Experiment-1

Date:

Aim:

To create a EC2 instance with Linux Operating System and explore its features.

Software Used:

Amazon Web Service Management Console

Background Information:

Amazon Elastic Compute Cloud (Amazon EC2) provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 reduces hardware costs to help develop and deploy applications faster. Amazon EC2 can be used to launch as many or as few virtual servers as required, configure security and networking, and manage storage. The user can add capacity (scale up) to handle compute-heavy tasks, such as monthly or yearly processes, or spikes in website traffic. When usage decreases, the user can reduce capacity (scale down) again providing elasticity.

The architecture of the EC2 is shown in the figure below. The Amazon EC2 instance deployed within an Amazon Virtual Private Cloud (VPC). The EC2 instance is within an Availability Zone in the Region. The EC2 instance is secured with a security group, which is a virtual firewall that controls incoming and outgoing traffic. A private key is stored on the local computer and a public key is stored on the instance. Both keys are specified as a key pair to prove the identity of the user. In this scenario, the instance is backed by an Amazon EBS volume. The VPC communicates with the internet using an internet gateway.

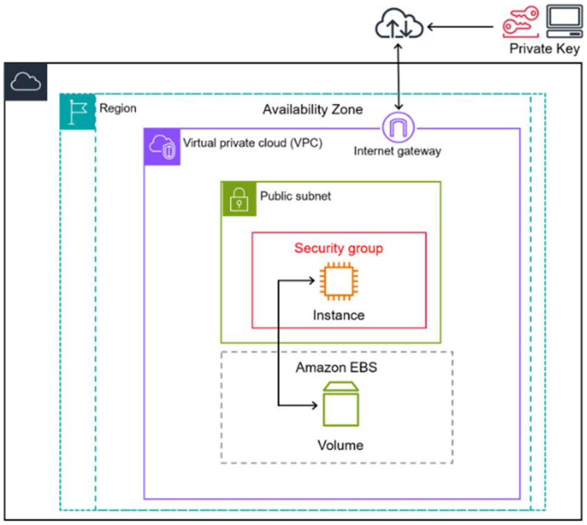


Figure 1 The architecture of EC2

The features of Elastic Cloud Compute are listed below:

- **Instances:** Virtual Servers.
- **Amazon Machine Images (AMIs):** Preconfigured templates for the instances that package the components you need for your server(including the operating system and additional software).

- **Instance types:** Various configurations of CPU, memory, storage, networking capacity, and graphics hardware for your instances.

- **Key pairs:** Secure login information for your instances. AWS stores the public key and you store the private key in a secure place.

- **Instance store volumes:** Storage volumes for temporary data that is deleted when you stop, hibernate, or terminate your instance.

- **Amazon EBS volumes:** Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS).

- **Regions, Availability Zones, Local Zones, AWS Outposts, and Wavelength Zones:** Multiple physical locations for your resources, such as instances and Amazon EBS volumes.

- **Security groups:** A virtual firewall that allows you to specify the protocols, ports, and source IP ranges that can reach your instances, and the destination IP ranges to which your instances can connect.

- **Elastic IP addresses:** Static IPv4 addresses for dynamic cloud computing.

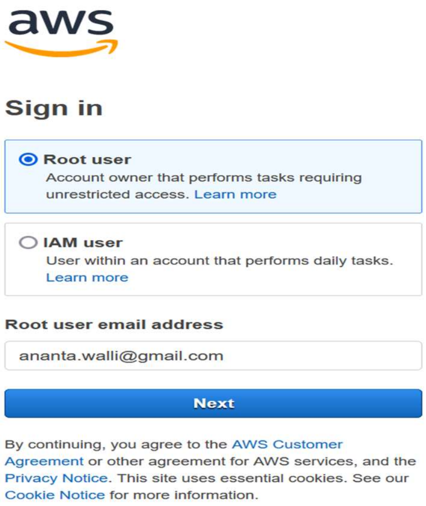
- **Tags:** Metadata that you can create and assign to your Amazon EC2 resources.

- **Virtual private clouds (VPCs):** Virtual networks you can create that are logically isolated from the rest of the AWS Cloud. You can optionally connect these virtual networks to your own network.

Procedure:

To launch an EC2 instance, following steps need to be followed:

Step 1: Log into AWS Management Console with your registered email address and password. Make sure to choose the root user option.



aws

Sign in

● Root user
Account owner that performs tasks requiring unrestricted access. Learn more

○ IAM user
User within an account that performs daily tasks. Learn more

Root user email address

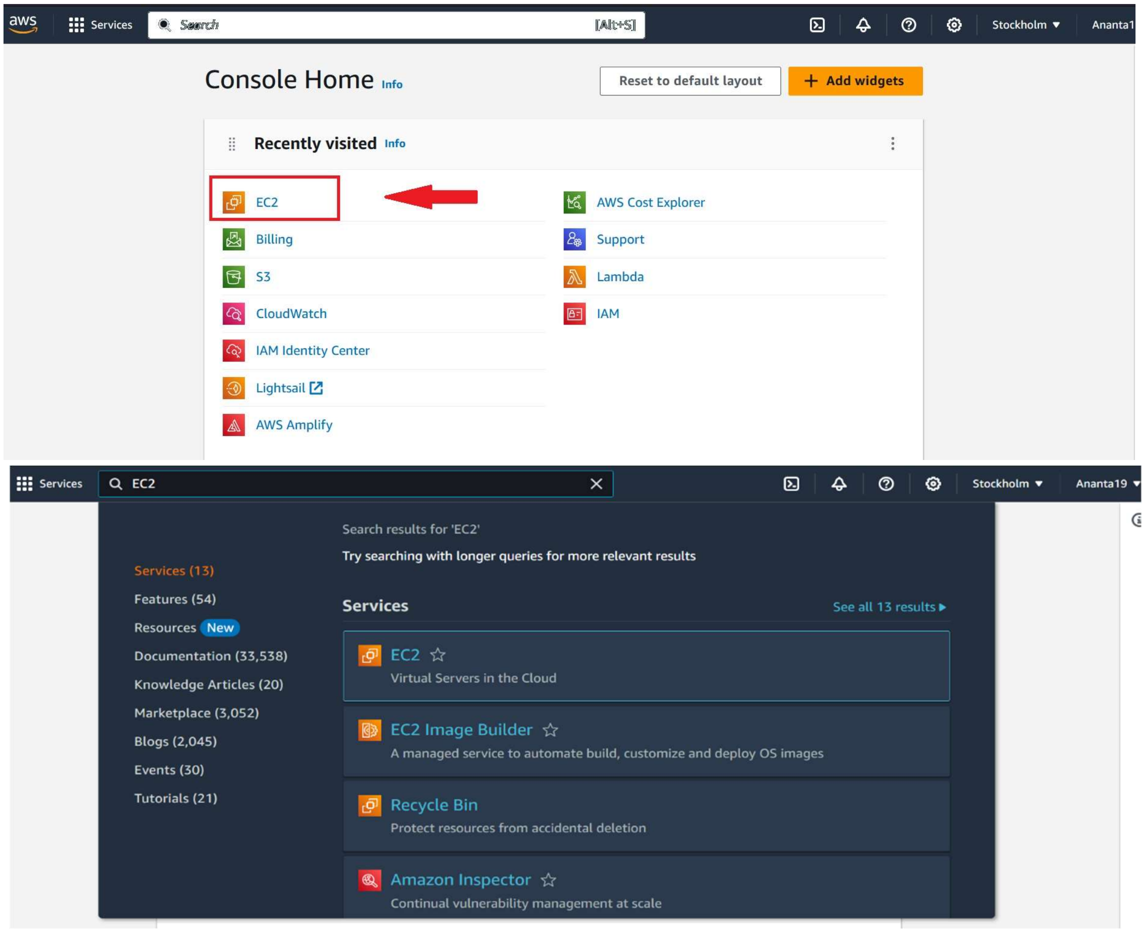
ananta.walli@gmail.com

Next

By continuing, you agree to the AWS Customer Agreement or other agreement for AWS services, and the Privacy Notice. This site uses essential cookies. See our Cookie Notice for more information.

Figure 2 Login portal

Step 2: Once logged in, console home tab will be shown on the screen. Either select the EC2 option mentioned on the console home or search 'EC2' on the search bar located on the top of the screen.



Search results for 'EC2'

Try searching with longer queries for more relevant results

EC2 ☆
Virtual Servers in the Cloud

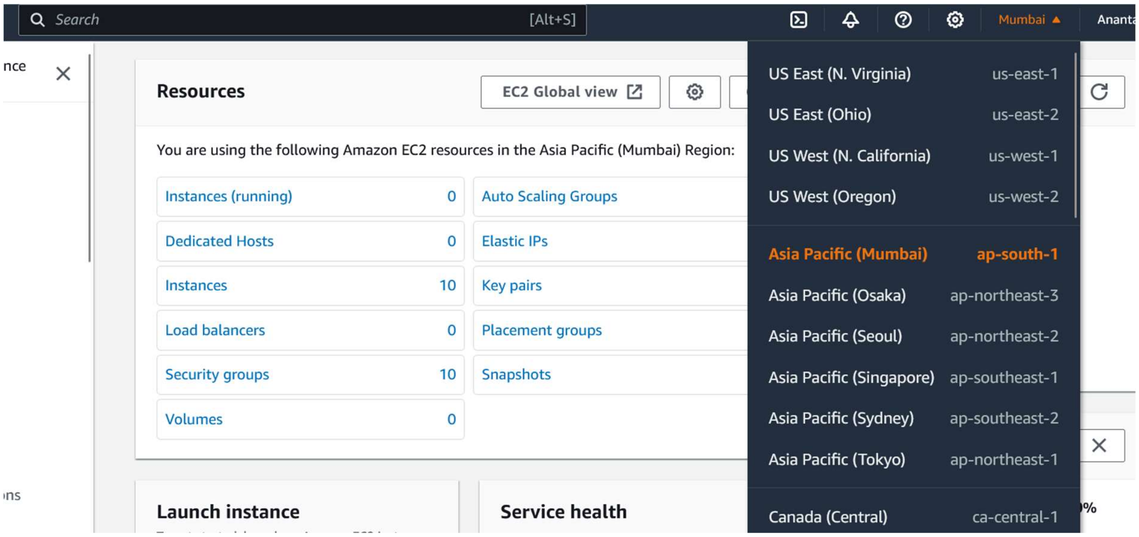
EC2 Image Builder ☆
A managed service to automate build, customize and deploy OS images

Recycle Bin
Protect resources from accidental deletion

Amazon Inspector ☆
Continual vulnerability management at scale

Figure 3 AWS Management Console menu

Step 3: After EC2 option was clicked, the screen will be redirected to EC2 dashboard, make sure to choose availability zone at this point. Any one can be chosen but in order to reduce latency, we will chose Mumbai (ap-south-1) for the instance.

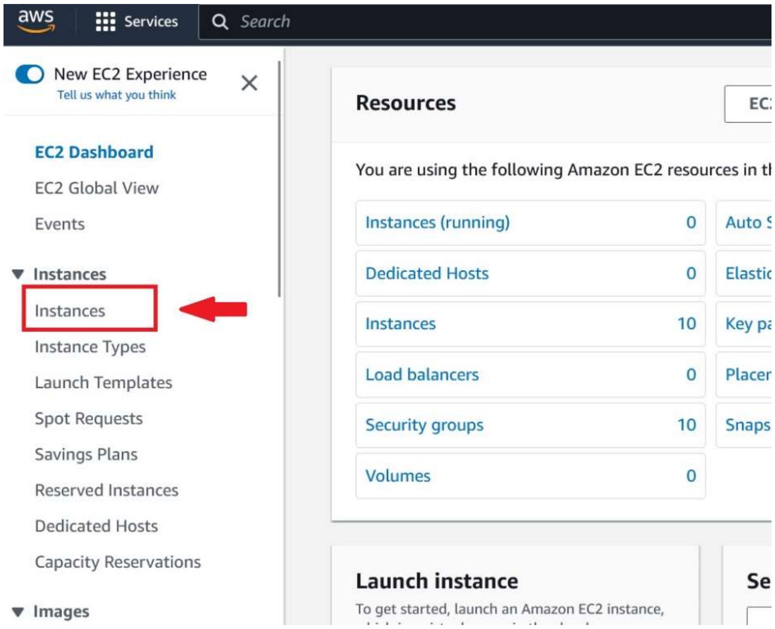

Search
[Alt+S]
Mumbai ▲
Ananta
nce
ns
Resources
EC2 Global view
US East (N. Virginia) us-east-1
US East (Ohio) us-east-2
US West (N. California) us-west-1
US West (Oregon) us-west-2
Asia Pacific (Mumbai) ap-south-1
Asia Pacific (Osaka) ap-northeast-3
Asia Pacific (Seoul) ap-northeast-2
Asia Pacific (Singapore) ap-southeast-1
Asia Pacific (Sydney) ap-southeast-2
Asia Pacific (Tokyo) ap-northeast-1
Canada (Central) ca-central-1
Launch instance
Service health

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

Instances (running)	0	Auto Scaling Groups
Dedicated Hosts	0	Elastic IPs
Instances	10	Key pairs
Load balancers	0	Placement groups
Security groups	10	Snapshots
Volumes	0	

Figure 4 Availability Zone menu

Step 4: After Mumbai has been chosen, we will look into the left navigation panel and click on Instances.



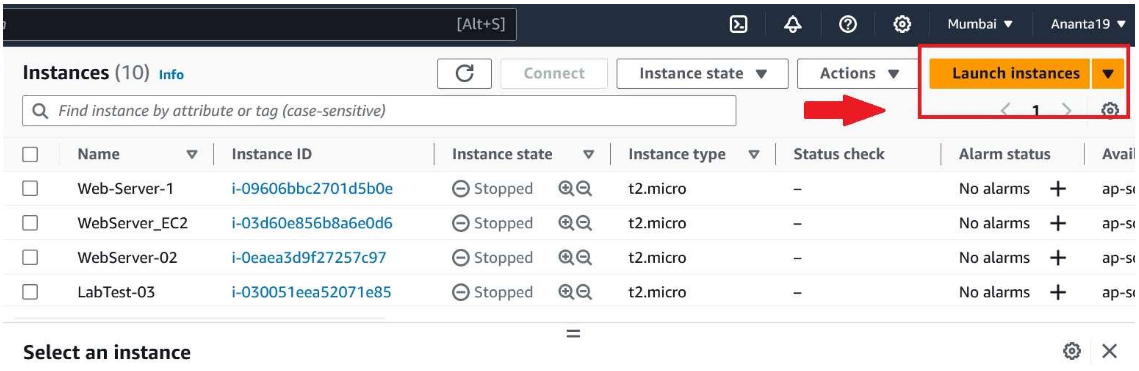
You are using the following Amazon EC2 resources in tl

Instances (running)	0	Auto S
Dedicated Hosts	0	Elastic
Instances	10	Key pa
Load balancers	0	Placer
Security groups	10	Snaps
Volumes	0	

To get started, launch an Amazon EC2 instance,

Figure 5 Navigation Panel(left)

Step 5: After clicking on the instances option, we will now click on Launch Instance button, located on the extreme right, to start creating the instance as required.



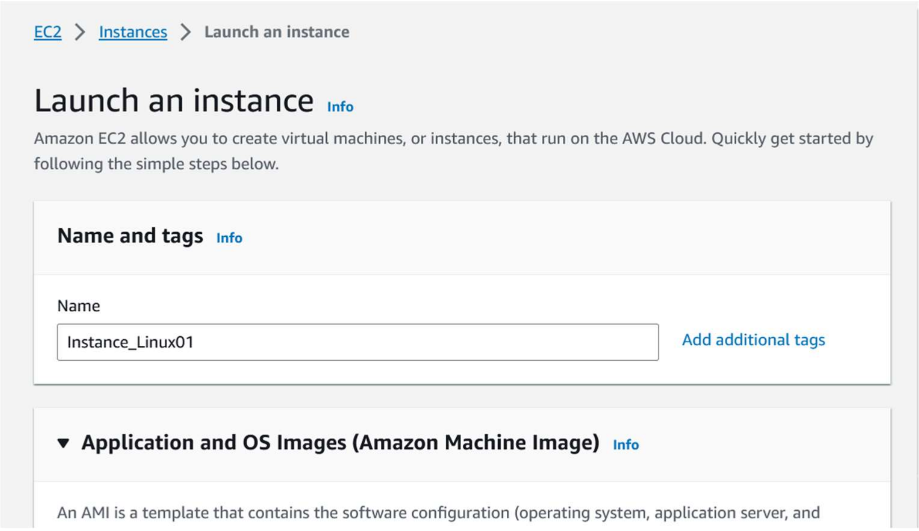
Instances (10)

Find instance by attribute or tag (case-sensitive)

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Avail
□	Web-Server-1	i-09606bbc2701d5b0e	Stopped	t2.micro	–	No alarms +	ap-s
□	WebServer_EC2	i-03d60e856b8a6e0d6	Stopped	t2.micro	–	No alarms +	ap-s
□	WebServer-02	i-0eaea3d9f27257c97	Stopped	t2.micro	–	No alarms +	ap-s
□	LabTest-03	i-030051eea52071e85	Stopped	t2.micro	–	No alarms +	ap-s

Figure 6 Instance Dashboard

Step6: The first step to launch an instance will be to name it. You can give it any name as required.



Launch an instance

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

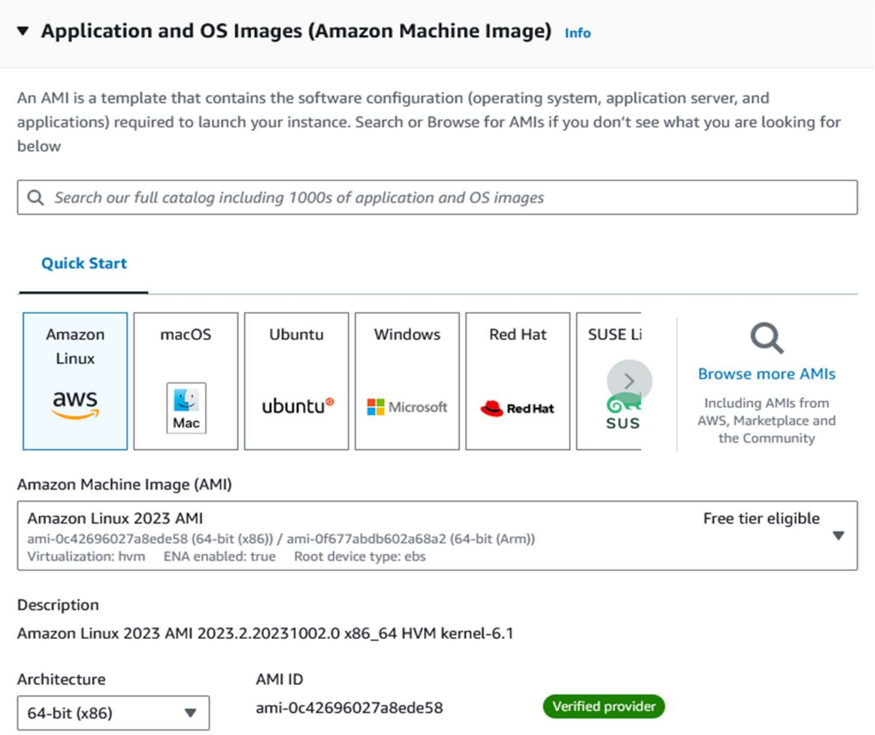
Name: Instance_Linux01

▼ Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and

Figure 7 Naming The Instance

Step 7: Then we need an AMI for the instance to be created. For this experiment we need Linux, so we will choose the **Amazon Linux** option. In case we need some other, we can always click on the **browse more AMIs** option.



▼ 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

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible
ami-0c42696027a8ede58 (64-bit (x86)) / ami-0f677abdb602a68a2 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

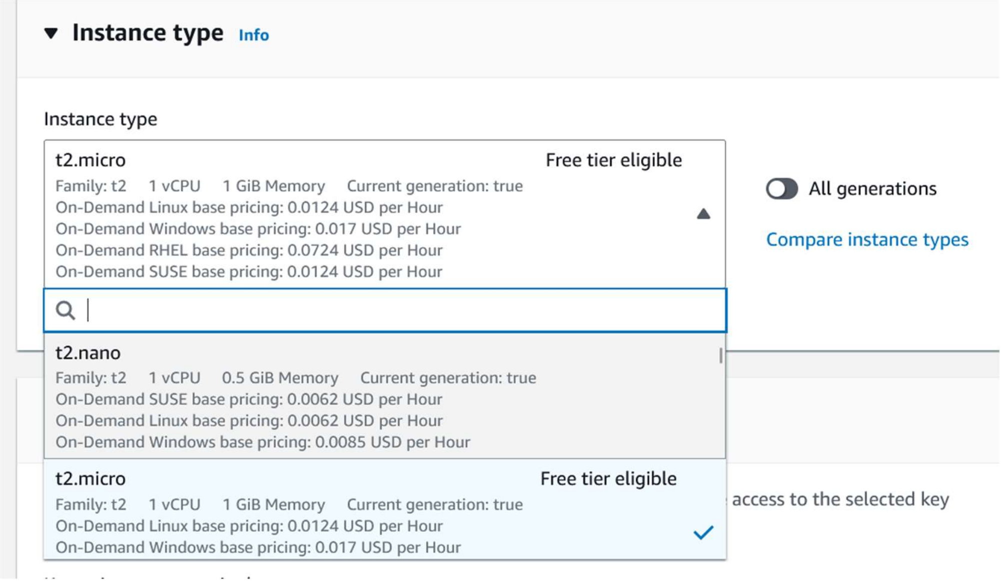
Amazon Linux 2023 AMI 2023.2.20231002.0 x86_64 HVM kernel-6.1

Architecture: 64-bit (x86)

AMI ID: ami-0c42696027a8ede58

Figure 8 Choosing AMI for Instance

Step 8: Now, its time to choose the Instance type. We are using the free tier version, so for that we will choose the t2.micro option .



▼ Instance type Info

Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0124 USD per Hour
On-Demand Windows base pricing: 0.017 USD per Hour
On-Demand RHEL base pricing: 0.0724 USD per Hour
On-Demand SUSE base pricing: 0.0124 USD per Hour

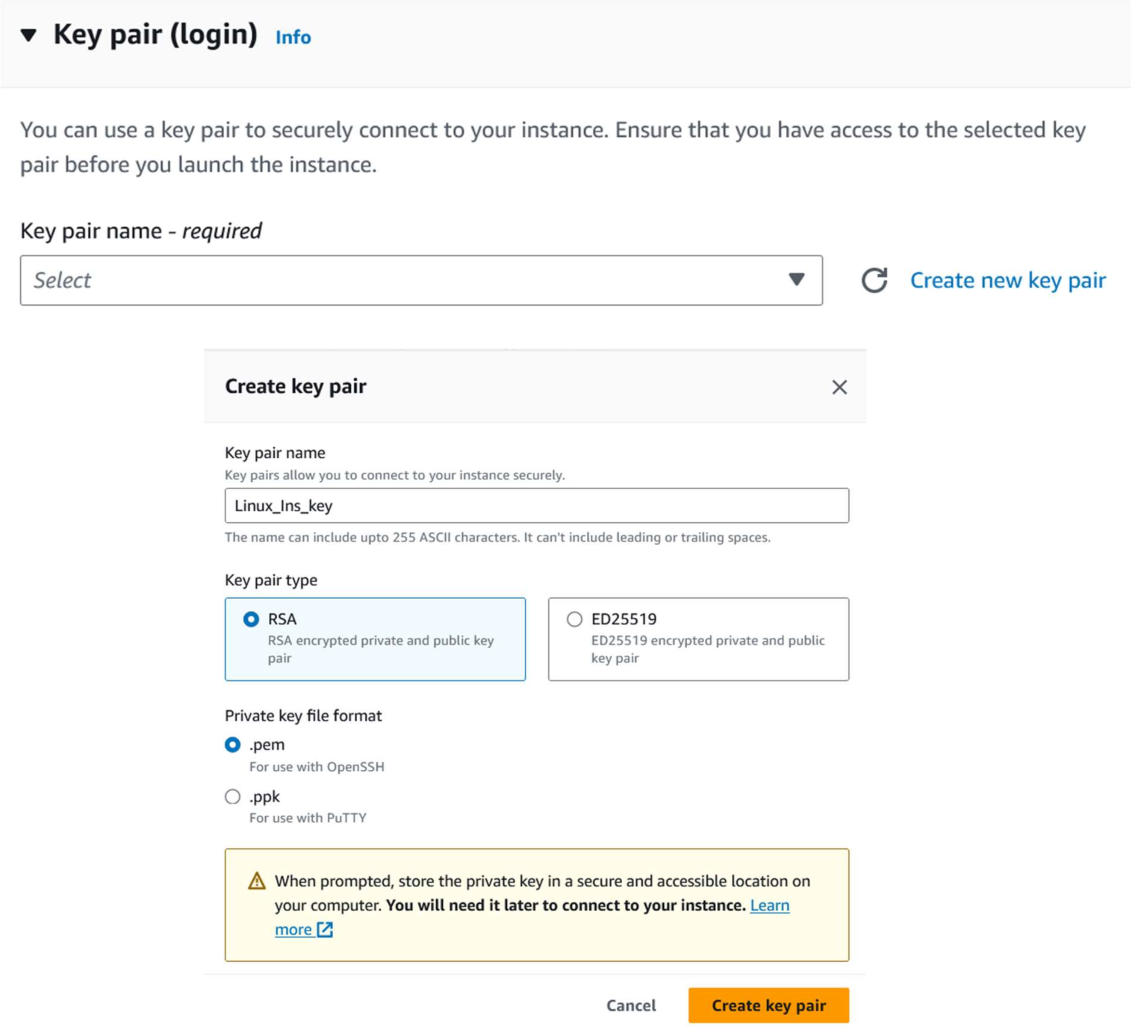
t2.nano
Family: t2 1 vCPU 0.5 GiB Memory Current generation: true
On-Demand SUSE base pricing: 0.0062 USD per Hour
On-Demand Linux base pricing: 0.0062 USD per Hour
On-Demand Windows base pricing: 0.0085 USD per Hour

t2.micro — Free tier eligible ✓
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0124 USD per Hour
On-Demand Windows base pricing: 0.017 USD per Hour

access to the selected key

Figure 9 Choosing Instance Type

Step 9: For Instance, we need a key pair. We can either select any existing key pair used for other instance or we can create one. So, we are going to create a key pair for this instance. The key pair is a set of security credentials that you use to prove your identity when connecting to an Amazon EC2 instance.



▼ 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 pair name

Key pairs allow you to connect to your instance securely.

Linux_Ins_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

Private key file format

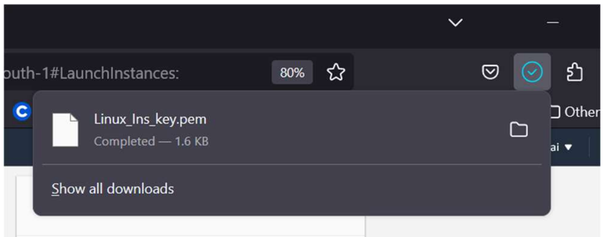
● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

⚠ 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

Figure 10 Creating Key Pair

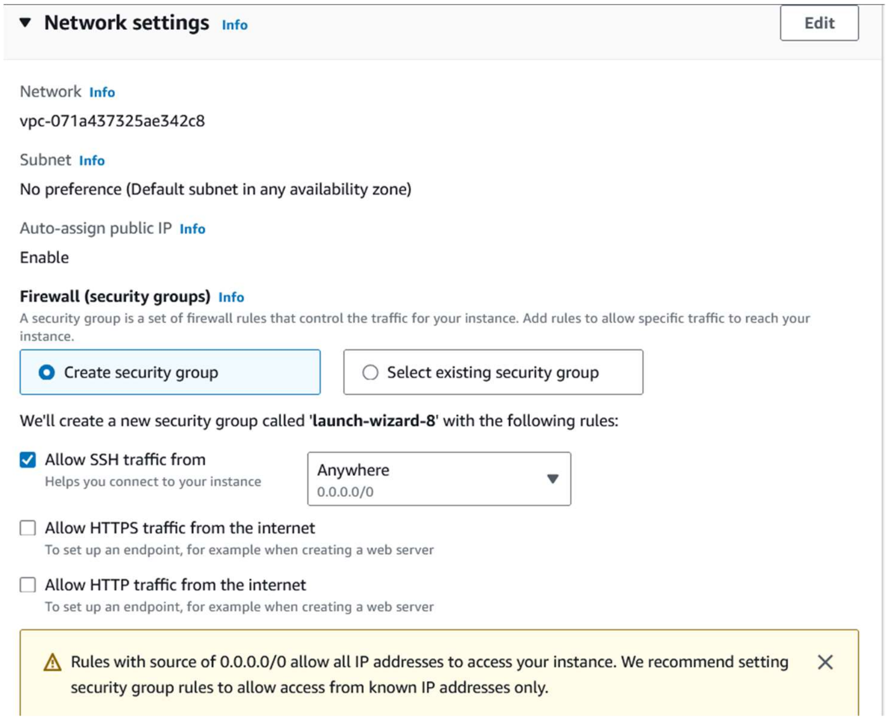
For key pair type we will choose RSA and for the private key file format we are going to choose .pem file. Then we will click on create key pair and it will be downloaded on the system in the form of .pem file.



Linux_Ins_key.pem
Completed — 1.6 KB

Figure 11 Key Pair Downloaded on System

Step 10: Now for the network setting, we are going to retain the default settings and go ahead with creating a security group for our instance. In case we are creating web servers, we will check mark HTTP and HTTPS so that this traffic is enabled. But we don't want web servers right now, so we will allow only SSH traffic.



▼ Network settings Info

Network Info
vpc-071a437325ae342c8

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-8**' with the following rules:

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

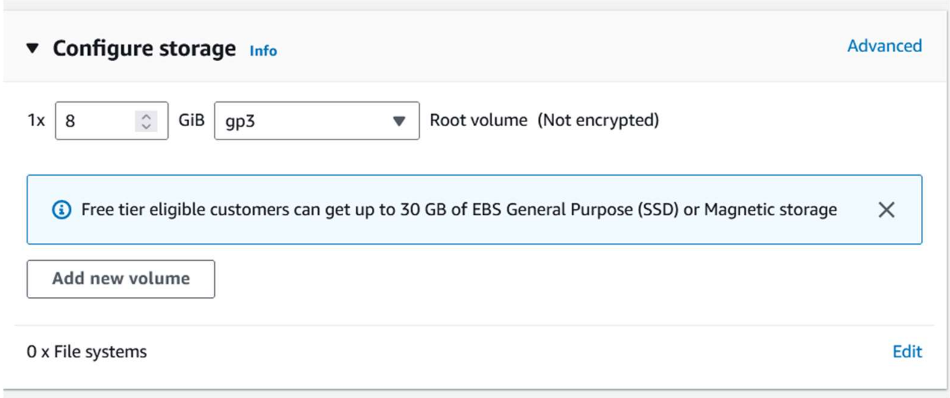
☐ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

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

⚠ 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.

Figure 12 Configuring Network Settings For Instance

Step 11: Again for the volume, we are going to retain default settings and let 8Gib root volume be attached and created for the instance.



▼ Configure storage Info

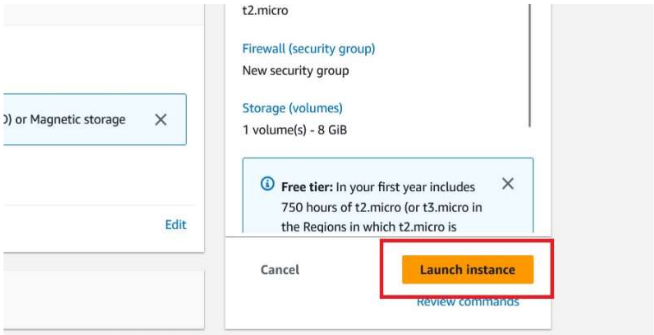
1x 8 GiB gp3 Root volume (Not encrypted)

ⓘ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

0 x File systems

Figure 13 Root volume attached to Instance

Step 12: We will now click on launch instance button on the right and wait for 10 secs for it to be created.



t2.micro

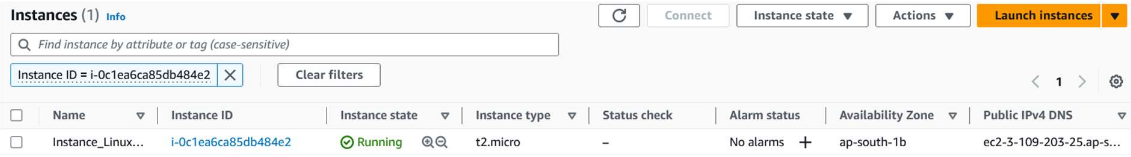
Firewall (security group)
New security group

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

Figure 14 Launching of Instance

Step 13: It takes about 30 secs for an instance to come into Running state. After it reaches the running state, it will look something like this:

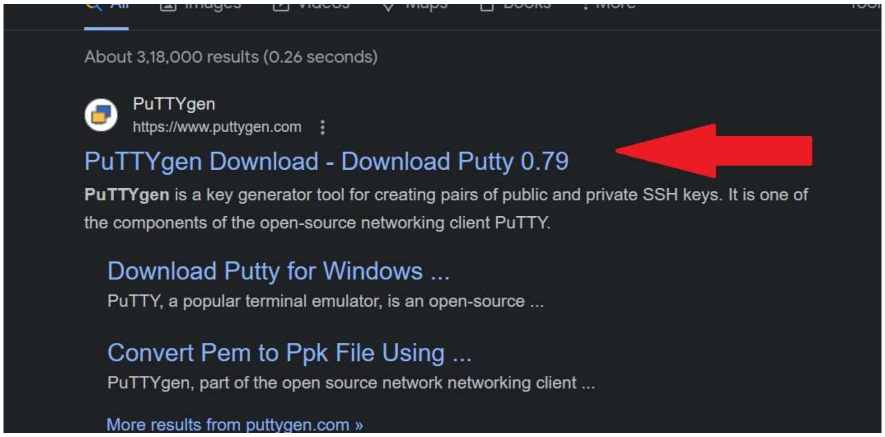


Instances (1) Info
Connect
Instance state ▼
Actions ▼
Launch instances ▼
Find instance by attribute or tag (case-sensitive)
Instance ID = i-0c1ea6ca85db484e2
Clear filters
1

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
□	Instance_Linux...	i-0c1ea6ca85db484e2	✓ Running	t2.micro	–	No alarms +	ap-south-1b	ec2-3-109-203-25.ap-s...

Figure 15 Checking the Instance State

Step 14: To now make the Linux machine run onto the system, we will download Puttygen, if not available on the system.



About 3,18,000 results (0.26 seconds)

PuTTYgen
https://www.puttygen.com

PuTTYgen Download - Download Putty 0.79

PuTTYgen is a key generator tool for creating pairs of public and private SSH keys. It is one of the components of the open-source networking client PuTTY.

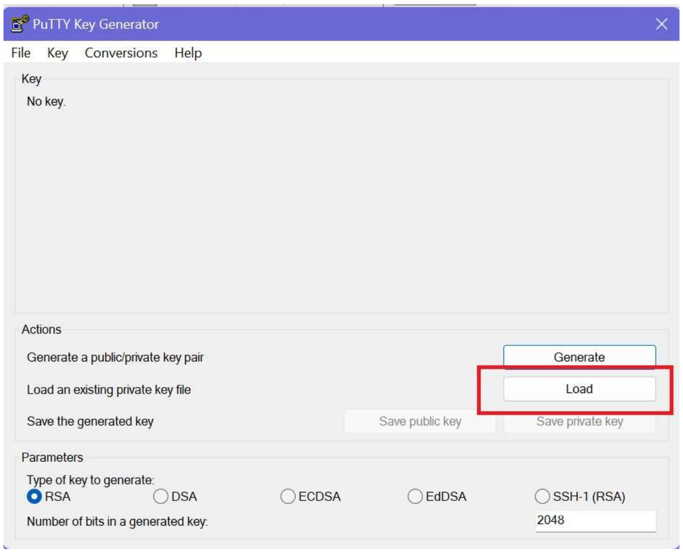
Download Putty for Windows ...
PuTTY, a popular terminal emulator, is an open-source ...

Convert Pem to Ppk File Using ...
PuTTYgen, part of the open source network networking client ...

More results from puttygen.com »

Figure 16 Downloading PuTTYgen on the System

Step 15: After downloading it, we will open up the PuTTYgen on the system. We will click on the load button and upload the .pem file of the key pair we had created while creating the instance.



PuTTY Key Generator

Key

No key.

Actions

Generate a public/private key pair — Generate

Load an existing private key file — Load

Save the generated key — Save public key — Save private key

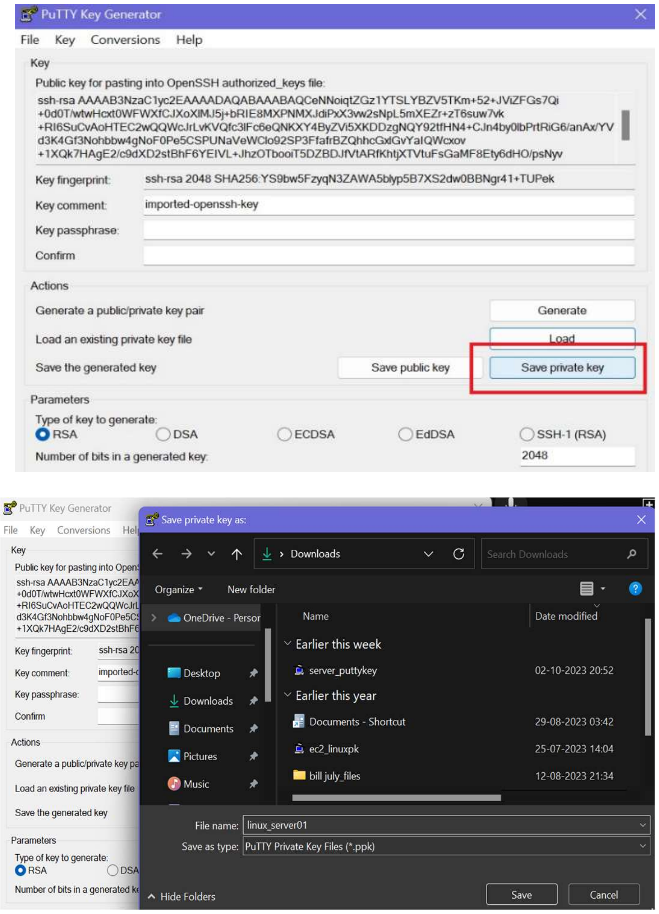
Parameters

Type of key to generate:
● RSA ○ DSA ○ ECDSA ○ EdDSA ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

Figure 17 Opening PuTTY Key Generator

Step 16: The key pair will be loaded and we will click on the option to save it as private key on the system. This will save it as PuTTY private key file on the system.



PuTTY Key Generator

Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCeNNoiqtZGz1YTSLYBZV5TKm+52+JViZFGs7Qi
+0d0T/wtwHcxt0WFWXfCJXoXlMJ5j+bRIE8MXPNMXJdiPxX3vw2sNpL5mXEZr+zT6suw7vk
+RI6SuCvAoHTEC2wQQWcJrLvKVQfc3lFc6eQNKXY4ByZVi5XKDDzgNQY92tfHN4+CJn4by0lbPrtRiG6/anAx/YV
d3K4Gf3Nohbbw4gNoF0Pe5CSPUNaVeWClo92SP3FfafrBZQhhcGxlGvYaIQWcxov
+1XQk7HAgE2/c9dXD2stBhF6YEIVL+JhzOTbooiT5DZBDJfVtARfKhtjXTVtuFsGaMF8Ety6dHO/psNyv
```

Key fingerprint: ssh-rsa 2048 SHA256:YS9bw5FzyqN3ZAWA5blyp5B7XS2dw0BBNgr41+TUPek

Key comment: imported-openssh-key

Key passphrase:

Confirm:

Actions

Generate a public/private key pair — Generate

Load an existing private key file — Load

Save the generated key — Save public key — Save private key

Parameters

Type of key to generate:
● RSA ○ DSA ○ ECDSA ○ EdDSA ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

Save private key as:

Downloads

Name	Date modified
Earlier this week	
server_puttykey	02-10-2023 20:52
Earlier this year	
Documents - Shortcut	29-08-2023 03:42
ec2_linuxpk	25-07-2023 14:04
bill july_files	12-08-2023 21:34

File name: linux_server01

Save as type: PuTTY Private Key Files (*.ppk)

Figure 18 Saving the private key on system

Step 17: Next, we will open PuTTY on the system, we will click on Session->SSH->Auth->Credentials on the left pane and attach the private key that was just saved on the system. The file can be attached under the option **Private key file for authentication.**

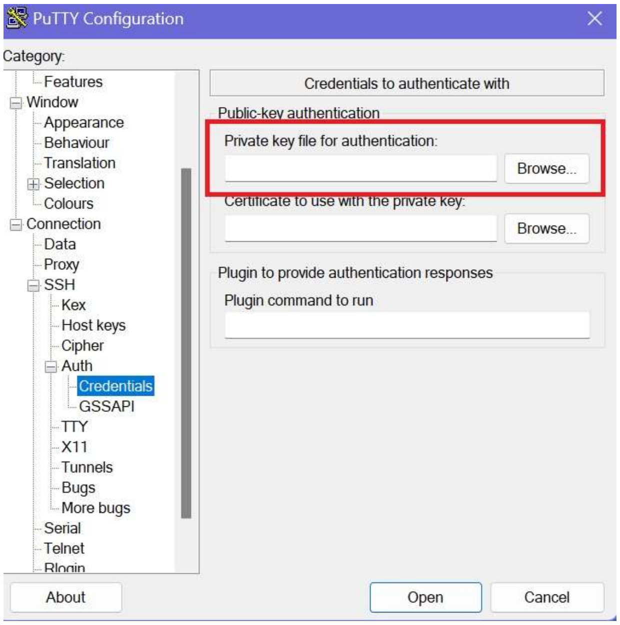
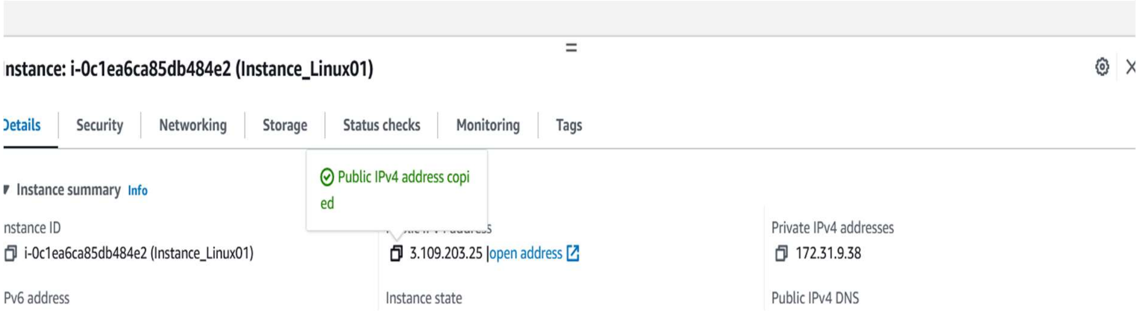


Figure 19 Opening up of private key for authentication

Step 18: We will now go to Session and copy paste the public IPv4 address of the instance created on AWS.


nstance: i-0c1ea6ca85db484e2 (Instance_Linux01)

)etails | Security | Networking | Storage | Status checks | Monitoring | Tags

Public IPv4 address copi
ed

Instance summary Info

nstance ID
i-0c1ea6ca85db484e2 (Instance_Linux01)

Public IPv4 address
3.109.203.25 | open address

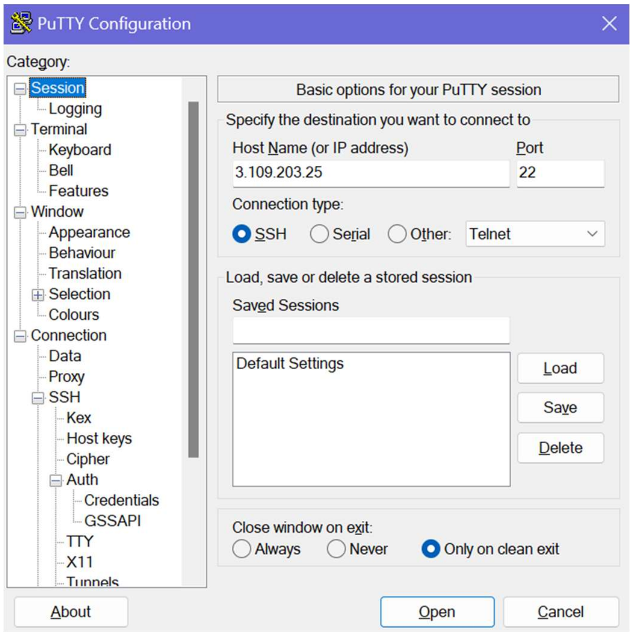
Private IPv4 addresses
172.31.9.38

Pv6 address

Instance state

Public IPv4 DNS

Figure 20 Copying the Public IPv4 address of Instance to be launched



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 3.109.203.25

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

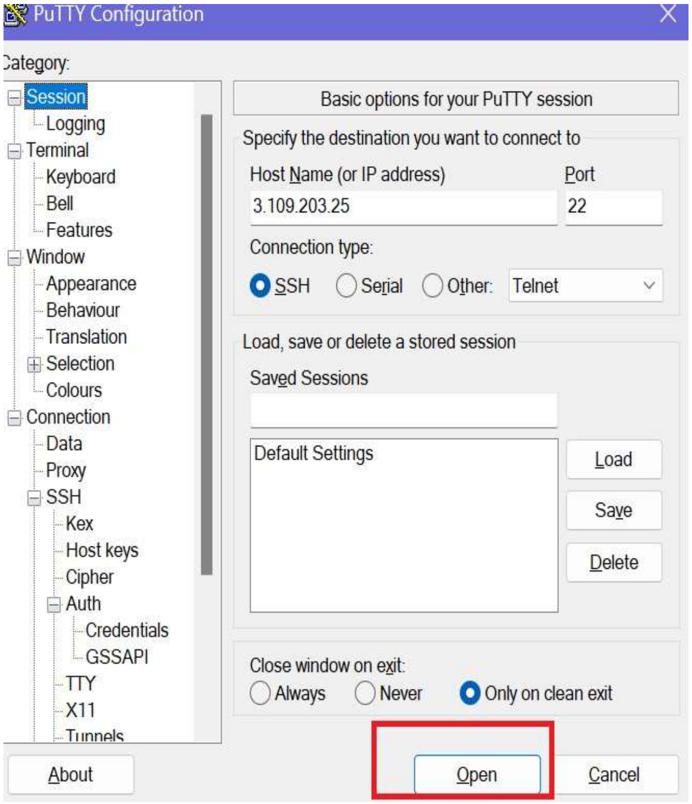
Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 21 The Ip address is pasted under Host Name

Step 19: We will then click on the Open option at bottom and enter the command ec2-user and our Amazon Linux Server will successfully be launched.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 3.109.203.25

Port: 22

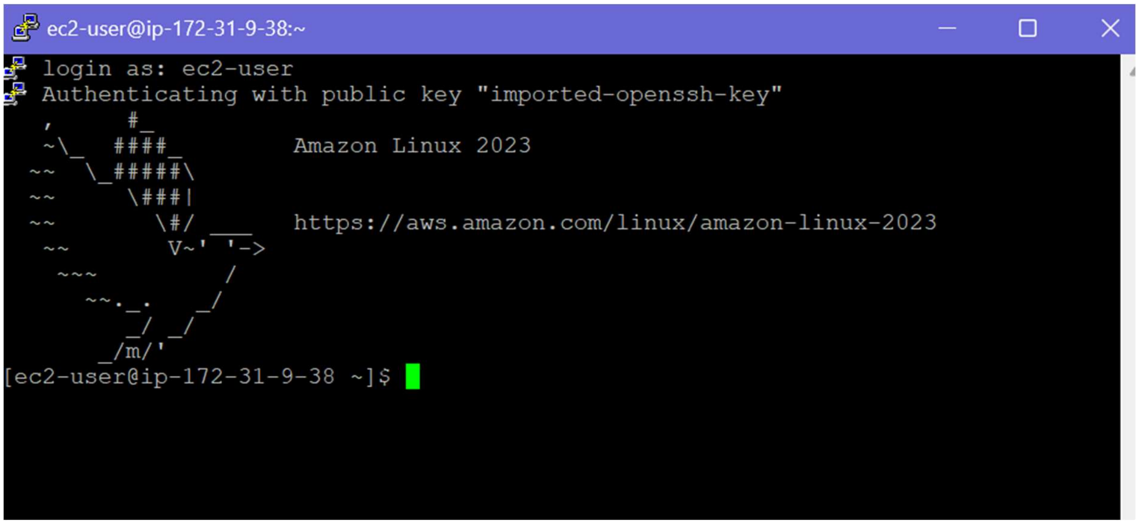
Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 22 Clicking on the Open Button to launch Instance



```
   login as: ec2-user
   Authenticating with public key "imported-openssh-key"
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-9-38 ~]$ █
```

Figure 23 The Linux Instance is launched successfully

Step 21: As this instance is successfully launched, we can close it and stop the instance as required.

Result:

The EC2 instance with Linux AMI and suitable configurations has been successfully launched on the system.