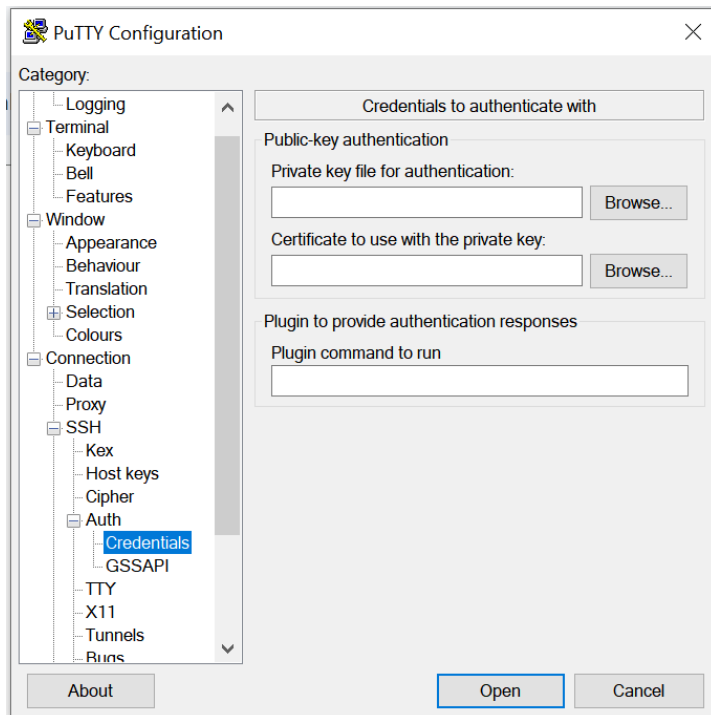
Once saved, go back to Ec2 console and copy public ip of your instance.



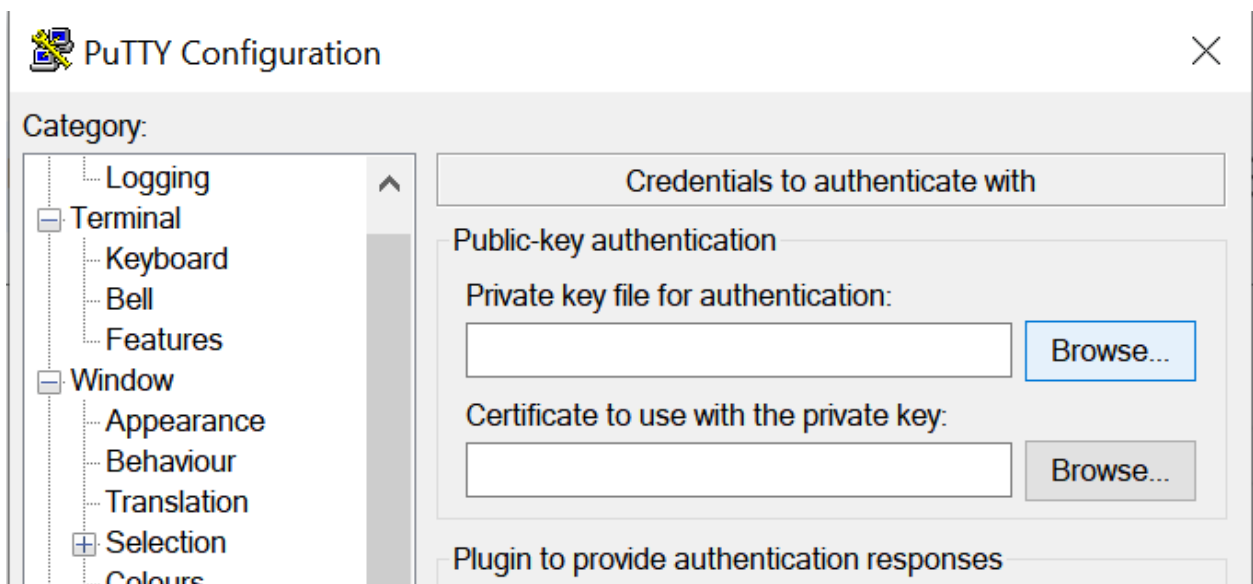
Paste the public ip in host name in Putty.

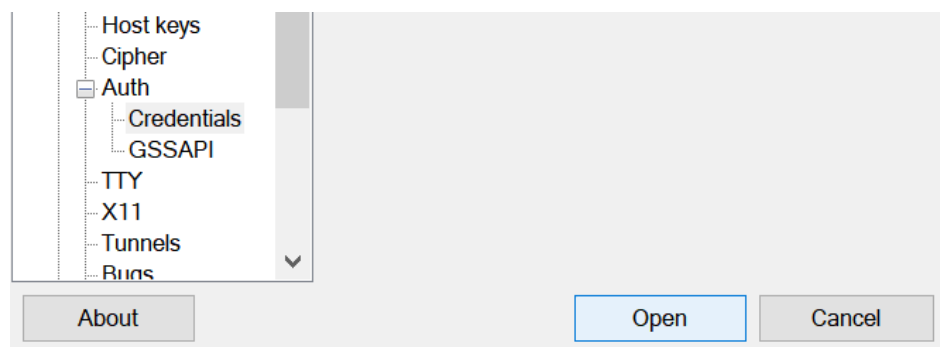
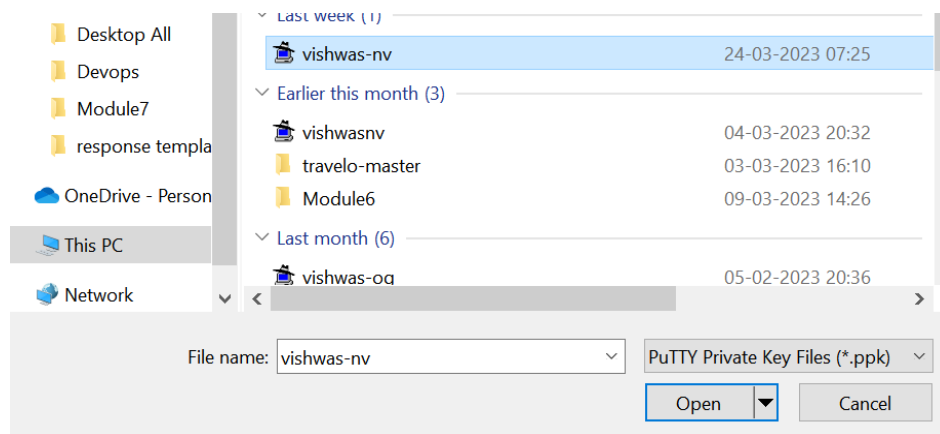


Now in left pane, go to ssh >> Auth >> Credentials:

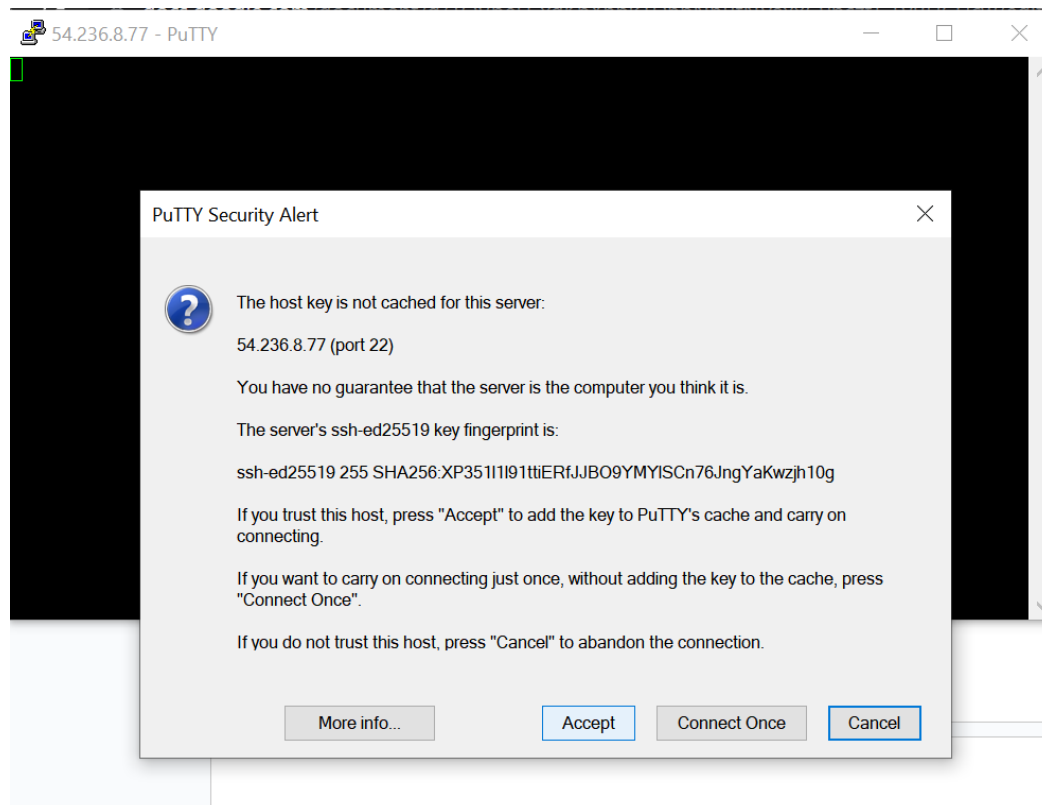


Click on Browse, select the ppk key and click on open.

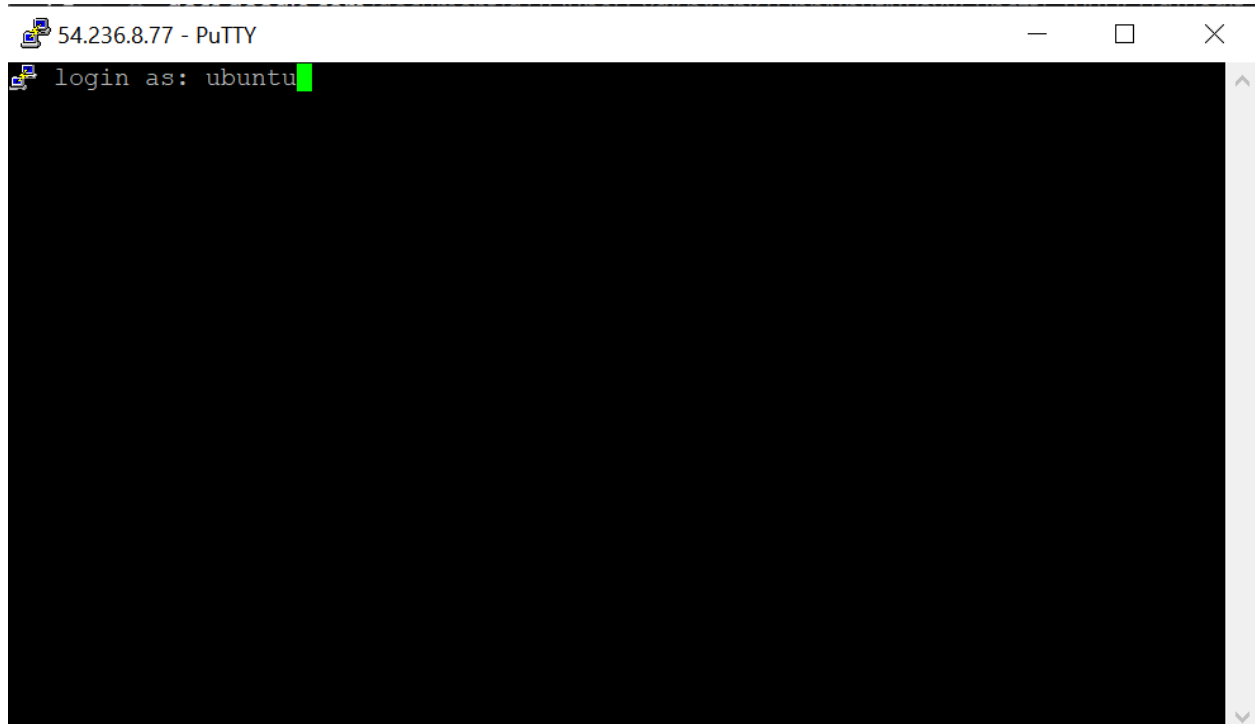




Click on Accept.

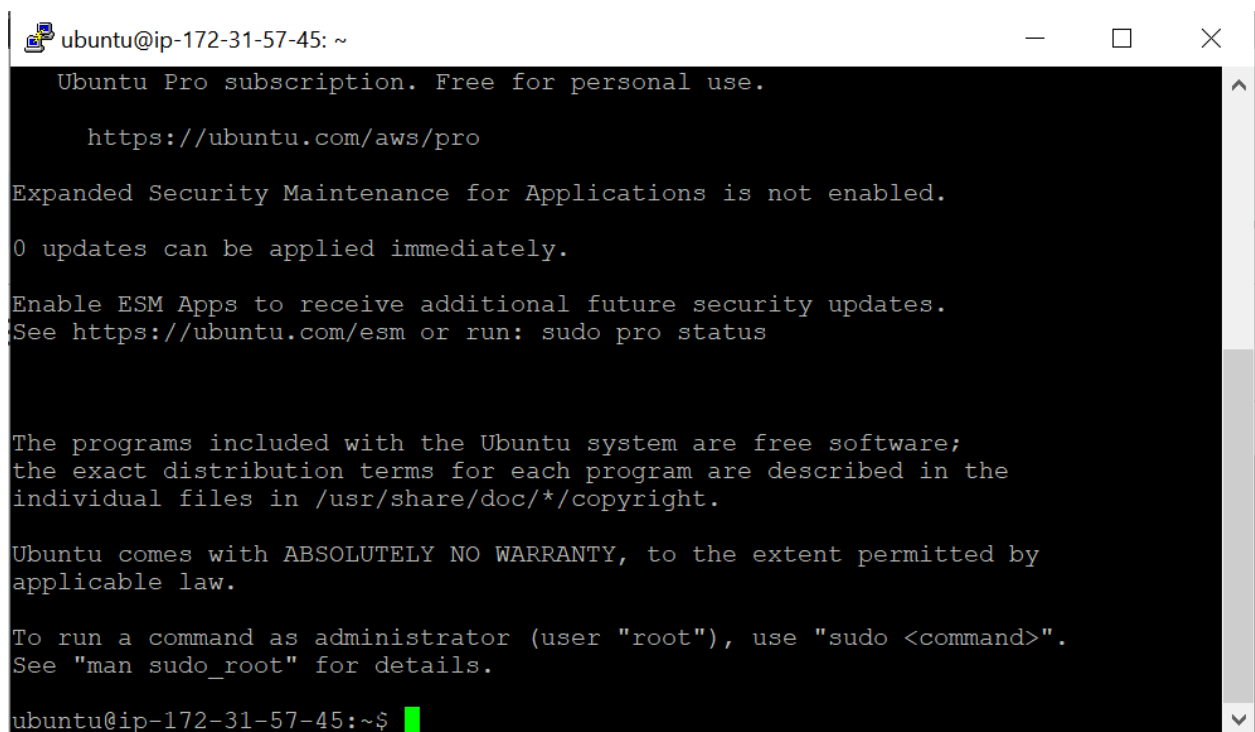


Provide the username as **ubuntu** as it is an ubuntu instance.
For Amazon linux and Redhat instances this username will be **ec2-user**.



A screenshot of a PuTTY terminal window. The title bar reads "54.236.8.77 - PuTTY". The terminal content shows the login prompt "login as: ubuntu" followed by a green cursor. The rest of the terminal is black and empty.

You are now successfully connected.



A screenshot of a terminal window. The title bar reads "ubuntu@ip-172-31-57-45: ~". The terminal content displays the following text:
Ubuntu Pro subscription. Free for personal use.
<https://ubuntu.com/aws/pro>
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: `sudo pro status`

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.

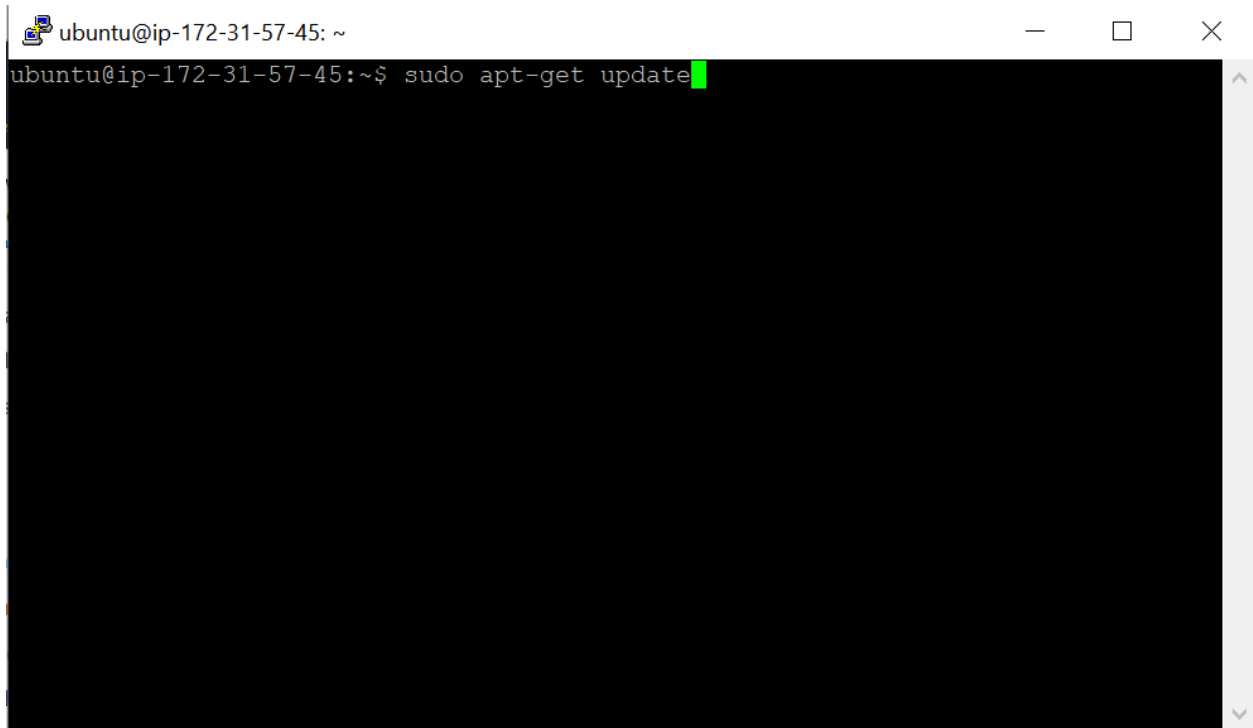
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
ubuntu@ip-172-31-57-45:~\$

Now run the following command as shown in pictures below to update the system and install nginx webserver on it.

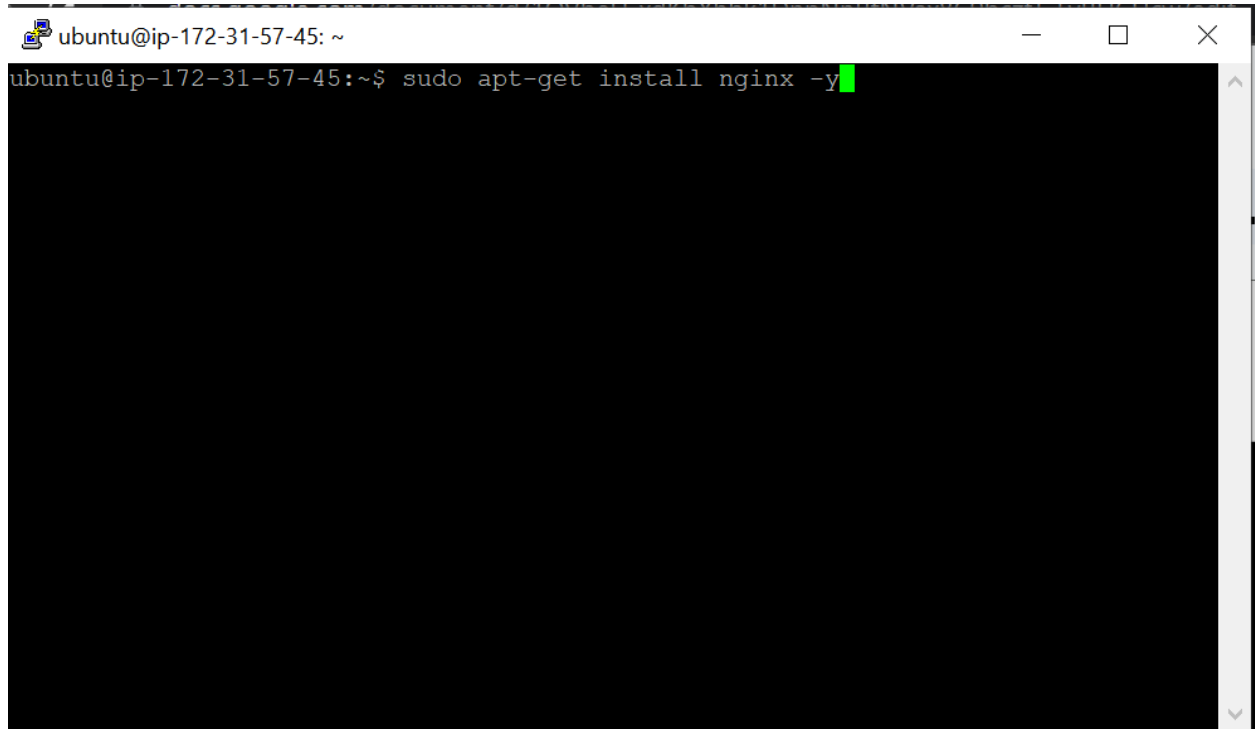
Commands:

```
sudo apt-get update
```

```
sudo apt-get install nginx -y
```

A terminal window with a black background and white text. The title bar at the top shows 'ubuntu@ip-172-31-57-45: ~' and standard window control buttons. The terminal content shows the prompt 'ubuntu@ip-172-31-57-45:~\$' followed by the command 'sudo apt-get update'. A green cursor is positioned at the end of the command. The rest of the terminal area is empty.

```
ubuntu@ip-172-31-57-45: ~  
ubuntu@ip-172-31-57-45:~$ sudo apt-get update
```

A terminal window with a title bar showing 'ubuntu@ip-172-31-57-45: ~'. The command 'sudo apt-get install nginx -y' has been entered, and a green cursor is visible at the end of the line. The rest of the terminal is black.

```
ubuntu@ip-172-31-57-45: ~  
ubuntu@ip-172-31-57-45:~$ sudo apt-get install nginx -y
```

Now paste the public ip on the browser, you will be able to see the web server running.



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

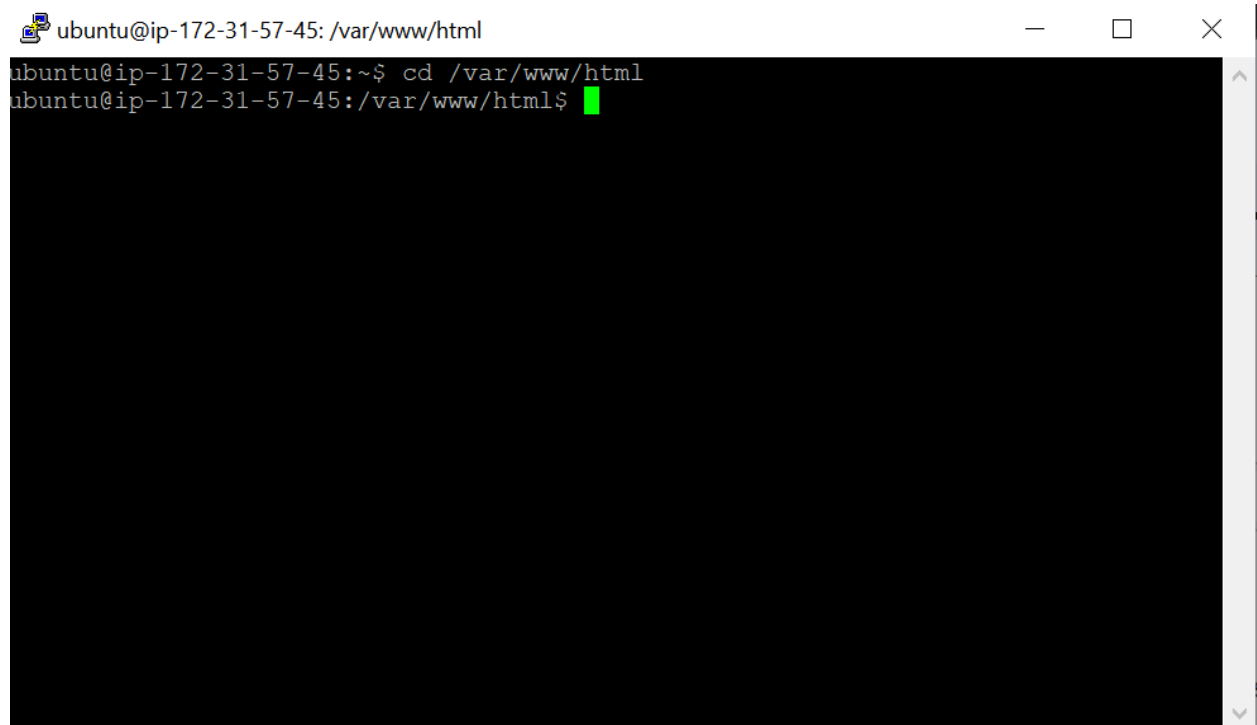
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Change the default website with a hello world page

Use the following command in sequence to change the default page to hello world page.

```
cd /var/www/html  
ls  
sudo su  
sudo rm index.-nginx-debian.html  
echo "hello world">>index.html
```



A terminal window titled 'ubuntu@ip-172-31-57-45: /var/www/html' with standard window controls (minimize, maximize, close). The terminal shows the following commands and output:

```
ubuntu@ip-172-31-57-45:~$ cd /var/www/html  
ubuntu@ip-172-31-57-45:/var/www/html$
```

The prompt is followed by a green cursor. The terminal has a scrollbar on the right side.


```
ubuntu@ip-172-31-57-45: /var/www/html
ubuntu@ip-172-31-57-45:~$ cd /var/www/html
ubuntu@ip-172-31-57-45:/var/www/html$ ls
index.nginx-debian.html
ubuntu@ip-172-31-57-45:/var/www/html$
```

```
root@ip-172-31-57-45: /var/www/html
ubuntu@ip-172-31-57-45:~$ cd /var/www/html
ubuntu@ip-172-31-57-45:/var/www/html$ ls
index.nginx-debian.html
ubuntu@ip-172-31-57-45:/var/www/html$ sudo su
root@ip-172-31-57-45:/var/www/html#
```

```
root@ip-172-31-57-45: /var/www/html
ubuntu@ip-172-31-57-45:~$ cd /var/www/html
ubuntu@ip-172-31-57-45:/var/www/html$ ls
index.nginx-debian.html
ubuntu@ip-172-31-57-45:/var/www/html$ sudo su
root@ip-172-31-57-45:/var/www/html# rm index.nginx-debian.html
root@ip-172-31-57-45:/var/www/html#
root@ip-172-31-57-45:/var/www/html# echo "hello world">index.html
root@ip-172-31-57-45:/var/www/html#
```

```
root@ip-172-31-57-45: /var/www/html
ubuntu@ip-172-31-57-45:~$ cd /var/www/html
ubuntu@ip-172-31-57-45:/var/www/html$ ls
index.nginx-debian.html
ubuntu@ip-172-31-57-45:/var/www/html$ sudo su
root@ip-172-31-57-45:/var/www/html# rm index.nginx-debian.html
root@ip-172-31-57-45:/var/www/html#
root@ip-172-31-57-45:/var/www/html# echo "hello world">index.html
root@ip-172-31-57-45:/var/www/html# ls
index.html
root@ip-172-31-57-45:/var/www/html#
```

Now, refresh the webpage, you will be able to see the hello world page.



hello world