Name: Balasaheb Narayan Hinge

Class: TE A (Batch:C)

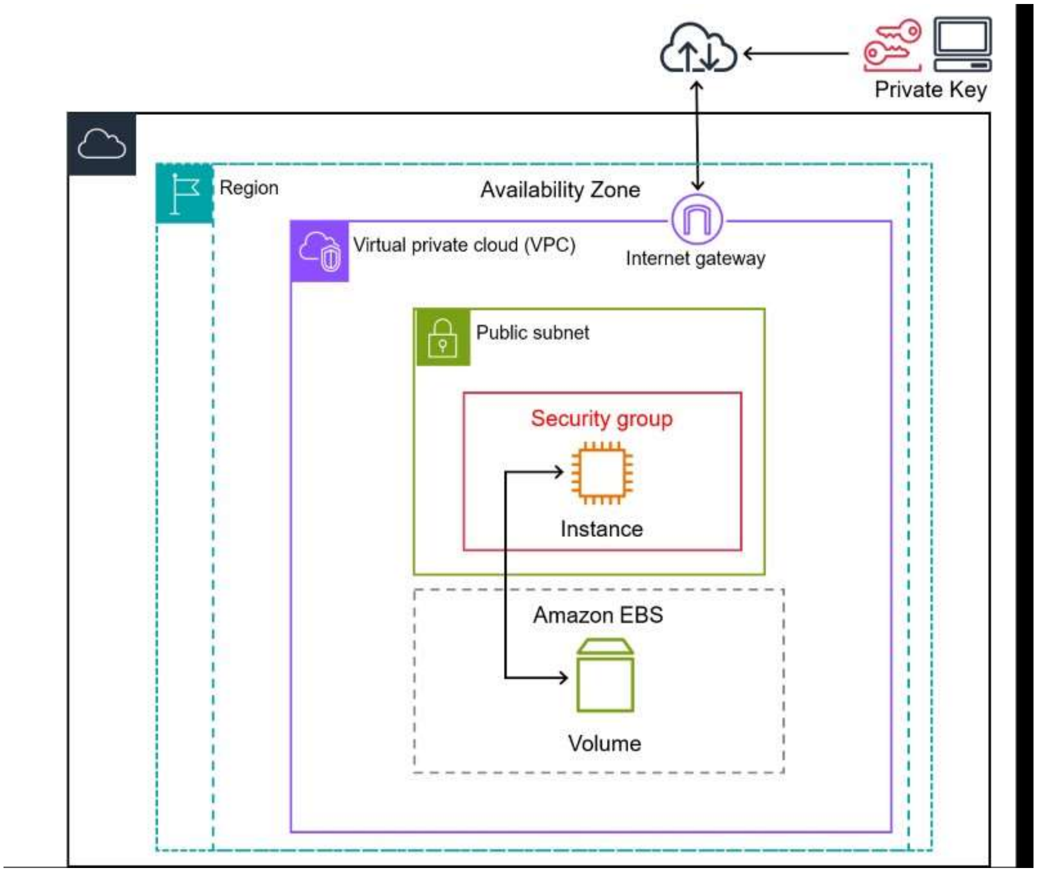
Roll no.: 21CO047

Problem Statement:
Case study on Amazon EC2 and learn about Amazon EC2 web services

---

● What is Amazon EC2?

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 so you can develop and deploy applications faster.



In this example, 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.

Features of Amazon EC2

Amazon EC2 provides the following high-level features:

Instances

Virtual servers.

Amazon Machine Images (AMIs)

Preconfigured templates for your 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

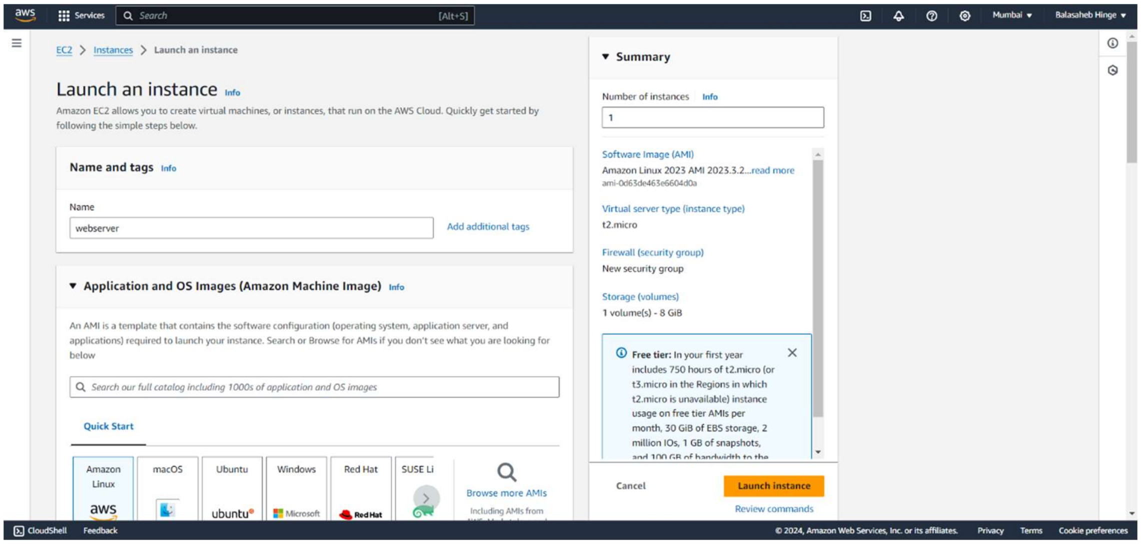
Static IPv4 addresses for dynamic cloud computing.

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.



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: webserver

Application and OS Images (Amazon Machine Image)

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

Summary

Number of instances: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.3.2...read more
ami-0d63de463e6604d0a

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 IOs, 1 GB of snapshots, and 100 GB of bandwidth to the

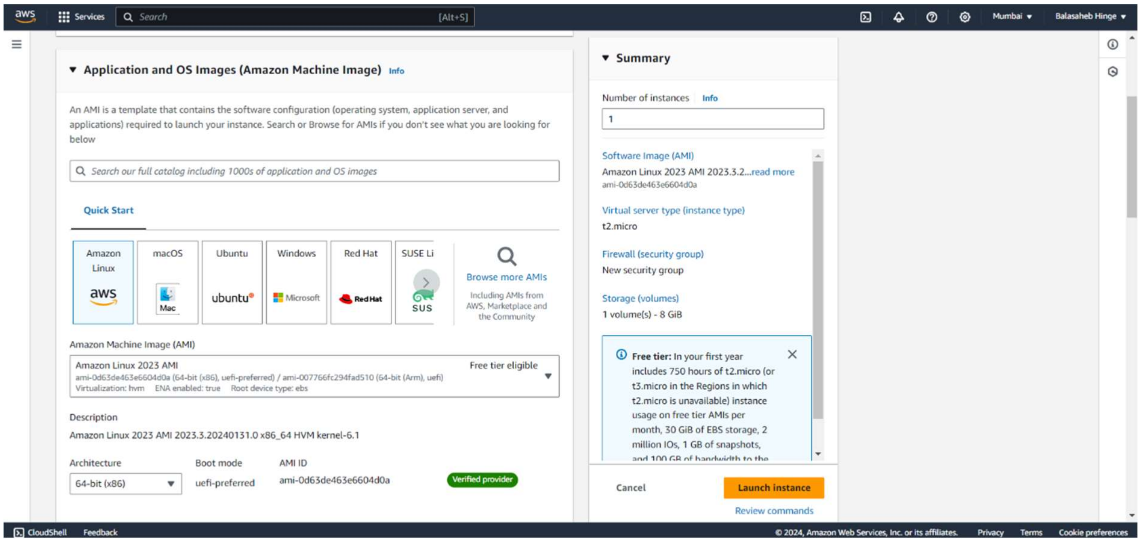
To launch an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.

2. From the EC2 console dashboard, in the Launch instance box, choose Launch instance.
3. Under Name and tags, for Name, enter a descriptive name for your instance.
4. Under Application and OS Images (Amazon Machine Image), do the following:

a. Choose Quick Start, and then choose Amazon Linux. This is the operating system (OS) for your instance.

b. From Amazon Machine Image (AMI), select an HVM version of Amazon Linux 2. Notice that these AMIs are marked Free Tier eligible. An *Amazon Machine Image (AMI)* is a basic configuration that serves as a template for your instance.



Application and OS Images (Amazon Machine Image)

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-0d63de463e6604d0a (64-bit (x86), uefi-preferred) / ami-007766fc294fad510 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Amazon Linux 2023 AMI 2023.3.20240131.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID
64-bit (x86)	uefi-preferred	ami-0d63de463e6604d0a

Summary

Number of instances: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.3.2...read more
ami-0d63de463e6604d0a

Virtual server type (instance type)
t2.micro

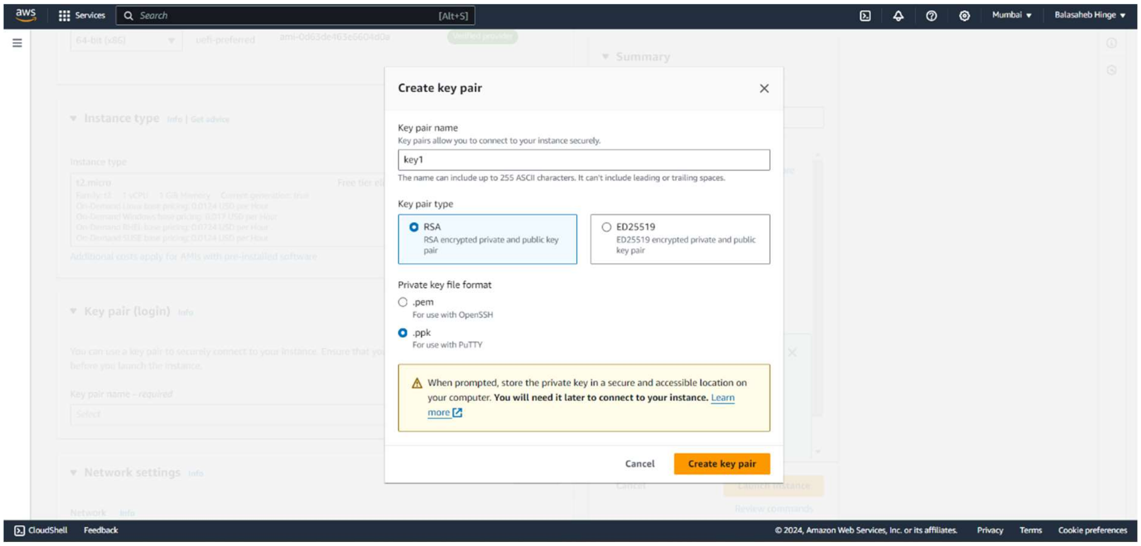
Firewall (security group)
New security group

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 IOs, 1 GB of snapshots, and 100 GB of bandwidth to the

© 2024, Amazon Web Services, Inc. or its affiliates.

5. Under Instance type, from the Instance type list, you can select the hardware configuration for your instance. Choose the `t2.micro` instance type, which is selected by default. The `t2.micro` instance type is eligible for the Free Tier. In Regions where `t2.micro` is unavailable, you can use a `t3.micro` instance under the Free Tier.
6. Under Key pair (login), for Key pair name, choose the key pair that you created when getting set up.



Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

key1

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

Key pair type

● RSA
RSA encrypted private and public key pair

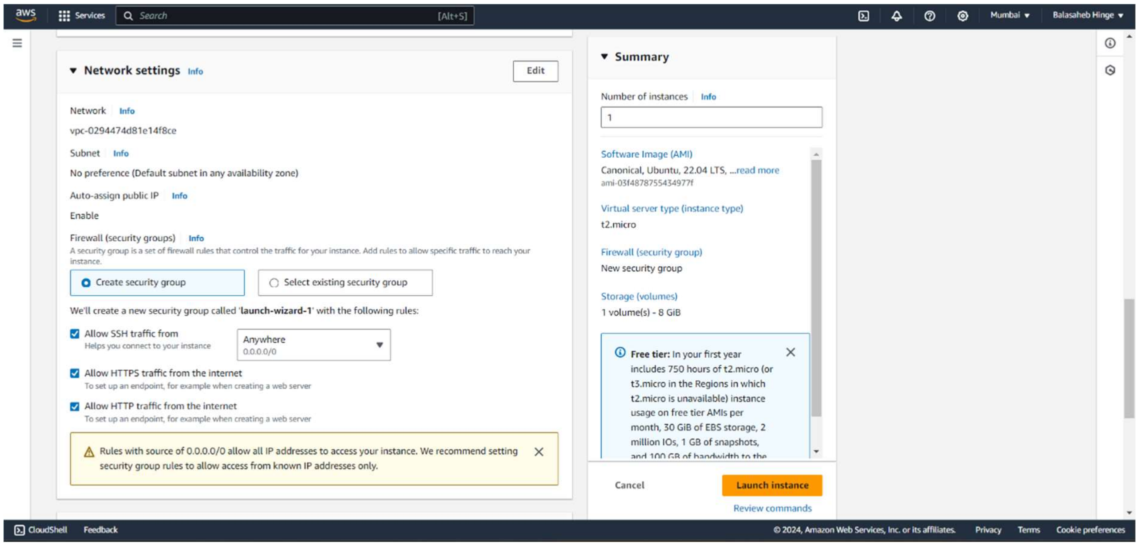
○ ED25519
ED25519 encrypted private and public key pair

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



Network settings Info

Network Info
vpc-0294474d81e14f8ce

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-1**' 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.

Summary

Number of instances Info
1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-03f4878755434977f

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 IOs, 1 GB of snapshots, and 100 GB of bandwidth to the

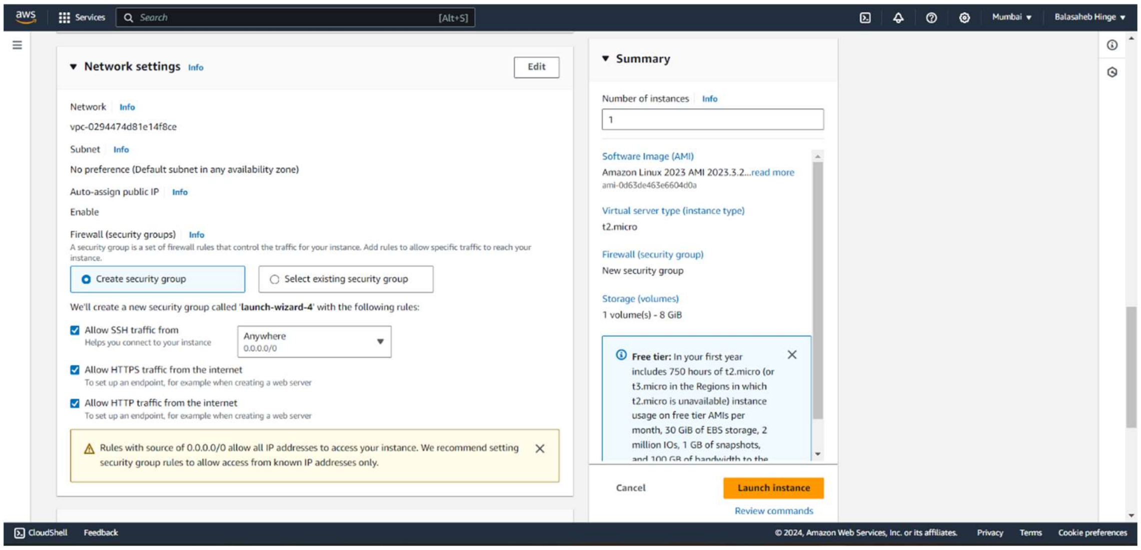
1. Next to Network settings, choose Edit.For Security group name, you'll see that the wizard created and selected a security group for you. You can use this security group, or alternatively you can select the security group that you created when getting set up using the following steps:

a. Choose Select existing security group.

b. From Common security groups, choose your security group from the list of existing security groups.

2. Keep the default selections for the other configuration settings for your instance.

3. Review a summary of your instance configuration in the Summary panel, and when you're ready, choose Launch instance.



Network settings

Network: vpc-0294474d81e14f8ce

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-4**' with the following rules:

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

☑ 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.

Summary

Number of instances: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.3.2...read more
ami-0d63de463e6604d0a

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 IOs, 1 GB of snapshots, and 100 GB of bandwidth to the

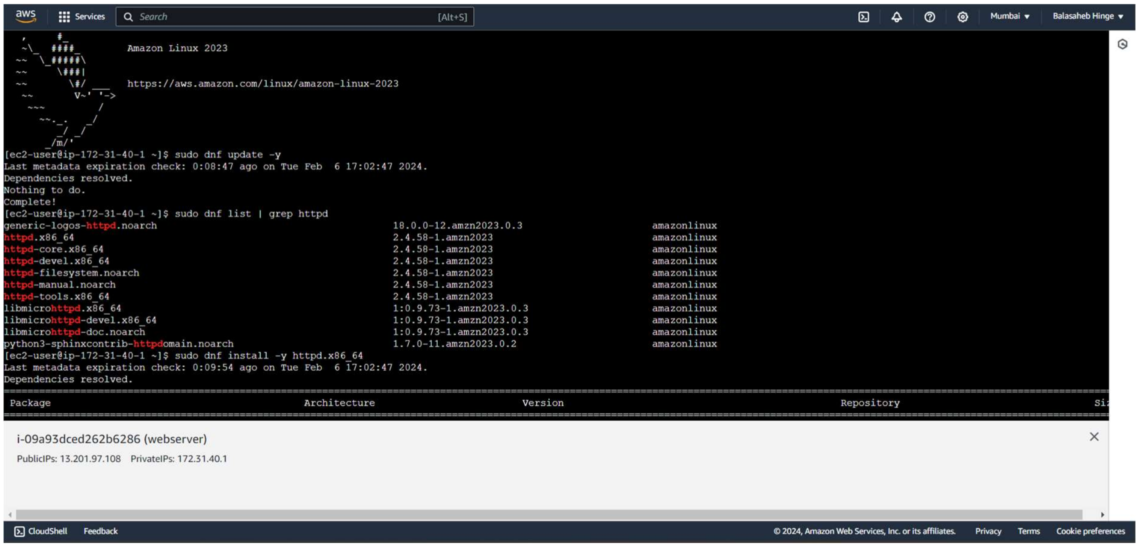
1. A confirmation page lets you know that your instance is launching. Choose View all instances to close the confirmation page and return to the console.
2. On the Instances screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is `pending`. After the instance starts, its state changes to `running` and it receives a public DNS name. If the Public IPv4 DNS column is hidden, choose the settings icon (⚙) in the top-right corner, toggle on Public IPv4 DNS, and choose Confirm.
3. It can take a few minutes for the instance to be ready for you to connect to it. Check that your instance has passed its status checks; you can view this information in the Status check column.

Install Apache Web Server on Amazon Linux 2023

1Step 1: Update the System Package to Latest

Let’s start updating the packages to the latest ones before installing the Apache web server to ***httpd***.

sudo yum update -y



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-40-1 ~]$ sudo dnf update -y
Last metadata expiration check: 0:08:47 ago on Tue Feb  6 17:02:47 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-40-1 ~]$ sudo dnf list | grep httpd
generic-logos-httpd.noarch                            18.0.0-12.amzn2023.0.3                   amazonlinux
httpd.x86_64                                          2.4.58-1.amzn2023                        amazonlinux
httpd-core.x86_64                                     2.4.58-1.amzn2023                        amazonlinux
httpd-devel.x86_64                                    2.4.58-1.amzn2023                        amazonlinux
httpd-filesystem.noarch                               2.4.58-1.amzn2023                        amazonlinux
httpd-manual.noarch                                   2.4.58-1.amzn2023                        amazonlinux
httpd-tools.x86_64                                    2.4.58-1.amzn2023                        amazonlinux
libmicrohttpd.x86_64                                  1:0.9.73-1.amzn2023.0.3                  amazonlinux
libmicrohttpd-devel.x86_64                            1:0.9.73-1.amzn2023.0.3                  amazonlinux
libmicrohttpd-doc.noarch                              1:0.9.73-1.amzn2023.0.3                  amazonlinux
python3-sphinxcontrib-httpdomain.noarch               1.7.0-11.amzn2023.0.2                    amazonlinux
[ec2-user@ip-172-31-40-1 ~]$ sudo dnf install -y httpd.x86_64
Last metadata expiration check: 0:09:54 ago on Tue Feb  6 17:02:47 2024.
Dependencies resolved.
==========================================================================================================================
 Package                                  Architecture                  Version                                  Repository                  Si
==========================================================================================================================
```

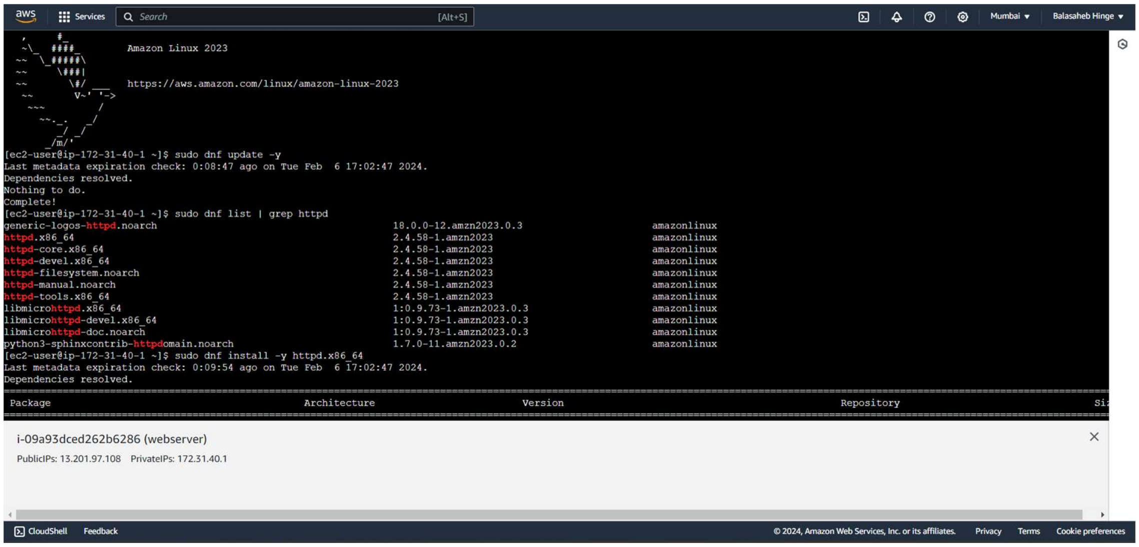
i-09a93dced262b6286 (webserver)

PublicIPs: 13.201.97.108 PrivateIPs: 172.31.40.1

Step 2: Search for the *httpd* package

Apache Web Server package is known as **httpd** for Linux systems. Let’s search for it by below commands.

sudo dnf list | grep httpd



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-40-1 ~]$ sudo dnf update -y
Last metadata expiration check: 0:08:47 ago on Tue Feb  6 17:02:47 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-40-1 ~]$ sudo dnf list | grep httpd
generic-logos-httpd.noarch                             18.0.0-12.amzn2023.0.3                   amazonlinux
httpd.x86_64                                           2.4.58-1.amzn2023                        amazonlinux
httpd-core.x86_64                                      2.4.58-1.amzn2023                        amazonlinux
httpd-devel.x86_64                                     2.4.58-1.amzn2023                        amazonlinux
httpd-filesystem.noarch                                2.4.58-1.amzn2023                        amazonlinux
httpd-manual.noarch                                    2.4.58-1.amzn2023                        amazonlinux
httpd-tools.x86_64                                     2.4.58-1.amzn2023                        amazonlinux
libmicrohttpd.x86_64                                   1:0.9.73-1.amzn2023.0.3                  amazonlinux
libmicrohttpd-devel.x86_64                             1:0.9.73-1.amzn2023.0.3                  amazonlinux
libmicrohttpd-doc.noarch                               1:0.9.73-1.amzn2023.0.3                  amazonlinux
python3-sphinxcontrib-httpdomain.noarch                1.7.0-11.amzn2023.0.2                    amazonlinux
[ec2-user@ip-172-31-40-1 ~]$ sudo dnf install -y httpd.x86_64
Last metadata expiration check: 0:09:54 ago on Tue Feb  6 17:02:47 2024.
Dependencies resolved.
================================================================================================================================
 Package                                    Architecture                  Version                                   Repository           Si
================================================================================================================================
```

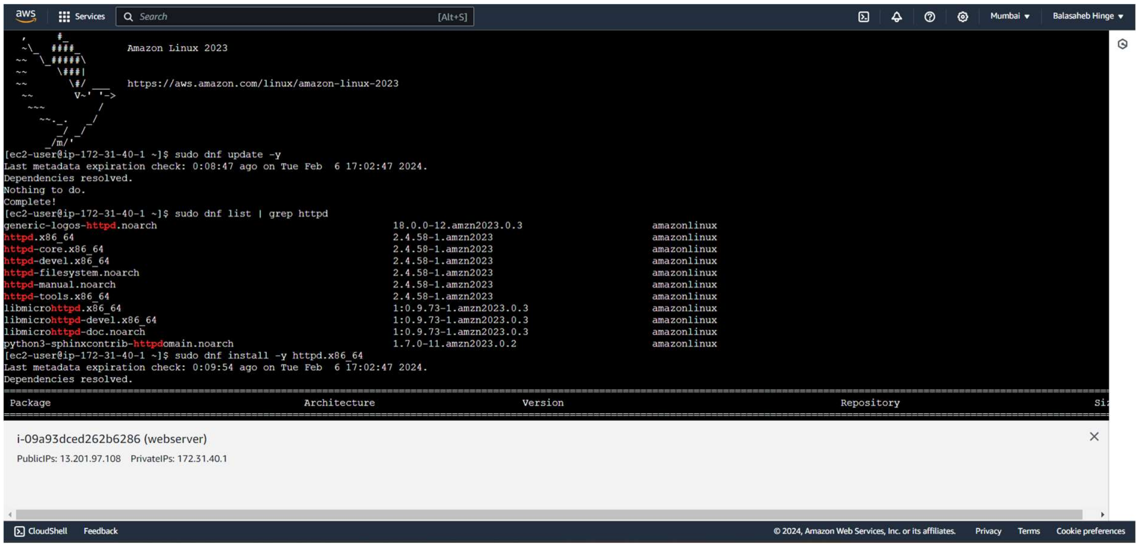
i-09a93dced262b6286 (webserver)

PublicIPs: 13.201.97.108 PrivateIPs: 172.31.40.1

Found the package as seen in the screenshot(*httpd.x86_64*).

Step 3: Install Apache Web Server

sudo yum install -y httpd.x86_64



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-40-1 ~]$ sudo dnf update -y
Last metadata expiration check: 0:08:47 ago on Tue Feb  6 17:02:47 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-40-1 ~]$ sudo dnf list | grep httpd
generic-logos-httpd.noarch                             18.0.0-12.amzn2023.0.3                   amazonlinux
httpd.x86_64                                           2.4.58-1.amzn2023                        amazonlinux
httpd-core.x86_64                                      2.4.58-1.amzn2023                        amazonlinux
httpd-devel.x86_64                                     2.4.58-1.amzn2023                        amazonlinux
httpd-filesystem.noarch                                2.4.58-1.amzn2023                        amazonlinux
httpd-manual.noarch                                    2.4.58-1.amzn2023                        amazonlinux
httpd-tools.x86_64                                     2.4.58-1.amzn2023                        amazonlinux
libmicrohttpd.x86_64                                   1:0.9.73-1.amzn2023.0.3                  amazonlinux
libmicrohttpd-devel.x86_64                             1:0.9.73-1.amzn2023.0.3                  amazonlinux
libmicrohttpd-doc.noarch                               1:0.9.73-1.amzn2023.0.3                  amazonlinux
python3-sphinxcontrib-httpdomain.noarch                1.7.0-11.amzn2023.0.2                    amazonlinux
[ec2-user@ip-172-31-40-1 ~]$ sudo dnf install -y httpd.x86_64
Last metadata expiration check: 0:09:54 ago on Tue Feb  6 17:02:47 2024.
Dependencies resolved.
================================================================================================================================
 Package                                    Architecture                  Version                                   Repository           Si
================================================================================================================================
```

i-09a93dced262b6286 (webserver)

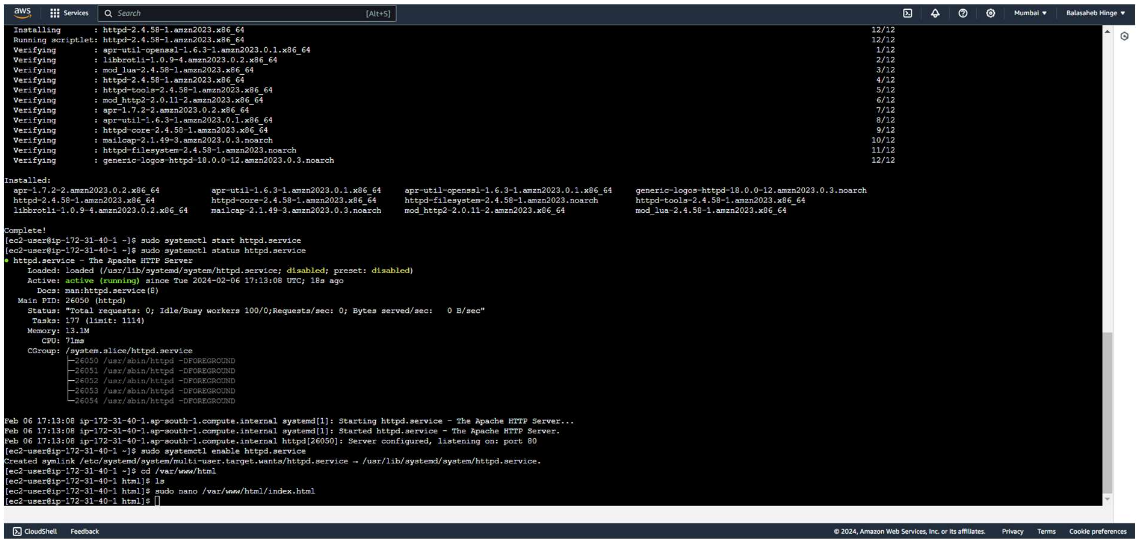
PublicIPs: 13.201.97.108 PrivateIPs: 172.31.40.1

As soon as you hit the above command, apache is installed quickly and you see the above message.

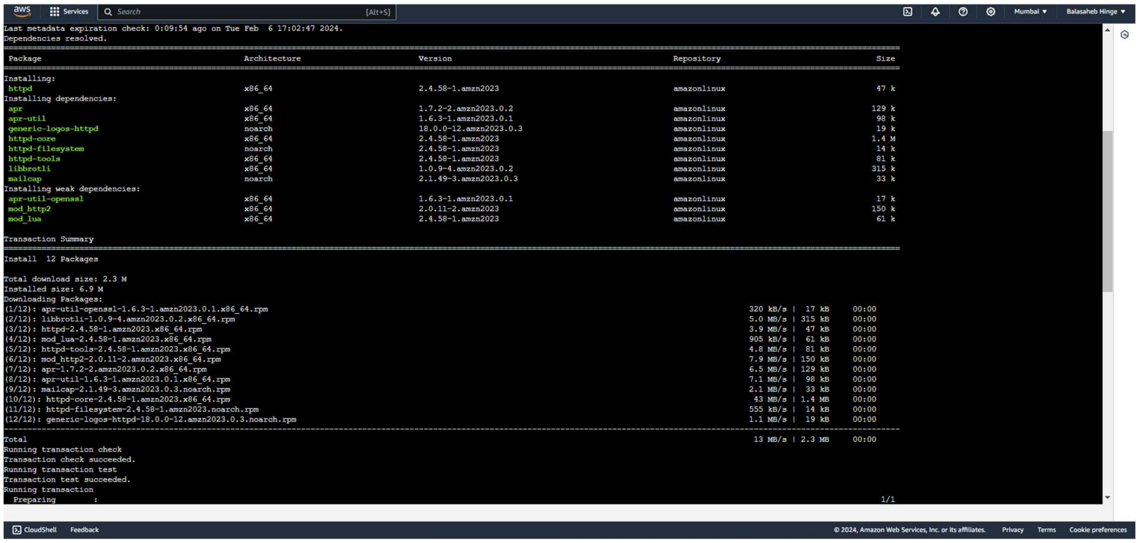
Step 4: Start the Apache Web Server

We are done with the installation. Time to start the **Apache web server**. Fire the below command to start the server.

sudo systemctl start httpd.service



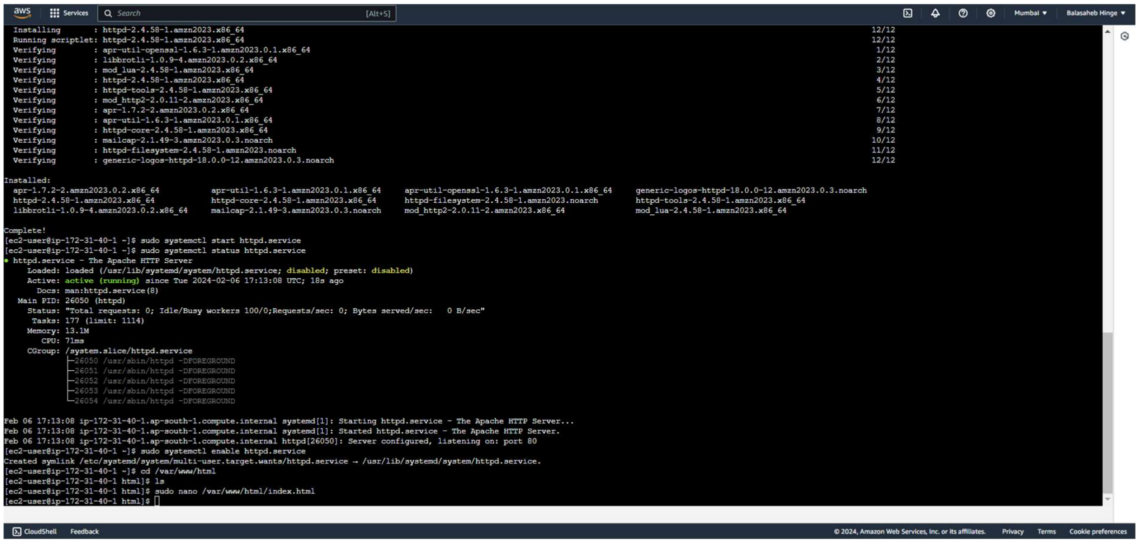
```
  Installing       : httpd-2.4.58-1.amzn2023.x86_64                                            12/12
  Running scriptlet: httpd-2.4.58-1.amzn2023.x86_64                                            12/12
  Verifying        : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64                               1/12
  Verifying        : libbrotli-1.0.9-4.amzn2023.0.2.x86_64                                      2/12
  Verifying        : mod_lua-2.4.58-1.amzn2023.x86_64                                           3/12
  Verifying        : httpd-2.4.58-1.amzn2023.x86_64                                             4/12
  Verifying        : httpd-tools-2.4.58-1.amzn2023.x86_64                                       5/12
  Verifying        : mod_http2-2.0.11-2.amzn2023.x86_64                                         6/12
  Verifying        : apr-1.7.2-2.amzn2023.0.2.x86_64                                            7/12
  Verifying        : apr-util-1.6.3-1.amzn2023.0.1.x86_64                                       8/12
  Verifying        : httpd-core-2.4.58-1.amzn2023.x86_64                                        9/12
  Verifying        : mailcap-2.1.49-3.amzn2023.0.3.noarch                                      10/12
  Verifying        : httpd-filesystem-2.4.58-1.amzn2023.noarch                                 11/12
  Verifying        : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                         12/12

Installed:
  apr-1.7.2-2.amzn2023.0.2.x86_64          apr-util-1.6.3-1.amzn2023.0.1.x86_64     apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64     generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
  httpd-2.4.58-1.amzn2023.x86_64           httpd-core-2.4.58-1.amzn2023.x86_64      httpd-filesystem-2.4.58-1.amzn2023.noarch        httpd-tools-2.4.58-1.amzn2023.x86_64
  libbrotli-1.0.9-4.amzn2023.0.2.x86_64    mailcap-2.1.49-3.amzn2023.0.3.noarch     mod_http2-2.0.11-2.amzn2023.x86_64               mod_lua-2.4.58-1.amzn2023.x86_64

Complete!
[ec2-user@ip-172-31-40-1 ~]$ sudo systemctl start httpd.service
[ec2-user@ip-172-31-40-1 ~]$ sudo systemctl status httpd.service
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: active (running) since Tue 2024-02-06 17:13:08 UTC; 18s ago
       Docs: man:httpd.service(8)
   Main PID: 26050 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 1114)
     Memory: 13.1M
        CPU: 71ms
     CGroup: /system.slice/httpd.service
             ├─26050 /usr/sbin/httpd -DFOREGROUND
             ├─26051 /usr/sbin/httpd -DFOREGROUND
             ├─26052 /usr/sbin/httpd -DFOREGROUND
             ├─26053 /usr/sbin/httpd -DFOREGROUND
             └─26054 /usr/sbin/httpd -DFOREGROUND

Feb 06 17:13:08 ip-172-31-40-1.ap-south-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Feb 06 17:13:08 ip-172-31-40-1.ap-south-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Feb 06 17:13:08 ip-172-31-40-1.ap-south-1.compute.internal httpd[26050]: Server configured, listening on: port 80
[ec2-user@ip-172-31-40-1 ~]$ sudo systemctl enable httpd.service
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-40-1 ~]$ cd /var/www/html
[ec2-user@ip-172-31-40-1 html]$ ls
[ec2-user@ip-172-31-40-1 html]$ sudo nano /var/www/html/index.html
[ec2-user@ip-172-31-40-1 html]$ 
```



```
Last metadata expiration check: 0:09:54 ago on Tue Feb  6 17:02:47 2024.
Dependencies resolved.
==============================================================================================================
 Package                     Architecture         Version                              Repository        Size
==============================================================================================================
Installing:
 httpd                       x86_64               2.4.58-1.amzn2023                    amazonlinux       47 k
Installing dependencies:
 apr                         x86_64               1.7.2-2.amzn2023.0.2                 amazonlinux      129 k
 apr-util                    x86_64               1.6.3-1.amzn2023.0.1                 amazonlinux       98 k
 generic-logos-httpd         noarch               18.0.0-12.amzn2023.0.3               amazonlinux       19 k
 httpd-core                  x86_64               2.4.58-1.amzn2023                    amazonlinux      1.4 M
 httpd-filesystem            noarch               2.4.58-1.amzn2023                    amazonlinux       14 k
 httpd-tools                 x86_64               2.4.58-1.amzn2023                    amazonlinux       81 k
 libbrotli                   x86_64               1.0.9-4.amzn2023.0.2                 amazonlinux      315 k
 mailcap                     noarch               2.1.49-3.amzn2023.0.3                amazonlinux       33 k
Installing weak dependencies:
 apr-util-openssl            x86_64               1.6.3-1.amzn2023.0.1                 amazonlinux       17 k
 mod_http2                   x86_64               2.0.11-2.amzn2023                    amazonlinux      150 k
 mod_lua                     x86_64               2.4.58-1.amzn2023                    amazonlinux       61 k

Transaction Summary
==============================================================================================================
Install  12 Packages

Total download size: 2.3 M
Installed size: 6.9 M
Downloading Packages:
(1/12): apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64.rpm                         320 kB/s |  17 kB     00:00
(2/12): libbrotli-1.0.9-4.amzn2023.0.2.x86_64.rpm                                5.0 MB/s | 315 kB     00:00
(3/12): httpd-2.4.58-1.amzn2023.x86_64.rpm                                       3.9 MB/s |  47 kB     00:00
(4/12): mod_lua-2.4.58-1.amzn2023.x86_64.rpm                                     905 kB/s |  61 kB     00:00
(5/12): httpd-tools-2.4.58-1.amzn2023.x86_64.rpm                                 4.8 MB/s |  81 kB     00:00
(6/12): mod_http2-2.0.11-2.amzn2023.x86_64.rpm                                   7.9 MB/s | 150 kB     00:00
(7/12): apr-1.7.2-2.amzn2023.0.2.x86_64.rpm                                      6.5 MB/s | 129 kB     00:00
(8/12): apr-util-1.6.3-1.amzn2023.0.1.x86_64.rpm                                 7.1 MB/s |  98 kB     00:00
(9/12): mailcap-2.1.49-3.amzn2023.0.3.noarch.rpm                                 2.1 MB/s |  33 kB     00:00
(10/12): httpd-core-2.4.58-1.amzn2023.x86_64.rpm                                  43 MB/s | 1.4 MB     00:00
(11/12): httpd-filesystem-2.4.58-1.amzn2023.noarch.rpm                           555 kB/s |  14 kB     00:00
(12/12): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm                   1.1 MB/s |  19 kB     00:00
--------------------------------------------------------------------------------------------------------------
Total                                                                             13 MB/s | 2.3 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                      1/1
```

Step 5: Check Apache Service Status

sudo systemctl status httpd.service



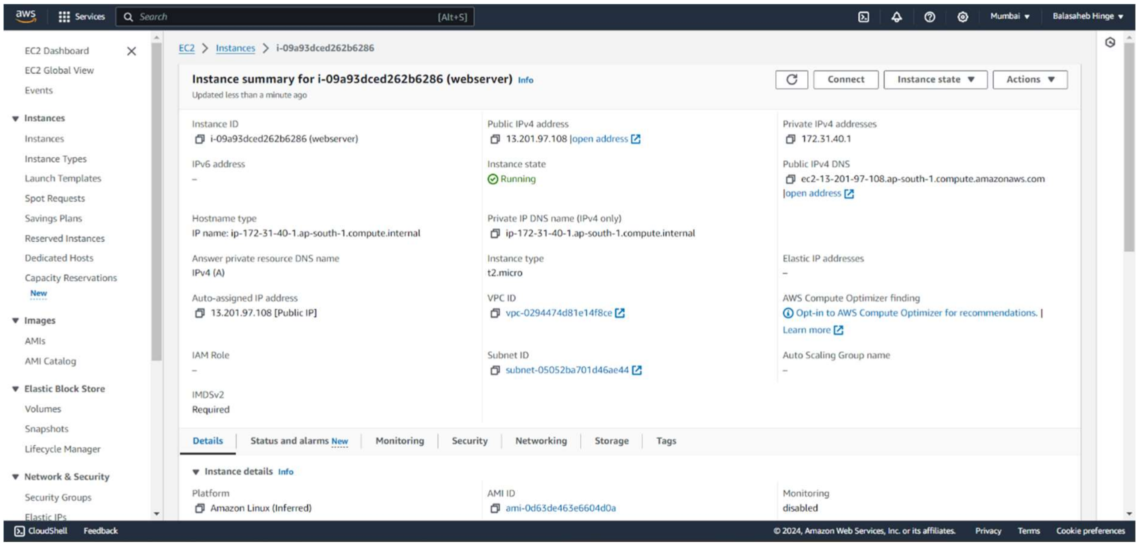
```
  Installing       : httpd-2.4.58-1.amzn2023.x86_64                                                           12/12
  Running scriptlet: httpd-2.4.58-1.amzn2023.x86_64                                                           12/12
  Verifying        : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64                                              1/12
  Verifying        : libbrotli-1.0.9-4.amzn2023.0.2.x86_64                                                     2/12
  Verifying        : mod_lua-2.4.58-1.amzn2023.x86_64                                                          3/12
  Verifying        : httpd-2.4.58-1.amzn2023.x86_64                                                            4/12
  Verifying        : httpd-tools-2.4.58-1.amzn2023.x86_64                                                      5/12
  Verifying        : mod_http2-2.0.11-2.amzn2023.x86_64                                                        6/12
  Verifying        : apr-1.7.2-2.amzn2023.0.2.x86_64                                                           7/12
  Verifying        : apr-util-1.6.3-1.amzn2023.0.1.x86_64                                                      8/12
  Verifying        : httpd-core-2.4.58-1.amzn2023.x86_64                                                       9/12
  Verifying        : mailcap-2.1.49-3.amzn2023.0.3.noarch                                                     10/12
  Verifying        : httpd-filesystem-2.4.58-1.amzn2023.noarch                                                11/12
  Verifying        : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                        12/12

Installed:
  apr-1.7.2-2.amzn2023.0.2.x86_64          apr-util-1.6.3-1.amzn2023.0.1.x86_64     apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64     generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
  httpd-2.4.58-1.amzn2023.x86_64           httpd-core-2.4.58-1.amzn2023.x86_64      httpd-filesystem-2.4.58-1.amzn2023.noarch        httpd-tools-2.4.58-1.amzn2023.x86_64
  libbrotli-1.0.9-4.amzn2023.0.2.x86_64    mailcap-2.1.49-3.amzn2023.0.3.noarch     mod_http2-2.0.11-2.amzn2023.x86_64               mod_lua-2.4.58-1.amzn2023.x86_64

Complete!
[ec2-user@ip-172-31-40-1 ~]$ sudo systemctl start httpd.service
[ec2-user@ip-172-31-40-1 ~]$ sudo systemctl status httpd.service
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: active (running) since Tue 2024-02-06 17:13:08 UTC; 18s ago
       Docs: man:httpd.service(8)
   Main PID: 26050 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 1114)
     Memory: 13.1M
        CPU: 71ms
     CGroup: /system.slice/httpd.service
             ├─26050 /usr/sbin/httpd -DFOREGROUND
             ├─26051 /usr/sbin/httpd -DFOREGROUND
             ├─26052 /usr/sbin/httpd -DFOREGROUND
             ├─26053 /usr/sbin/httpd -DFOREGROUND
             └─26054 /usr/sbin/httpd -DFOREGROUND

Feb 06 17:13:08 ip-172-31-40-1.ap-south-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Feb 06 17:13:08 ip-172-31-40-1.ap-south-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Feb 06 17:13:08 ip-172-31-40-1.ap-south-1.compute.internal httpd[26050]: Server configured, listening on: port 80
[ec2-user@ip-172-31-40-1 ~]$ sudo systemctl enable httpd.service
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-40-1 ~]$ cd /var/www/html
[ec2-user@ip-172-31-40-1 html]$ ls
[ec2-user@ip-172-31-40-1 html]$ sudo nano /var/www/html/index.html
[ec2-user@ip-172-31-40-1 html]$ 
```

Step 6: Verify the Installation

We have installed and started the Apache web server in our **Amazon Linux 2023** instance.

The instance also allows web traffic now, Let’s get the public IP of the instance and see what’s happening.

You can get the public IP of the instance by clicking the instance Id on the instance details screen as shown below.



Instance summary for i-09a93dced262b6286 (webserver)
Updated less than a minute ago

Instance ID
i-09a93dced262b6286 (webserver)

Public IPv4 address
13.201.97.108 |open address

Private IPv4 addresses
172.31.40.1

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-13-201-97-108.ap-south-1.compute.amazonaws.com |open address

Hostname type
IP name: ip-172-31-40-1.ap-south-1.compute.internal

Private IP DNS name (IPv4 only)
ip-172-31-40-1.ap-south-1.compute.internal

Answer private resource DNS name
IPv4 (A)

Instance type
t2.micro

Elastic IP addresses
–

Auto-assigned IP address
13.201.97.108 [Public IP]

VPC ID
vpc-0294474d81e14f8ce

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role
–

Subnet ID
subnet-05052ba701d46ae44

Auto Scaling Group name
–

IMDSv2
Required

Instance details

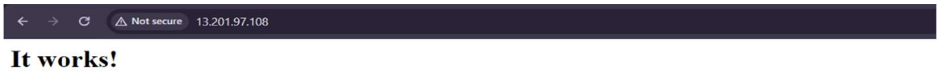
Platform
Amazon Linux (Inferred)

AMI ID
ami-0d63de463e6604d0a

Monitoring
disabled

Hit the public IP.

And I see the below page.



It works!

Step 7: Customize/Add the index.html

As we know that the document root is **/var/www/html**, let’s check the place to see if there is an HTML file.

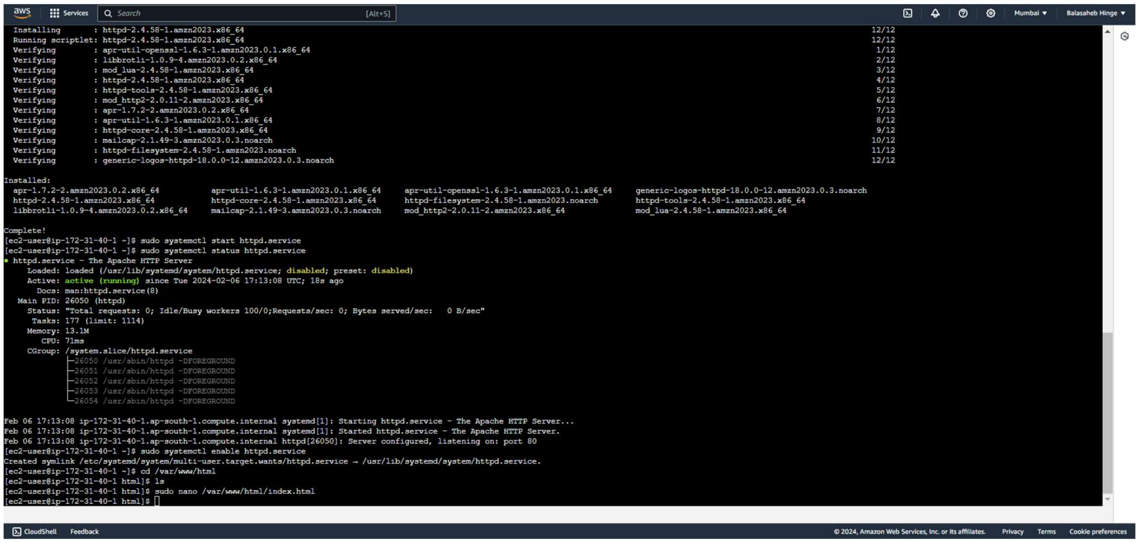
There is no HTML file present in the document root. Let’s add one.

DocumentRoot: Document Root is the directory from which Apache looks for and serves web files on your request. So we will create an index.html in **/var/www/html** folder

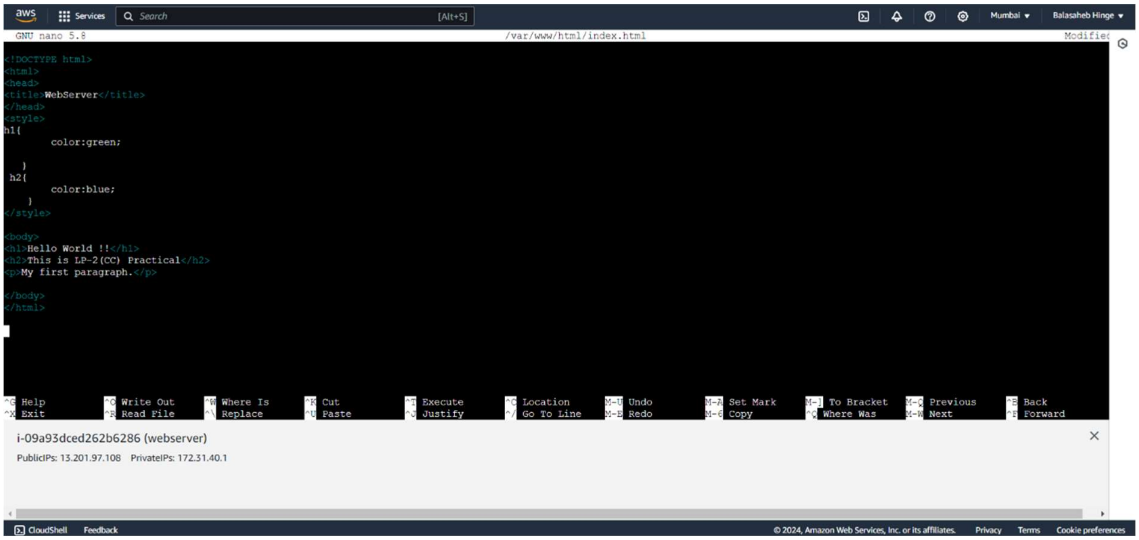
Create index.html

I used ***sudo nano*** command to create a file.

sudo nano /var/www/html/index.html



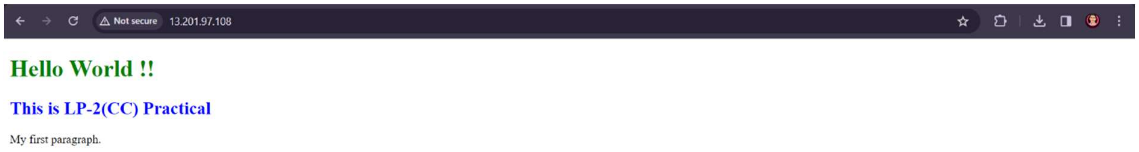
```
  Installing       : httpd-2.4.58-1.amzn2023.x86_64                                                                    12/12
  Running scriptlet: httpd-2.4.58-1.amzn2023.x86_64                                                                    12/12
  Verifying        : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64                                                       1/12
  Verifying        : libbrotli-1.0.9-4.amzn2023.0.2.x86_64                                                              2/12
  Verifying        : mod_lua-2.4.58-1.amzn2023.x86_64                                                                   3/12
  Verifying        : httpd-2.4.58-1.amzn2023.x86_64                                                                     4/12
  Verifying        : httpd-tools-2.4.58-1.amzn2023.x86_64                                                               5/12
  Verifying        : mod_http2-2.0.11-2.amzn2023.x86_64                                                                 6/12
  Verifying        : apr-1.7.2-2.amzn2023.0.2.x86_64                                                                    7/12
  Verifying        : apr-util-1.6.3-1.amzn2023.0.1.x86_64                                                               8/12
  Verifying        : httpd-core-2.4.58-1.amzn2023.x86_64                                                                9/12
  Verifying        : mailcap-2.1.49-3.amzn2023.0.3.noarch                                                              10/12
  Verifying        : httpd-filesystem-2.4.58-1.amzn2023.noarch                                                         11/12
  Verifying        : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                                 12/12

Installed:
  apr-1.7.2-2.amzn2023.0.2.x86_64          apr-util-1.6.3-1.amzn2023.0.1.x86_64     apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64     generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
  httpd-2.4.58-1.amzn2023.x86_64           httpd-core-2.4.58-1.amzn2023.x86_64      httpd-filesystem-2.4.58-1.amzn2023.noarch        httpd-tools-2.4.58-1.amzn2023.x86_64
  libbrotli-1.0.9-4.amzn2023.0.2.x86_64    mailcap-2.1.49-3.amzn2023.0.3.noarch     mod_http2-2.0.11-2.amzn2023.x86_64               mod_lua-2.4.58-1.amzn2023.x86_64

Complete!
[ec2-user@ip-172-31-40-1 ~]$ sudo systemctl start httpd.service
[ec2-user@ip-172-31-40-1 ~]$ sudo systemctl status httpd.service
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: active (running) since Tue 2024-02-06 17:13:08 UTC; 18s ago
       Docs: man:httpd.service(8)
   Main PID: 26050 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 1114)
     Memory: 13.1M
        CPU: 71ms
     CGroup: /system.slice/httpd.service
             ├─26050 /usr/sbin/httpd -DFOREGROUND
             ├─26051 /usr/sbin/httpd -DFOREGROUND
             ├─26052 /usr/sbin/httpd -DFOREGROUND
             ├─26053 /usr/sbin/httpd -DFOREGROUND
             └─26054 /usr/sbin/httpd -DFOREGROUND

Feb 06 17:13:08 ip-172-31-40-1.ap-south-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Feb 06 17:13:08 ip-172-31-40-1.ap-south-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Feb 06 17:13:08 ip-172-31-40-1.ap-south-1.compute.internal httpd[26050]: Server configured, listening on: port 80
[ec2-user@ip-172-31-40-1 ~]$ sudo systemctl enable httpd.service
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-40-1 ~]$ cd /var/www/html
[ec2-user@ip-172-31-40-1 html]$ ls
[ec2-user@ip-172-31-40-1 html]$ sudo nano /var/www/html/index.html
[ec2-user@ip-172-31-40-1 html]$ 
```



```
  GNU nano 5.8                              /var/www/html/index.html                              Modifie

<!DOCTYPE html>
<html>
<head>
<title>WebServer</title>
</head>
<style>
h1{
        color:green;

   }
 h2{
        color:blue;
    }
</style>

<body>
<h1>Hello World !!</h1>
<h2>This is LP-2(CC) Practical</h2>
<p>My first paragraph.</p>

</body>
</html>

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark   M-] To Bracket M-Q Previous   ^B Back
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy       ^Q Where Was   M-W Next       ^F Forward
```

i-09a93dced262b6286 (webserver)

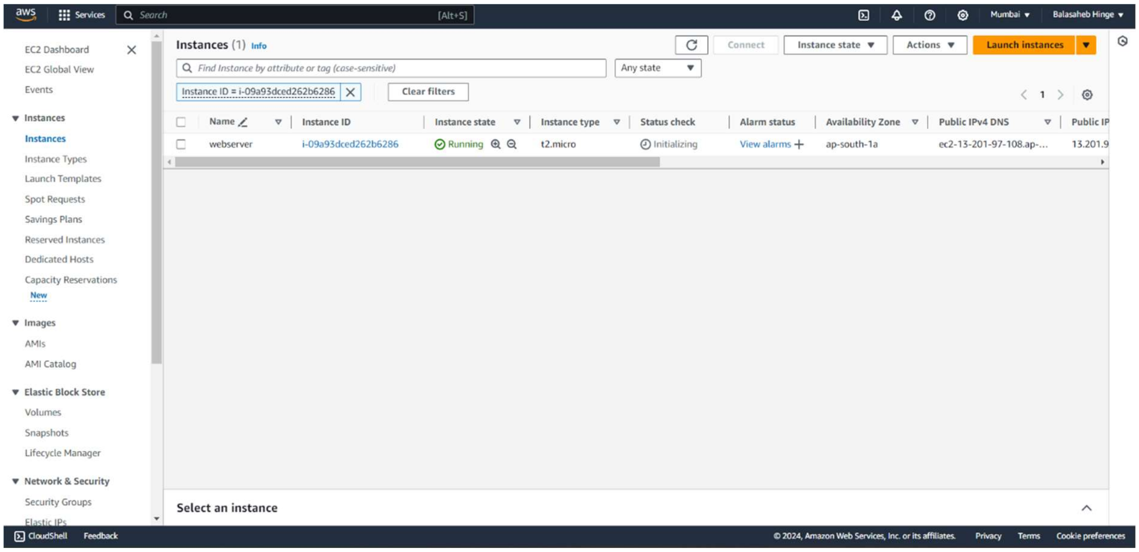
PublicIPs: 13.201.97.108 PrivateIPs: 172.31.40.1



Hello World !!

This is LP-2(CC) Practical

My first paragraph.



Instances (1) Info

Find Instance by attribute or tag (case-sensitive) | Any state

Instance ID = i-09a93dced262b6286

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
□	webserver	i-09a93dced262b6286	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-13-201-97-108.ap-...	13.201.9