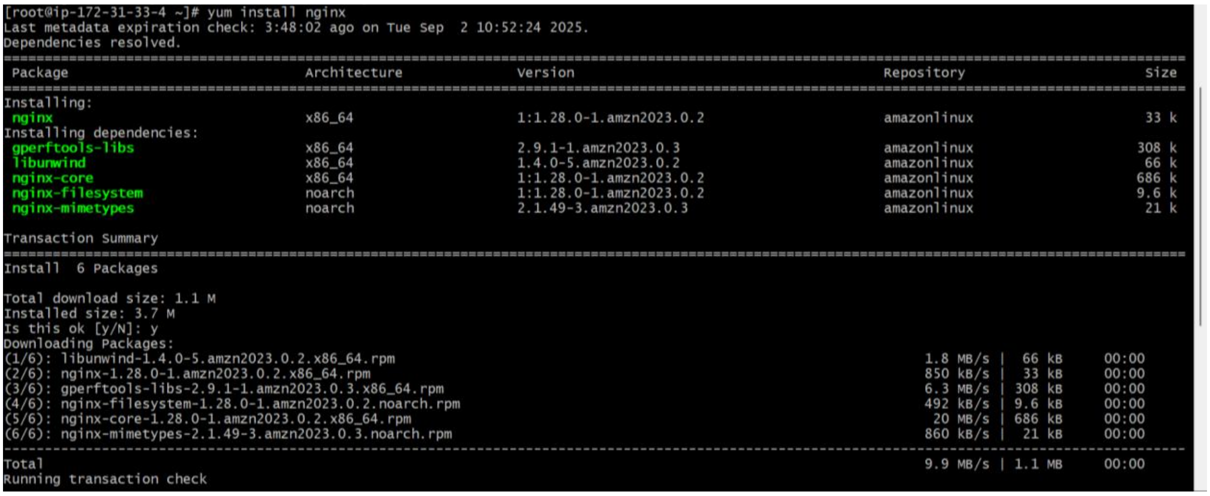
1. INSTALL NGINX AND RUN NGINX ON PORT NUMBER 81.

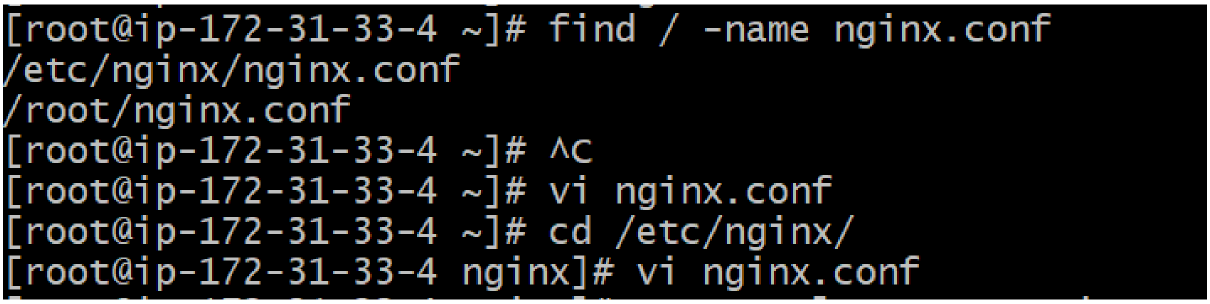


```
[root@ip-172-31-33-4 ~]# yum install nginx
Last metadata expiration check: 3:48:02 ago on Tue Sep  2 10:52:24 2025.
Dependencies resolved.
================================================================================
 Package              Architecture      Version                       Repository        Size
================================================================================
Installing:
 nginx                x86_64            1:1.28.0-1.amzn2023.0.2       amazonlinux       33 k
Installing dependencies:
 gperftools-libs      x86_64            2.9.1-1.amzn2023.0.3          amazonlinux      308 k
 libunwind            x86_64            1.4.0-5.amzn2023.0.2          amazonlinux       66 k
 nginx-core           x86_64            1:1.28.0-1.amzn2023.0.2       amazonlinux      686 k
 nginx-filesystem     noarch            1:1.28.0-1.amzn2023.0.2       amazonlinux      9.6 k
 nginx-mimetypes      noarch            2.1.49-3.amzn2023.0.3         amazonlinux       21 k

Transaction Summary
================================================================================
Install  6 Packages

Total download size: 1.1 M
Installed size: 3.7 M
Is this ok [y/N]: y
Downloading Packages:
(1/6): libunwind-1.4.0-5.amzn2023.0.2.x86_64.rpm            1.8 MB/s |  66 kB     00:00
(2/6): nginx-1.28.0-1.amzn2023.0.2.x86_64.rpm               850 kB/s |  33 kB     00:00
(3/6): gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64.rpm      6.3 MB/s | 308 kB     00:00
(4/6): nginx-filesystem-1.28.0-1.amzn2023.0.2.noarch.rpm    492 kB/s | 9.6 kB     00:00
(5/6): nginx-core-1.28.0-1.amzn2023.0.2.x86_64.rpm           20 MB/s | 686 kB     00:00
(6/6): nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm     860 kB/s |  21 kB     00:00
--------------------------------------------------------------------------------
Total                                                       9.9 MB/s | 1.1 MB     00:00
Running transaction check
```

- install nginx service in linux machine : yum install nginx

```
[root@ip-172-31-33-4 ~]# systemctl start nginx
[root@ip-172-31-33-4 ~]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
     Active: active (running) since Tue 2025-09-02 14:49:54 UTC; 3s ago
    Process: 38058 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 38059 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 38060 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 38061 (nginx)
      Tasks: 3 (limit: 1057)
     Memory: 3.2M
        CPU: 59ms
     CGroup: /system.slice/nginx.service
             ├─38061 "nginx: master process /usr/sbin/nginx"
             ├─38062 "nginx: worker process"
             └─38063 "nginx: worker process"

Sep 02 14:49:54 ip-172-31-33-4.eu-north-1.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Sep 02 14:49:54 ip-172-31-33-4.eu-north-1.compute.internal nginx[38059]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Sep 02 14:49:54 ip-172-31-33-4.eu-north-1.compute.internal nginx[38059]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Sep 02 14:49:54 ip-172-31-33-4.eu-north-1.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[root@ip-172-31-33-4 ~]#
```

- start the nginx service : systemctl start nginx



```
[root@ip-172-31-33-4 ~]# find / -name nginx.conf
/etc/nginx/nginx.conf
/root/nginx.conf
[root@ip-172-31-33-4 ~]# ^C
[root@ip-172-31-33-4 ~]# vi nginx.conf
[root@ip-172-31-33-4 ~]# cd /etc/nginx/
[root@ip-172-31-33-4 nginx]# vi nginx.conf
```

- find the file : file / -name nginx.conf
- go to the directory : cd /etc/nginx/
- view the file contents : vi nginx.conf

```
root@ip-172-31-33-4:/etc/nginx
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    keepalive_timeout   65;
    types_hash_max_size 4096;

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;

    # Load modular configuration files from the /etc/nginx/conf.d directory.
    # See http://nginx.org/en/docs/ngx_core_module.html#include
    # for more information.
    include /etc/nginx/conf.d/*.conf;

    server {
        listen       81;
        listen       [::]:80;
        server_name  _;
        root         /usr/share/nginx/html;

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;
-- INSERT --
```

- in nginx.conf change : LISTEN 80 to 81 by using ‘i’ – insert

- Enter : esc :wq! – to save and exit the file

```
[root@ip-172-31-33-4 nginx]# systemctl restart nginx
[root@ip-172-31-33-4 nginx]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
     Active: active (running) since Tue 2025-09-02 15:02:43 UTC; 16s ago
    Process: 38445 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 38447 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 38448 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 38449 (nginx)
      Tasks: 3 (limit: 1057)
     Memory: 3.2M
        CPU: 58ms
     CGroup: /system.slice/nginx.service
             ├─38449 "nginx: master process /usr/sbin/nginx"
             ├─38450 "nginx: worker process"
             └─38451 "nginx: worker process"
```

- Restart the service : systemctl restart nginx

- Now got o web and enter the – “ipv 4 publicadress: 81” (note : you can get the ip v4 adress in aws instance that you have launched before)
- You can see the nginx is running in port 81.

2. . DEPLOYING THE SAMPLE HTML FILE IN NGINX PORT 82

```
[root@ip-172-31-39-167 ~]# find / -name nginx.conf
/etc/nginx/nginx.conf
[root@ip-172-31-39-167 ~]# cd /etc/nginx/
[root@ip-172-31-39-167 nginx]# vi nginx.conf
[root@ip-172-31-39-167 nginx]# cd ~
[root@ip-172-31-39-167 ~]# cd /var/www/html
[root@ip-172-31-39-167 html]# vi index.html
```

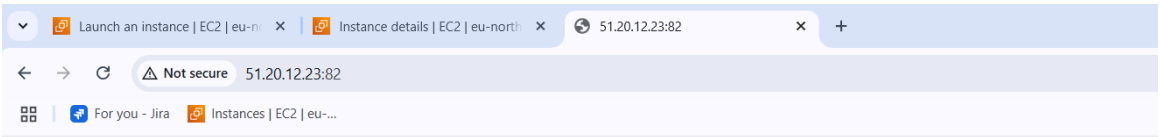
- Go to the html directory : cd /var/www/html/
- View the file : vi index.html

```
<h1> httpd running in port82 </h1>
~
~
~
~
~
~
~
~
~
~
~
~
```

- Enter the text or content you want o view on web page .
- Save by entering esc button and enter :wq! – to save and exit

```
[root@ip-172-31-39-167 html]# vi index.html
[root@ip-172-31-39-167 html]# systemctl restart nginx
[root@ip-172-31-39-167 html]#
```

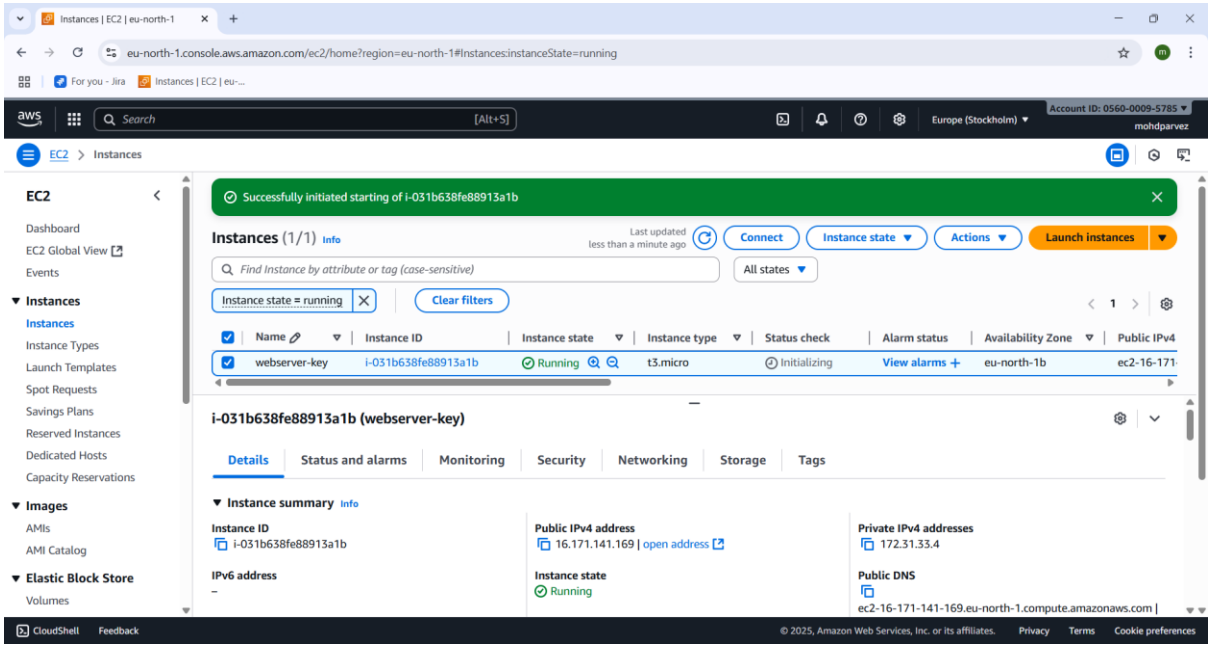
- Restart the service – systemctl restart nginx



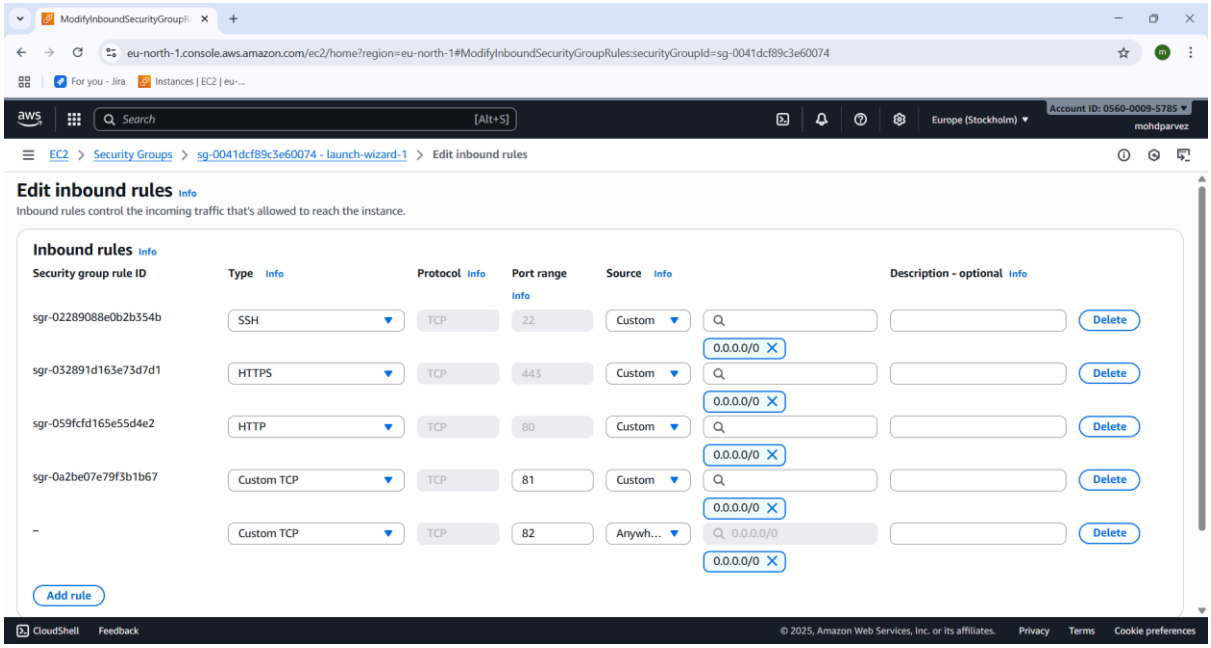
httpd running in port82

- Got to the web BROWSER and enter “ipv 4 publicadress: 81”
- You see the deployed html file is reflecting in the web page.

3. HOW TO INSTALL AND RUN HTTPD IN PORT 82

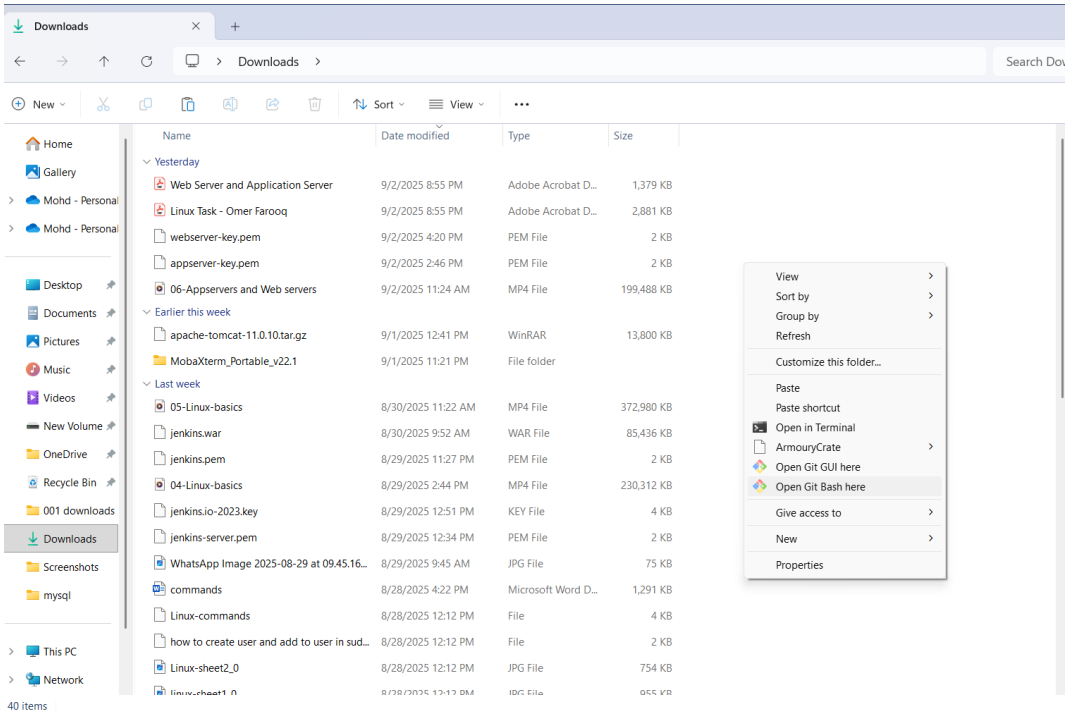


→ Lauch aws ec2 instance .

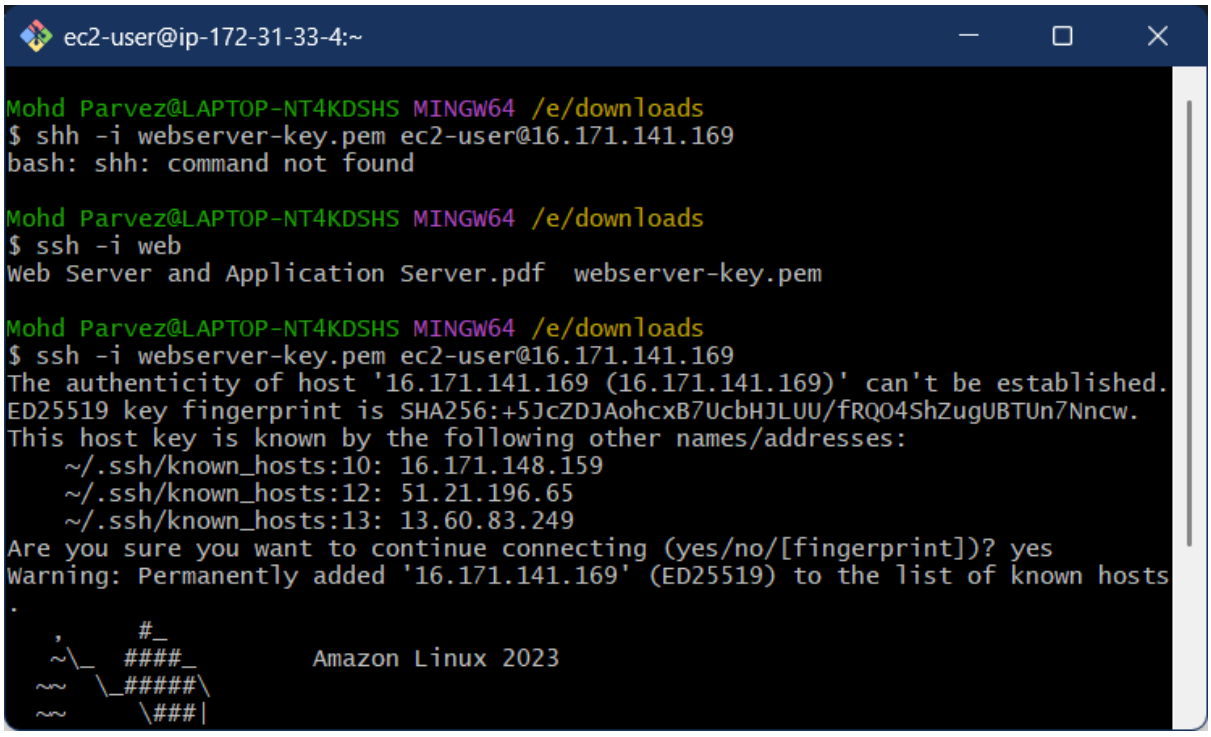


→ Go to security groups and click on edit inbound rules

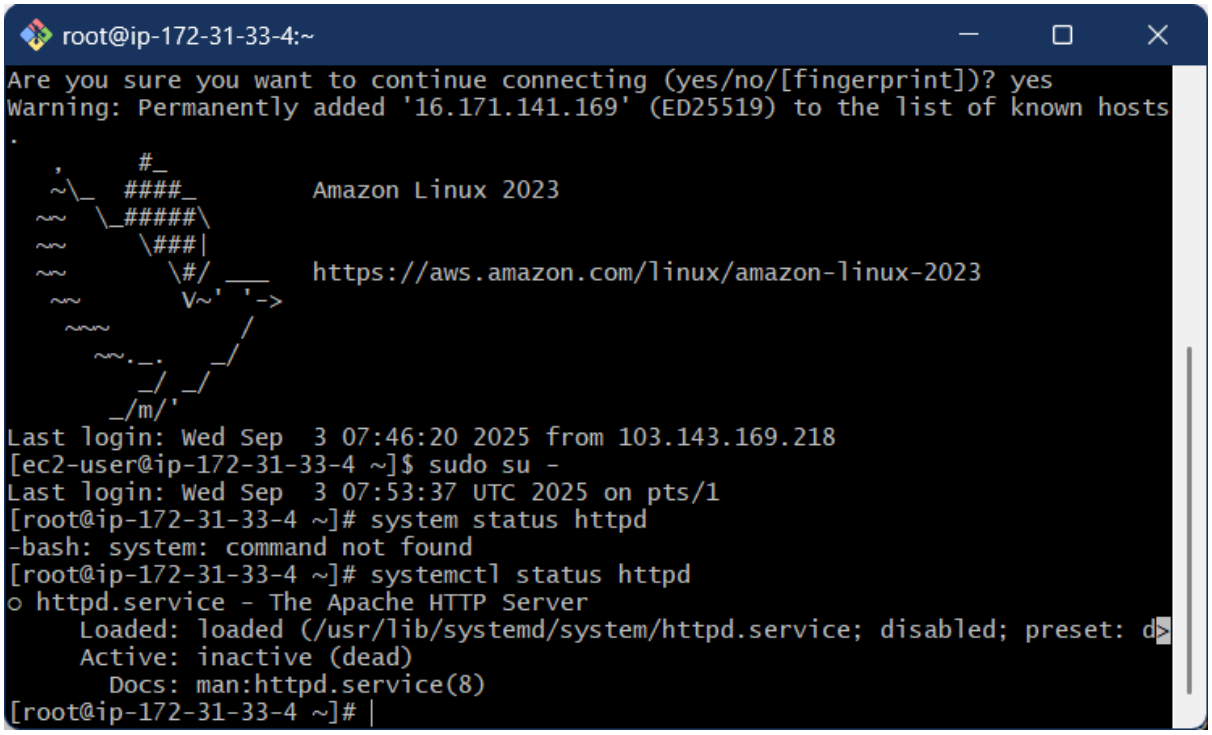
→ Add security group with custom tcp and give port no 82 and save.



→ Go to the location where keypar.pem is downloaded and right click and select open git bash here.

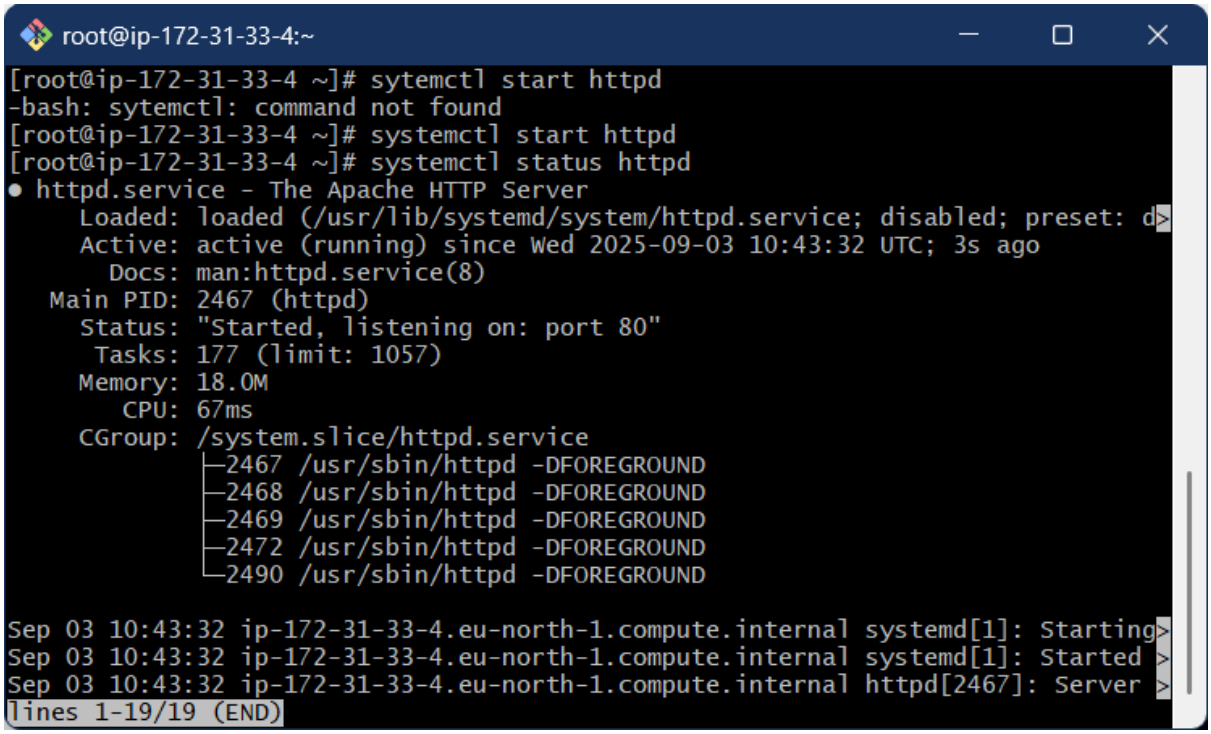


→ Connect to the remote machine by using : ssh -i “keypair.pem” ec2-user@ “ip_v4 _public address”.

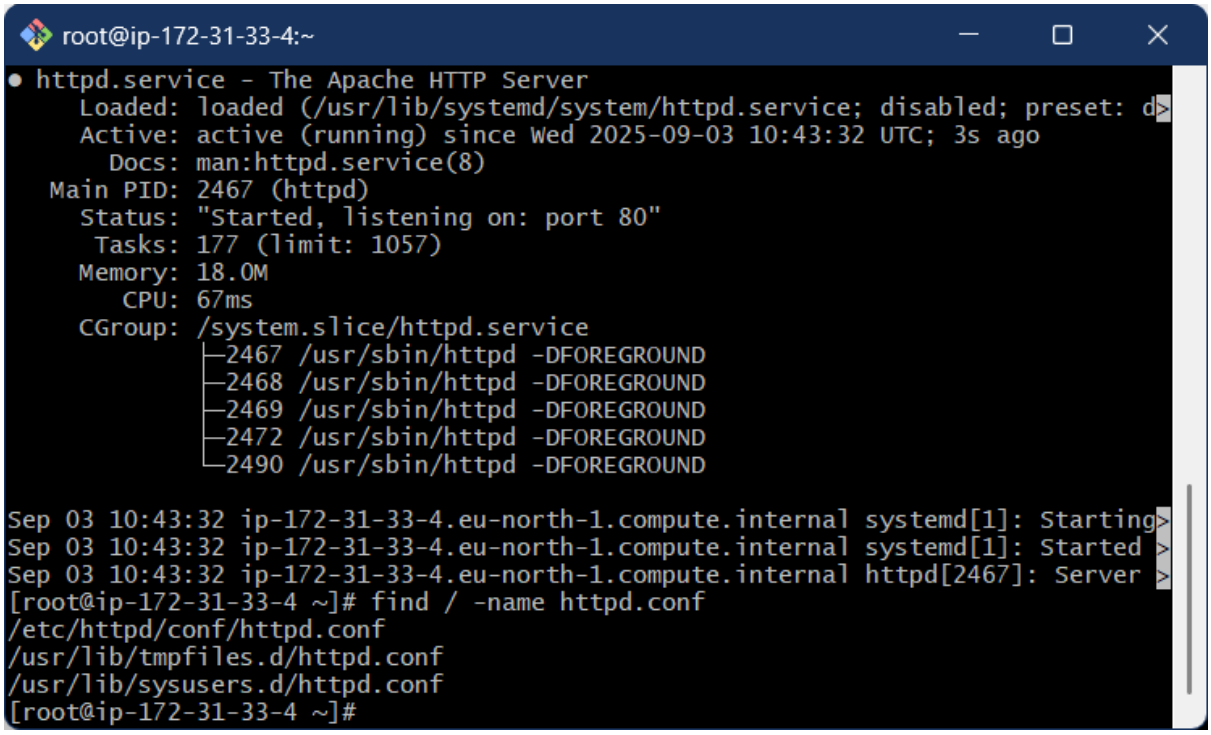


→ Install the httpd (apache) service : yum install httpd

→ Check the status of httpd (apache) : systemctl status httpd

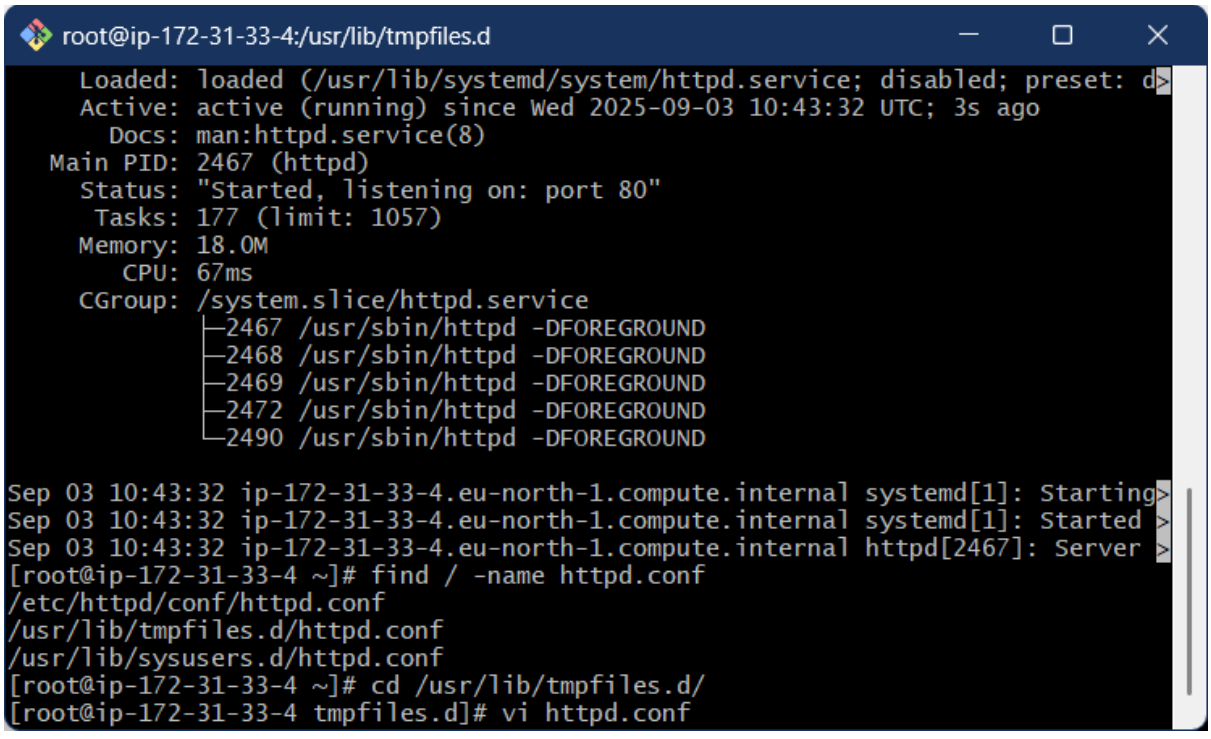


→ Start the service if it is inactive : systemctl start httpd



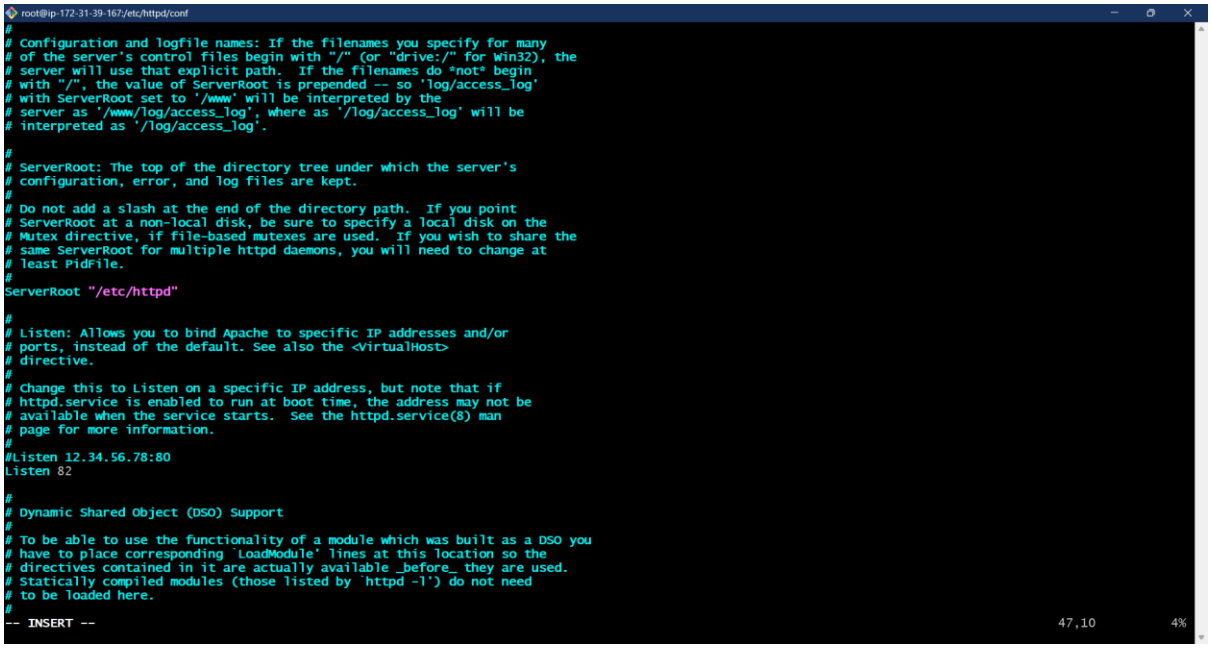
→ Find the configuration file using : find / -name httpd.conf

→ Go to the directory where the httpd. Conf file is located using cd command

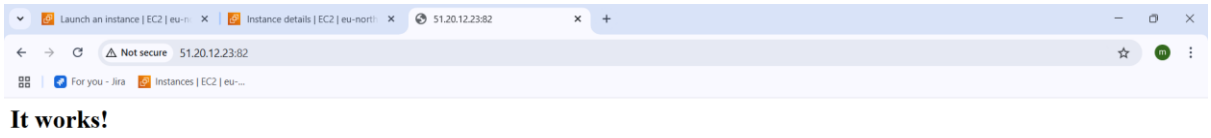


→ Cd /usr/lib/tmpfiles.d/ : going ton the directory where the httpd.conf file is present

→ See the content in the file : vi “filename”.conf



→ After entering in to file look LISTEN 80 and the change 80 to 82 by using “i” command
→ Save it by entering esc button :wq!
→ Run command : systemctl restart httpd
→



→ Copy the ipv4 adress and enter : “ipv4 public address” :82
→ You can see the the httpd is running on port 82.

4. Deploying sample index.html on httpd:

```
[root@ip-172-31-39-167 ~]# vi httpd.conf
[root@ip-172-31-39-167 ~]# systemctl stop httpd
[root@ip-172-31-39-167 ~]# systemctl start httpd
[root@ip-172-31-39-167 ~]# find / -name index.html
/usr/lib/python3.9/site-packages/awscli/customizations/sso/index.html
/usr/share/doc/oniguruma/index.html
/usr/share/doc/cyrus-sasl-lib/index.html
/usr/share/doc/python3-jinja2/html/index.html
/usr/share/httpd/noindex/index.html
[root@ip-172-31-39-167 ~]# find / -name html
/var/www/html
/usr/lib64/python3.9/html
/usr/share/doc/python3-jinja2/html
[root@ip-172-31-39-167 ~]# cd /var/www/html
[root@ip-172-31-39-167 html]#
```

→ Go to the html directory where the index.html file is present
→ Cd /var/www/html

```
[root@ip-172-31-39-167 html]# vi index.html
[root@ip-172-31-39-167 html]#
```

→ Enter : vi index.html – to vie the file contents

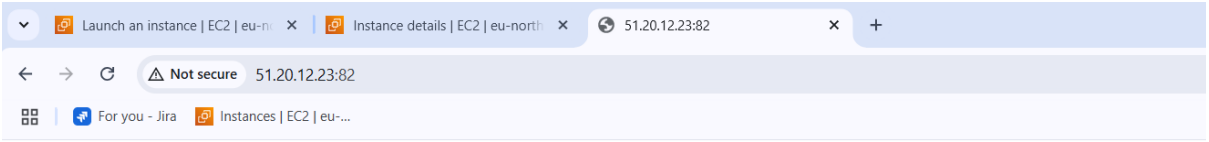
```
<h1> httpd running in port82 </h1>
~
~
~
~
~
~
~
~
~
~
~
~
```

→ In the file enter the out put you want

→ And save it by entering esc button :wq!

```
[root@ip-172-31-39-167 html]# systemctl restart httpd
[root@ip-172-31-39-167 html]#
```

→ Systemctl restart httpd

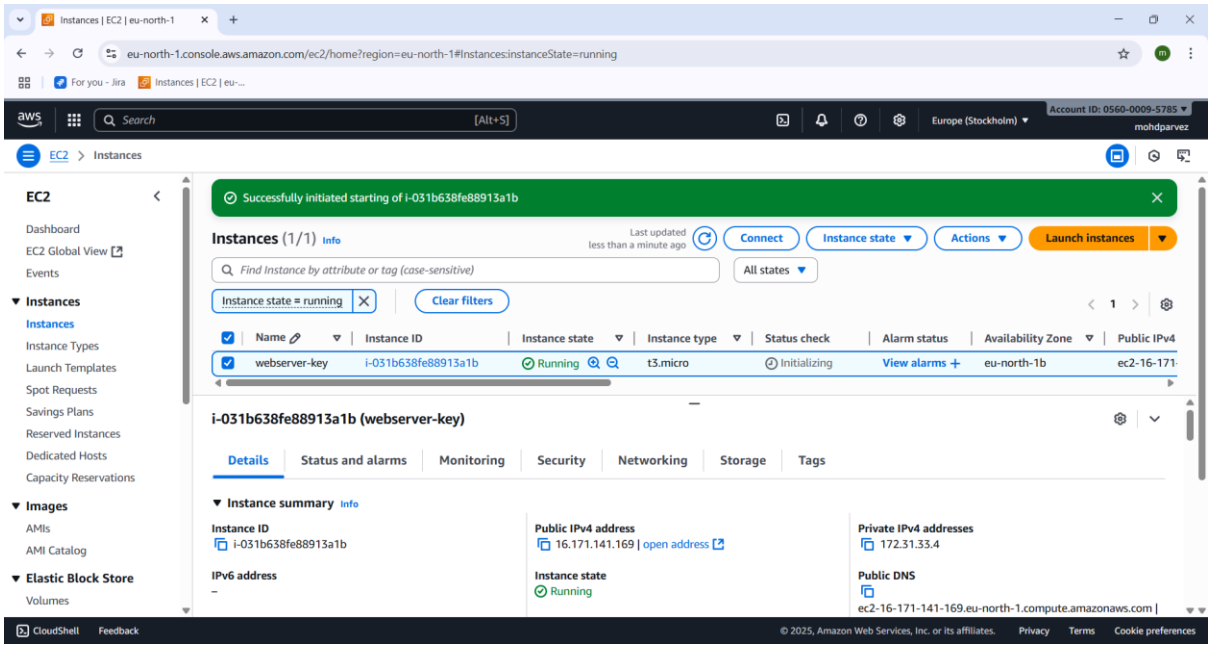


httpd running in port82

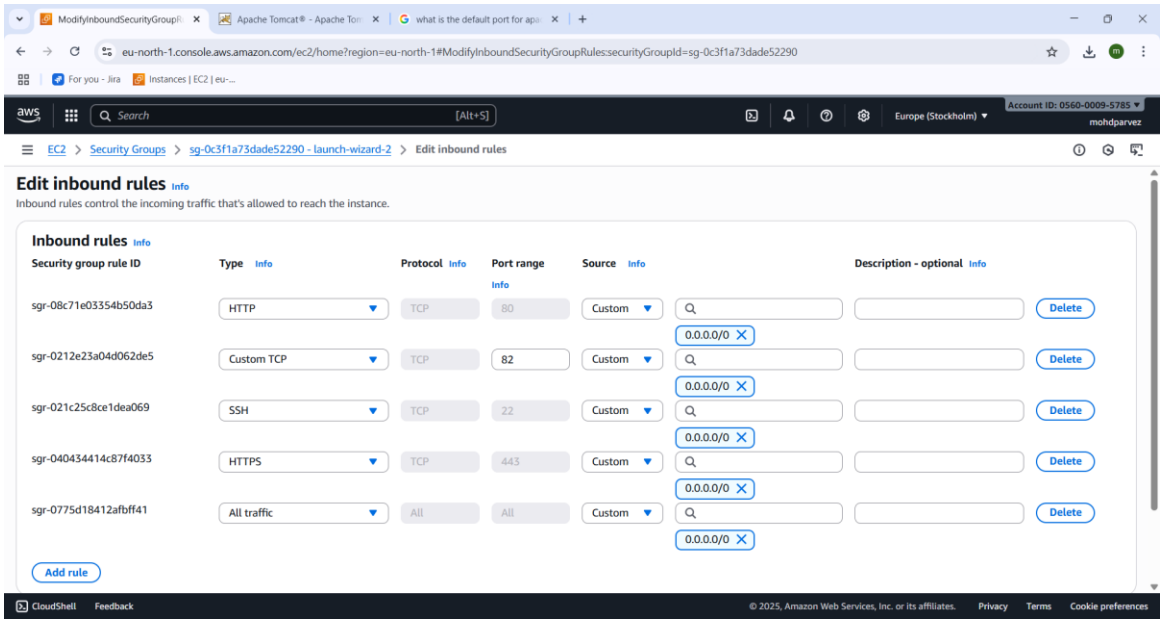
→ ENTER : AWS ipv4 adress and enter port : “ipv4 public address” :82

→ You can see the page with the index.html I have provided.

5. APACHE TOMCAT ON PORT NUMBER 8082

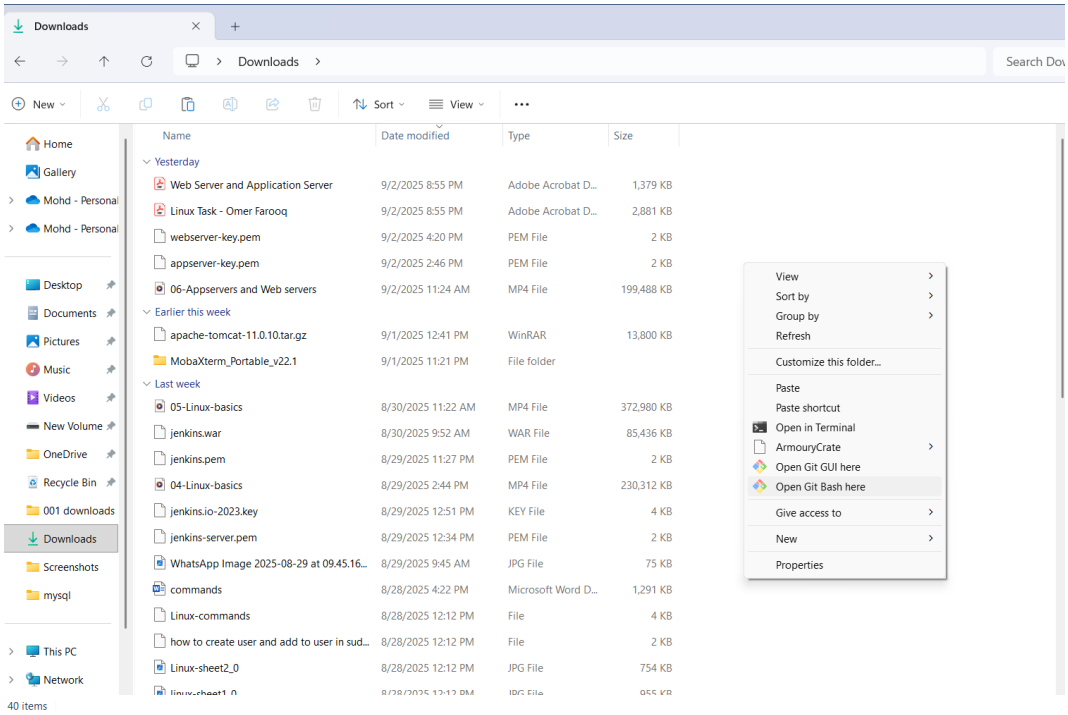


→ Lauch aws ec2 instance .

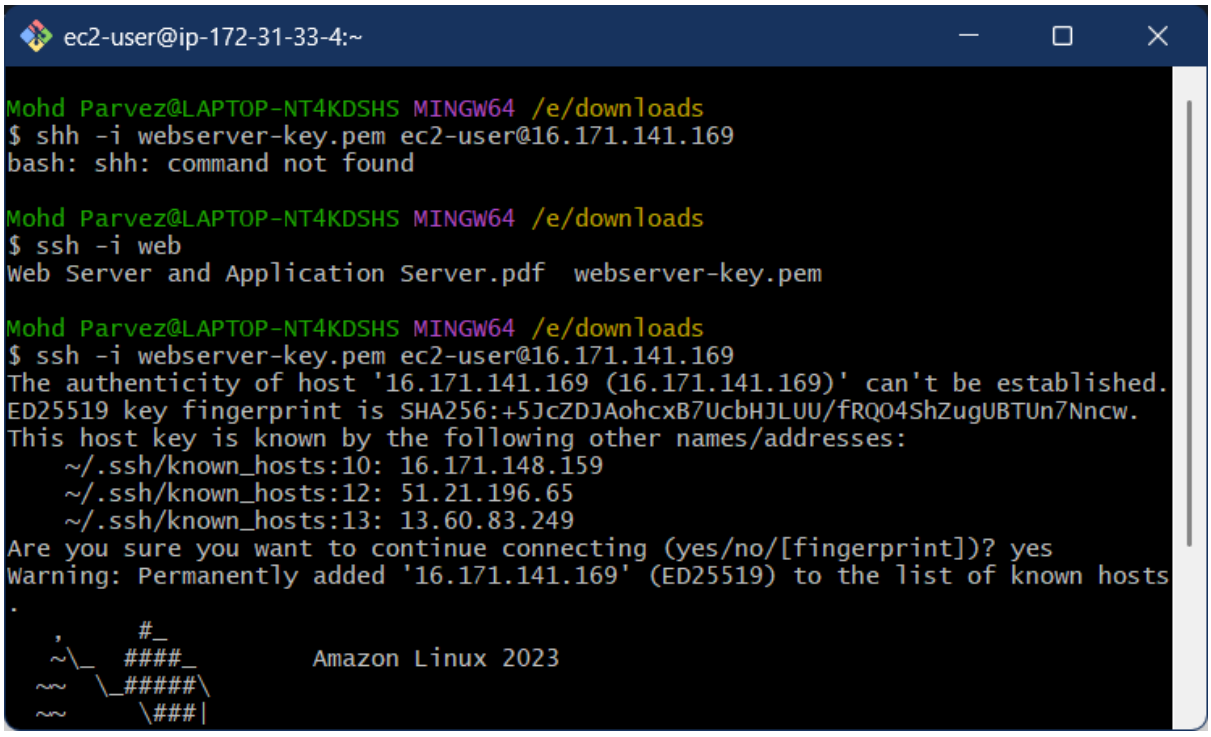


→ Go to security groups and click on edit inbound rules

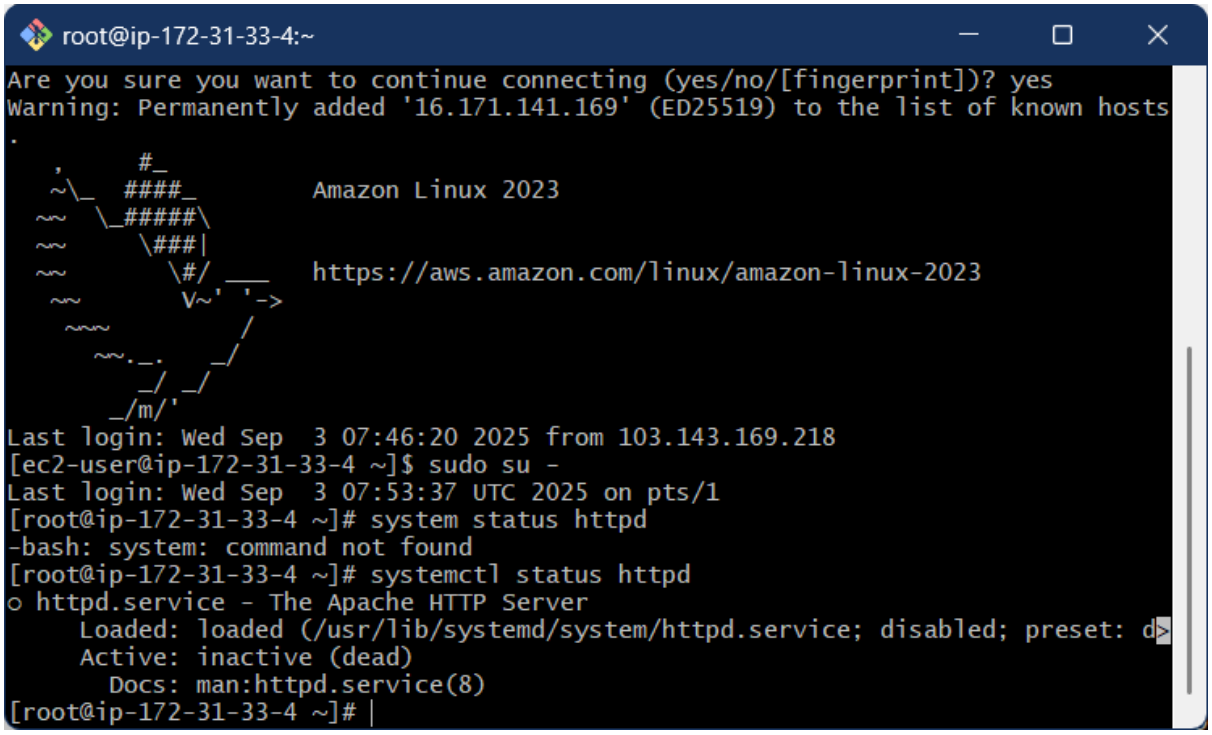
→ Add security group with custom tcp and give port no all and save.



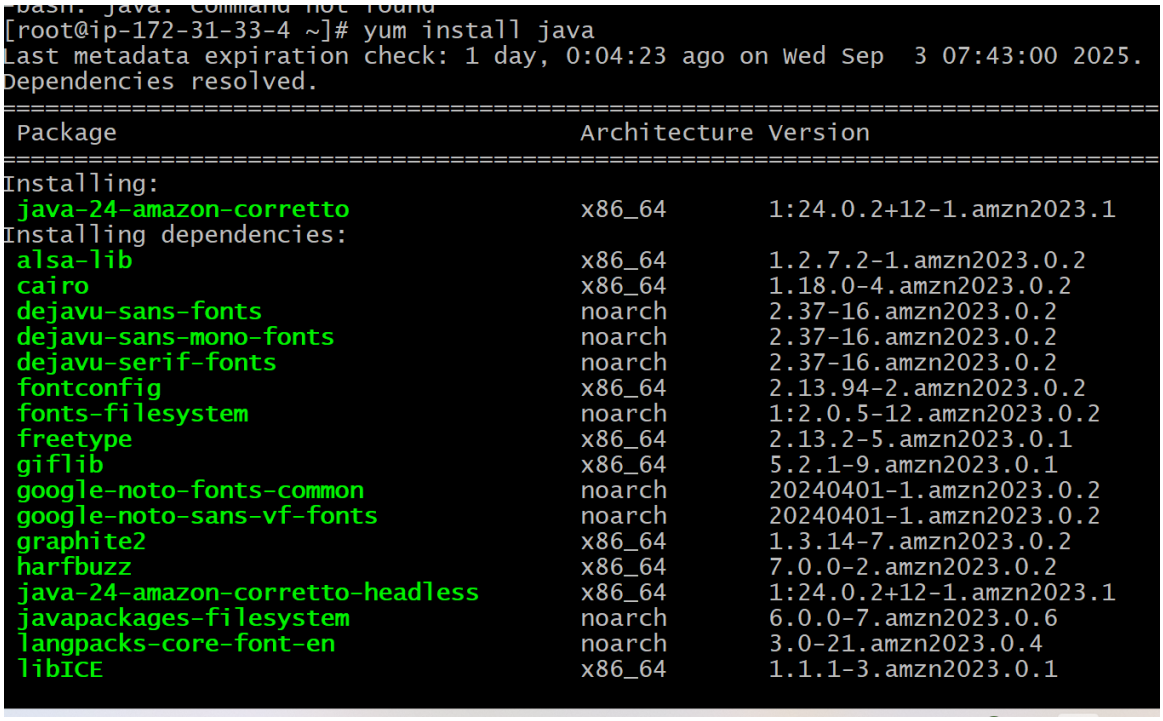
→ Go to the location where keypar.pem is downloaded and right click and select open git bash here.



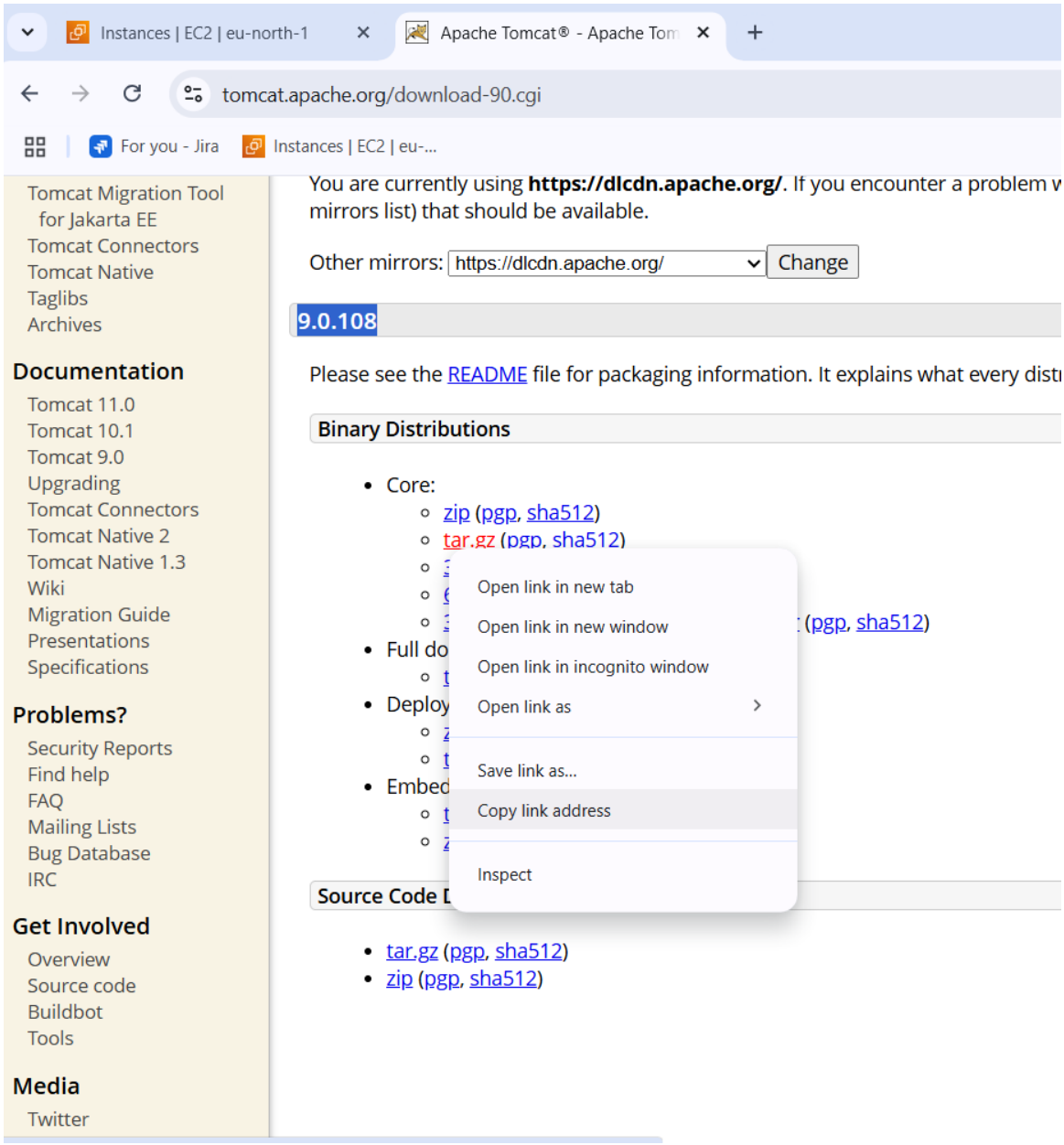
→ Connect to the remote machine by using : ssh -i “keypair.pem” ec2-user@ “ip_v4 _public address”.



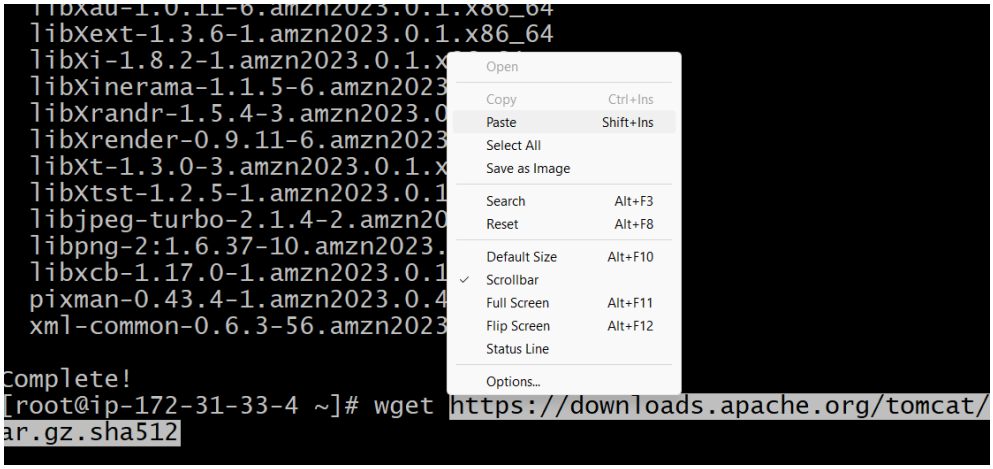
- Here you can see the aws ec2 remote machine has been connected .
- Switch to root user to not get any issues regarding the permission: sudo su –



- Install java because the java is the soure or language that is used for tomcat is creted using java script or language
- Yum install java.



- Every aaplication is cannot be installed by “yum”
- Go to google and search for apache tomcat download select version copy the link of the download file that is zip or tar.



- Enter wget “url” that uou cpied and enter to download the tomcat.

```
[root@ip-172-31-33-4 opt]# wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.
.gz
--2025-09-04 07:58:36--  https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.108/bin
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connec
HTTP request sent, awaiting response... 200 OK
Length: 13028093 (12M) [application/x-gzip]
Saving to: ‘apache-tomcat-9.0.108.tar.gz’

apache-tomcat-9.0.108.tar.g 100%[=========================================>]

2025-09-04 07:58:36 (228 MB/s) - ‘apache-tomcat-9.0.108.tar.gz’ saved [1302809

[root@ip-172-31-33-4 opt]#
```

- here you can see the file has been downloaded.
- Now you need to extract the file

```
[root@ip-172-31-33-4 opt]# ll
total 12724
-rw-r--r--. 1 root root 13028093 Jul 31 18:33 apache-tomcat-9.0.108.tar.gz
drwxr-xr-x. 4 root root       33 Aug 13 21:08 aws
[root@ip-172-31-33-4 opt]# tar xvf apache-tomcat-9.0.108.tar.gz
apache-tomcat-9.0.108/conf/
apache-tomcat-9.0.108/conf/catalina.policy
apache-tomcat-9.0.108/conf/catalina.properties
apache-tomcat-9.0.108/conf/context.xml
```

- ll – to list the files
- tar xvf “file name” .gz : to extract the tomcat file.

```
[root@ip-172-31-33-4 opt]# cd apache-tomcat-9.0.108/
[root@ip-172-31-33-4 apache-tomcat-9.0.108]# ls
BUILDING.txt     LICENSE  README.md      RUNNING.txt  conf  logs  webapps
CONTRIBUTING.md  NOTICE   RELEASE-NOTES  bin          lib   temp  work
[root@ip-172-31-33-4 apache-tomcat-9.0.108]# cd bin/
[root@ip-172-31-33-4 bin]# ls
bootstrap.jar       commons-daemon-native.tar.gz  digest.sh         shutdown.sh          tool-wrapper.sh
catalina-tasks.xml  commons-daemon.jar            makebase.bat      startup.bat          version.bat
catalina.bat        configtest.bat                makebase.sh       startup.sh           version.sh
catalina.sh         configtest.sh                 setclasspath.bat  tomcat-juli.jar
ciphers.bat         daemon.sh                     setclasspath.sh   tomcat-native.tar.gz
ciphers.sh          digest.bat                    shutdown.bat      tool-wrapper.bat
[root@ip-172-31-33-4 bin]#
```

- go to the tomcat directory where it is downloaded : cd “filename”/
- ls – to list the files you can find the configuration files of tomcat.
- Go to bin directory : cd bin/
- Ls - list the files you can find the startup.sh , sh- means shell & .bat is for windows

```
[root@ip-172-31-33-4 bin]# bash ./startup.sh
Using CATALINA_BASE:   /opt/apache-tomcat-9.0.108
Using CATALINA_HOME:   /opt/apache-tomcat-9.0.108
Using CATALINA_TMPDIR: /opt/apache-tomcat-9.0.108/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/apache-tomcat-9.0.108/bin/b
jar
Using CATALINA_OPTS:
Tomcat started.
[root@ip-172-31-33-4 bin]# |
```

- Startb the tomcat : bash ./startup.sh – to start the service.
- Note : systemctl start cannot be used for every application

```
[root@ip-172-31-33-4 bin]# netstat -na | grep 8080
tcp6       0      0 :::8080                 :::*                    LISTEN
[root@ip-172-31-33-4 bin]# ps -ef | grep 8080
root        3311    1899  0 08:09 pts/1    00:00:00 grep --color=auto 8080
[root@ip-172-31-33-4 bin]# |
```

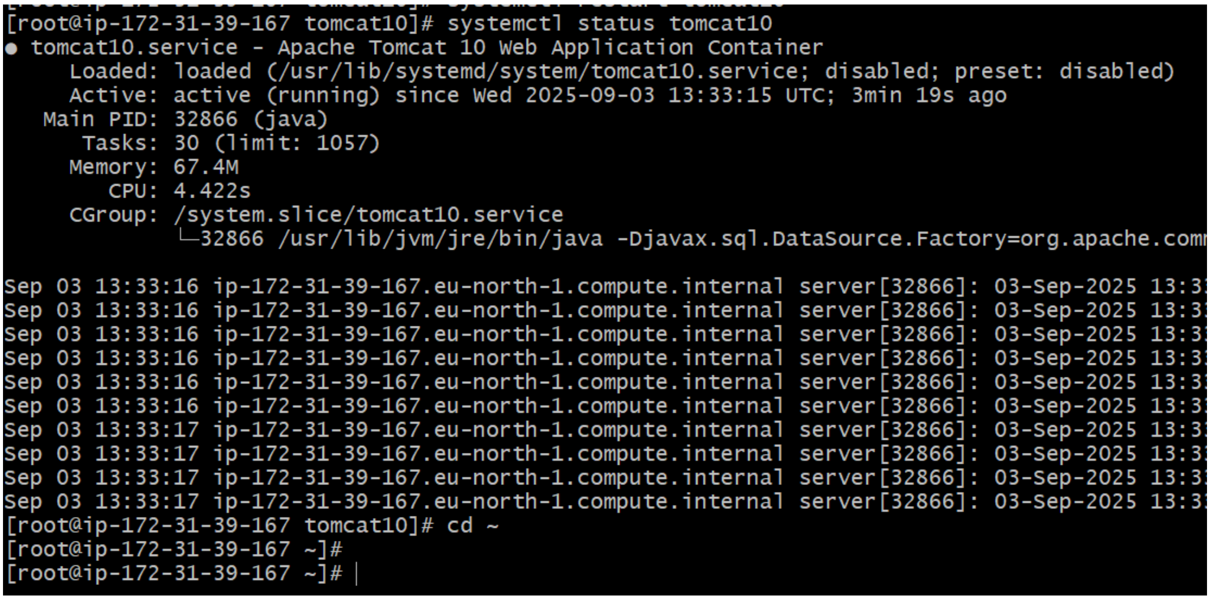
- Check the status using : netstat -na | grep 8080
- Ps -ef | grep 8080 - you see the port 8080 is listening.
- Default log files for tomcat is catlina.out you can see logs.
- Go to web page and enter : “ip “ :8080 – you can see tomcat apache is running on 8080 port

```
[root@ip-172-31-39-167 ~]# find / -name server.xml
/etc/tomcat10/server.xml
[root@ip-172-31-39-167 ~]# cd /etc/tomcat10/
[root@ip-172-31-39-167 tomcat10]# vi server.xml
```

- Fin dthe server.xml file
- Vi server .xml file

```
        Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
    -->
    <Connector port="8082" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443"
               maxParameterCount="1000"
               />
    <!-- A "Connector" using the shared thread pool-->
    <!--
    <Connector executor="tomcatThreadPool"
               port="8082" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443"
               maxParameterCount="1000"
               />
    -->
    <!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443 with HTTP/2
         This connector uses the NIO implementation. The default
         SSLImplementation will depend on the presence of the APR/native
```

- Change the port to 8082 and save it
- Restart the service and check status



- Got ot web page and enter : “ip address”:8082
- You can see the tomcat apache is running on port8082.

6. DEPLOY SAMPLE APPS ON WEB APPS

- Goto directory /opt/apache_tomat9/webapps
- Download the sample.war that is sample app file
- sample.war file in the directory .
- restart the service : systemctl restart tomcat 9
- Then go to web page an enter : ipadress:80/sample.
- Then you can see the sample app hasbeen deployed.

7. CREATE TOMCAT.SERVICE FILE

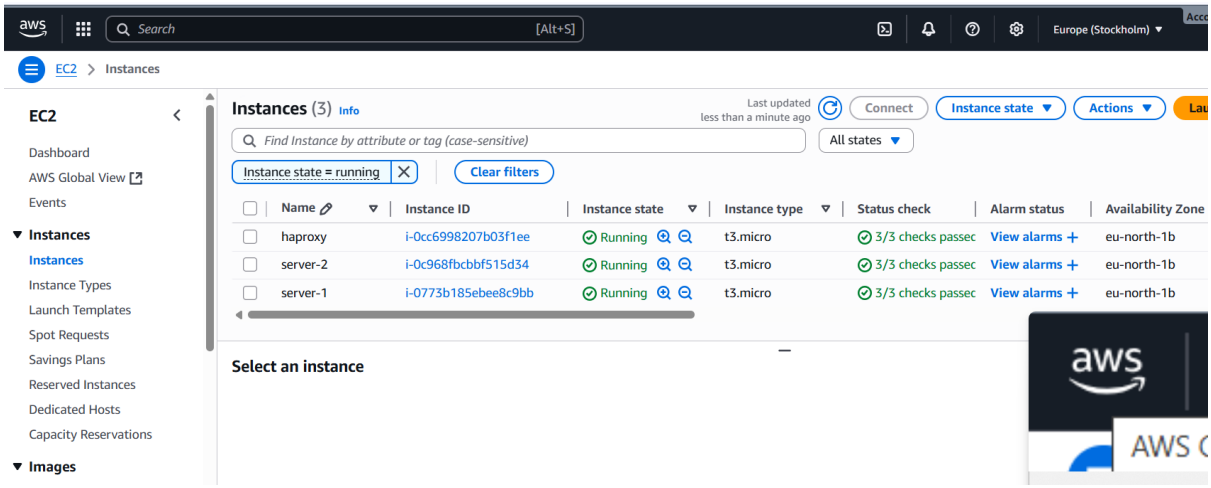
```
[root@ip-172-31-35-46 ~]# cd /etc/systemd/
[root@ip-172-31-35-46 systemd]# ls
coredump.conf  logind.conf    oomd.conf      sleep.conf   system.conf.d   user.conf
homed.conf     network        pstore.conf    system       timesyncd.conf
journald.conf  networkd.conf  resolved.conf  system.conf  user
[root@ip-172-31-35-46 systemd]# cd system
[root@ip-172-31-35-46 system]# ls
basic.target.wants                    nfs-idmapd.service.requires
chronyd.service.wants                 nfs-mountd.service.requires
cloud-init.target.wants               nfs-server.service.requires
ctrl-alt-del.target                   remote-fs.target.wants
dbus-org.freedesktop.home1.service    rpc-gssd.service.requires
dbus-org.freedesktop.network1.service rpc-statd-notify.service.requires
dbus-org.freedesktop.resolve1.service rpc-statd.service.requires
dbus.service                          sockets.target.wants
getty.target.wants                    sysinit.target.wants
multi-user.target.wants               sysstat.service.wants
network-online.target.wants           systemd-homed.service.wants
nfs-blkmap.service.requires           timers.target.wants
[root@ip-172-31-35-46 system]# vi tomcat.service
[root@ip-172-31-35-46 system]#
```

- Go to the default location of file that is system
- Add tomcat.service file : vi tomcat.service
  Add configuration :

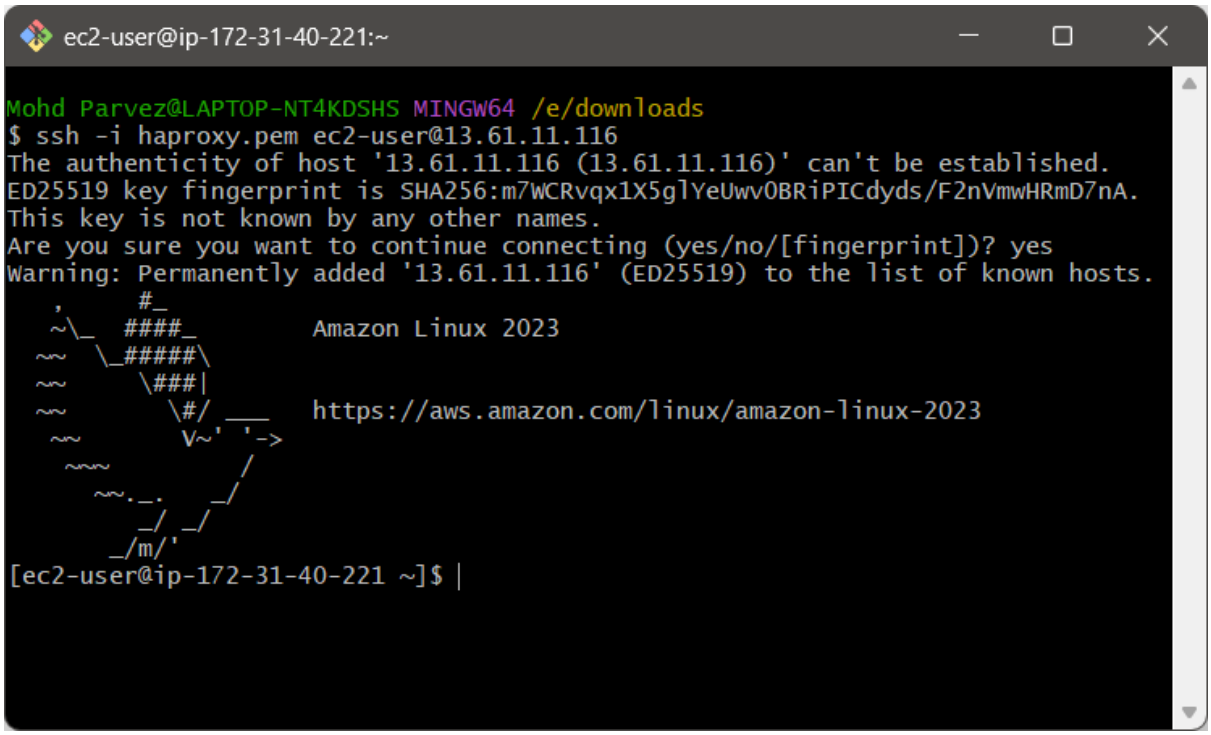
```
root@ip-172-31-35-46:/etc/systemd/system
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target

[Service]
Type=forking
Environment=JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat
ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh
User=tomcat
Group=tomcat
RestartSec=10
Restart=always

[Install]
WantedBy=multi-user.target
~
~
~
~
~
~
~
~
~
"tomcat.service" 19L, 456B                                    19,26         All
```

- save and exit
- restart the service
- make sure to create user with name tomcat an dgive access
- reload the daemon

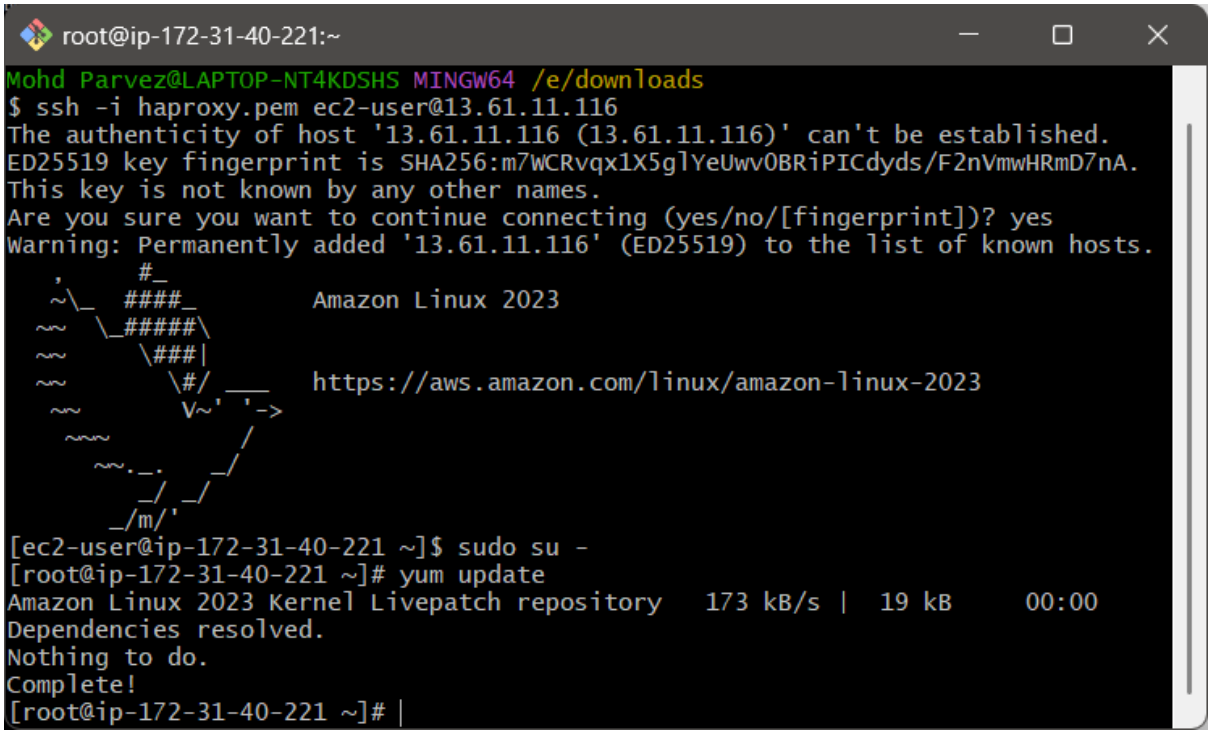
8. CONFIGURE HAPROXY SERVER.



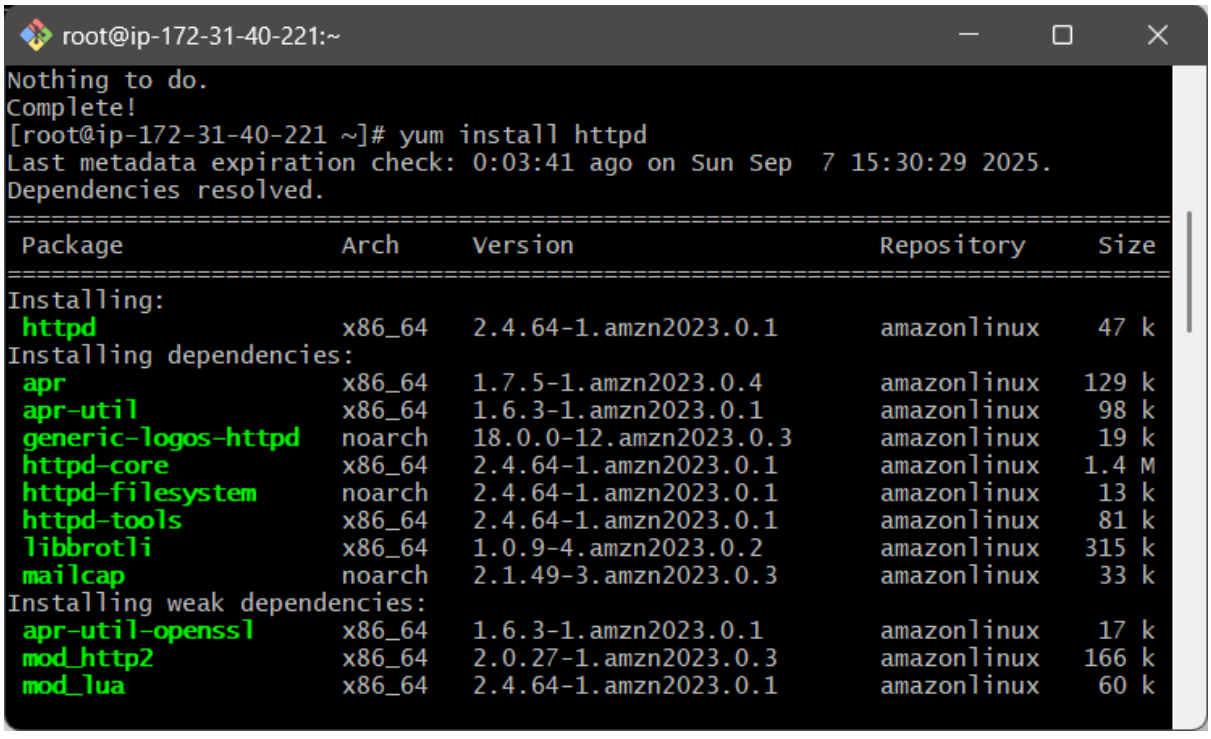
- Lauch three ec2 instance



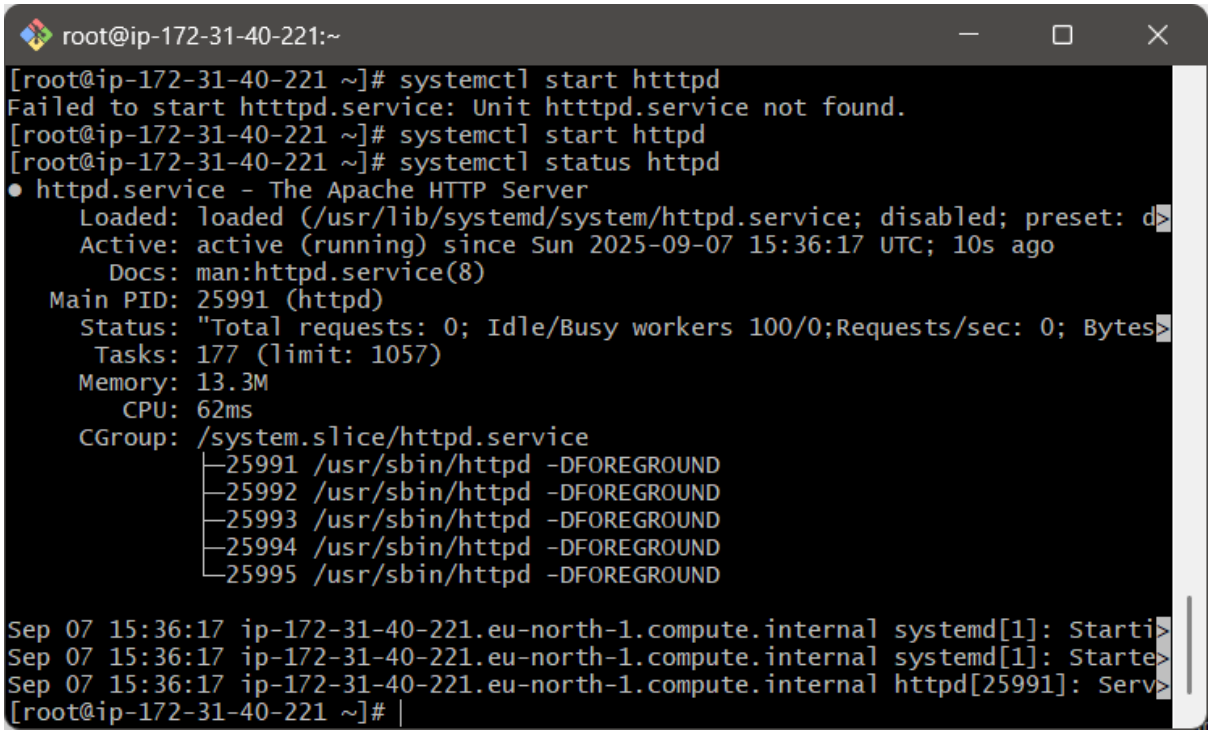
- Go to the location where keypar.pem is downloaded and right click and select open git bash here and connect to the server-1 instance.



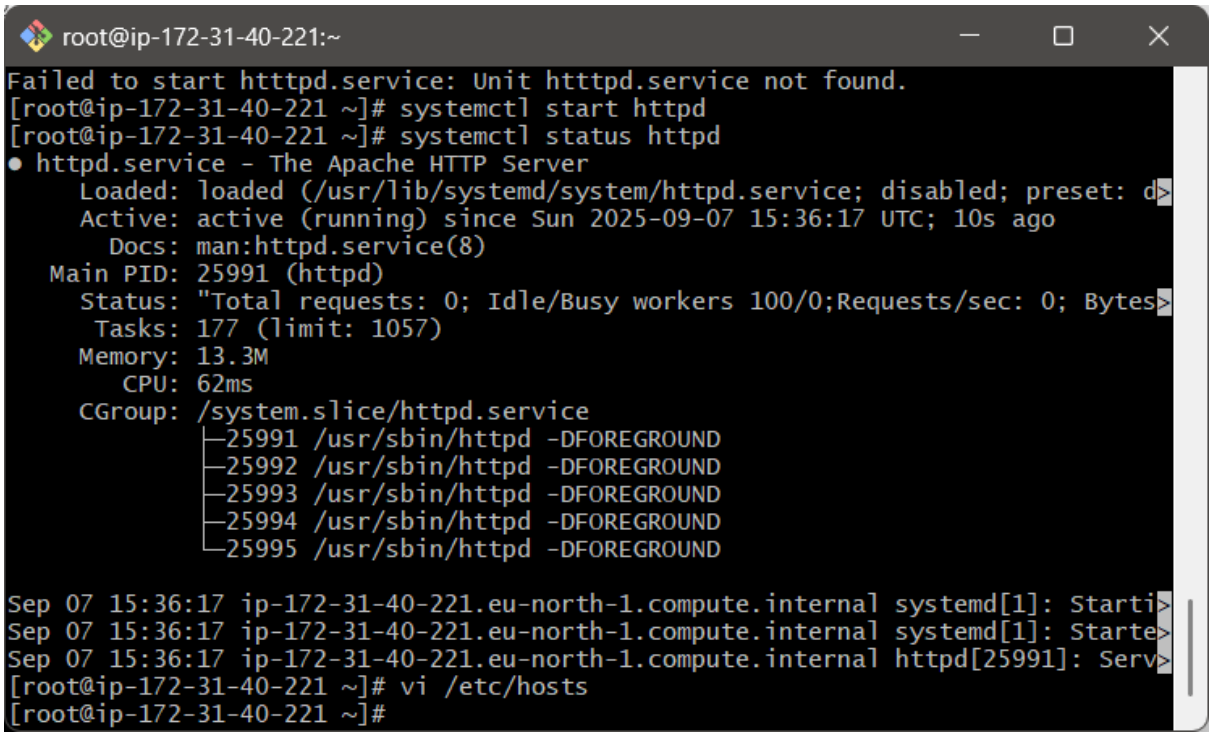
- Switch to root user : sudo su –
- Update the server : yum update



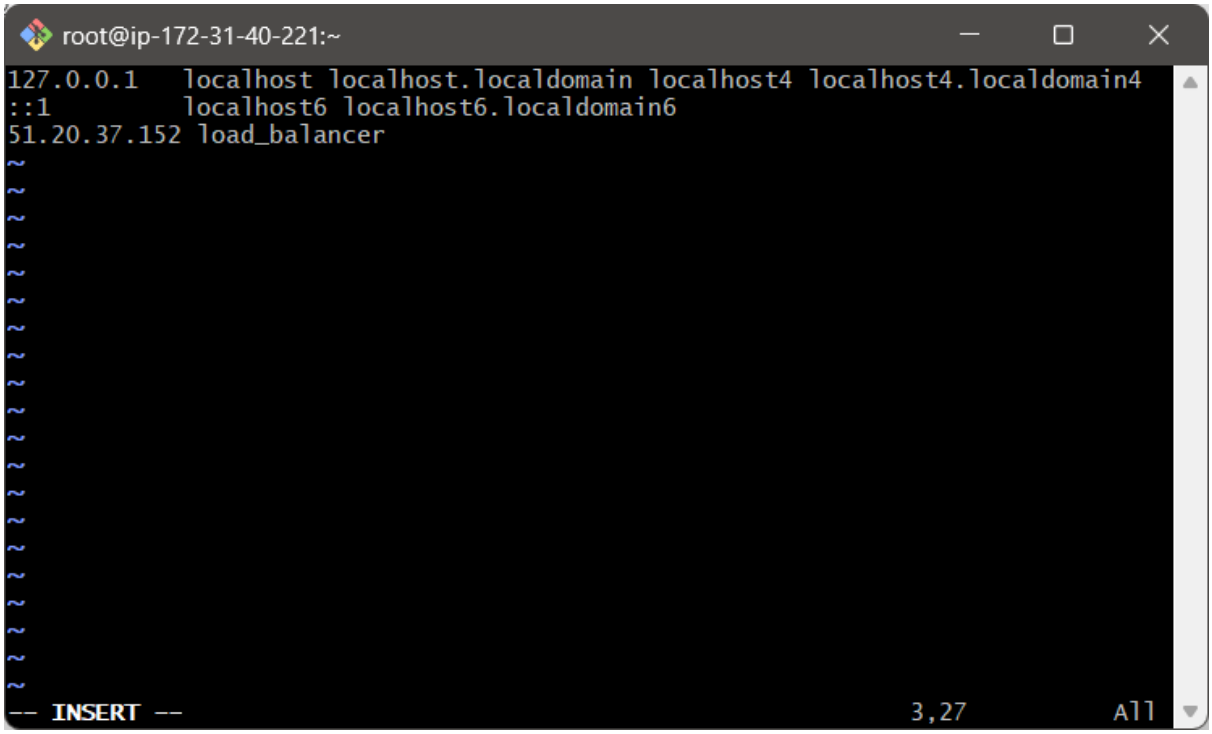
- Install httpd : yum install httpd
- Start the httpd service : systemctl start httpd



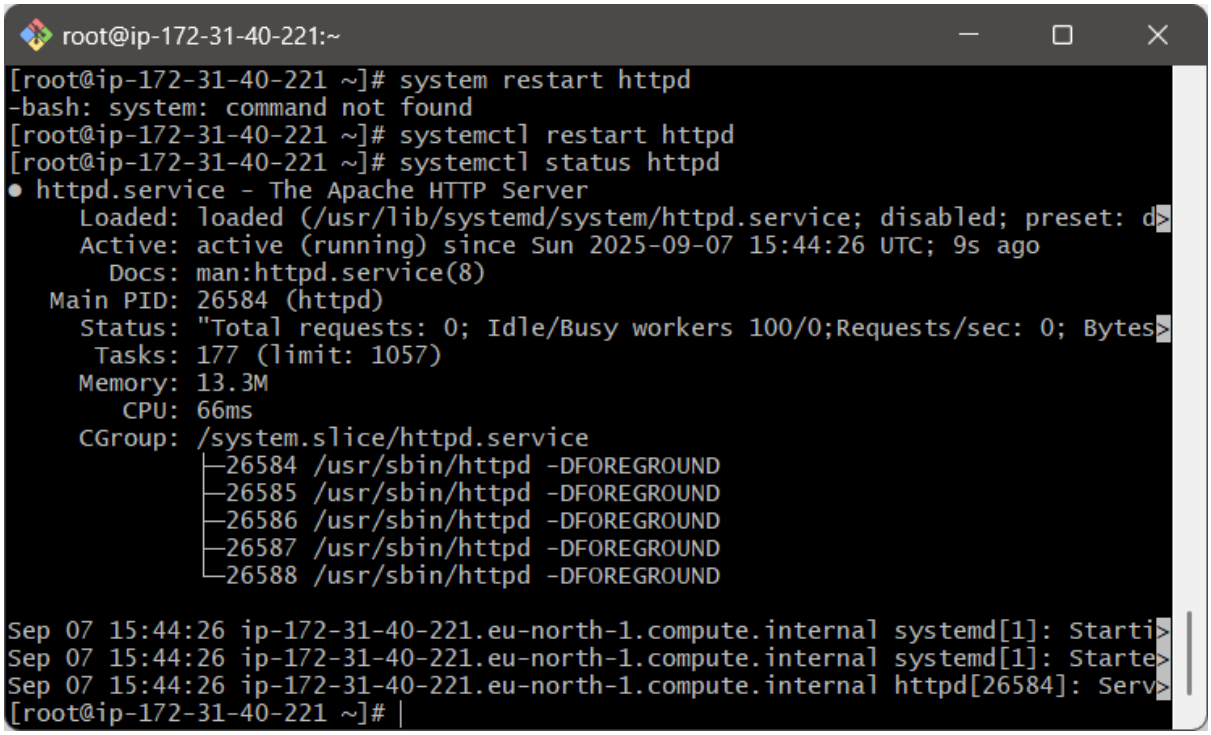
- Check the status :systemctl status httpd



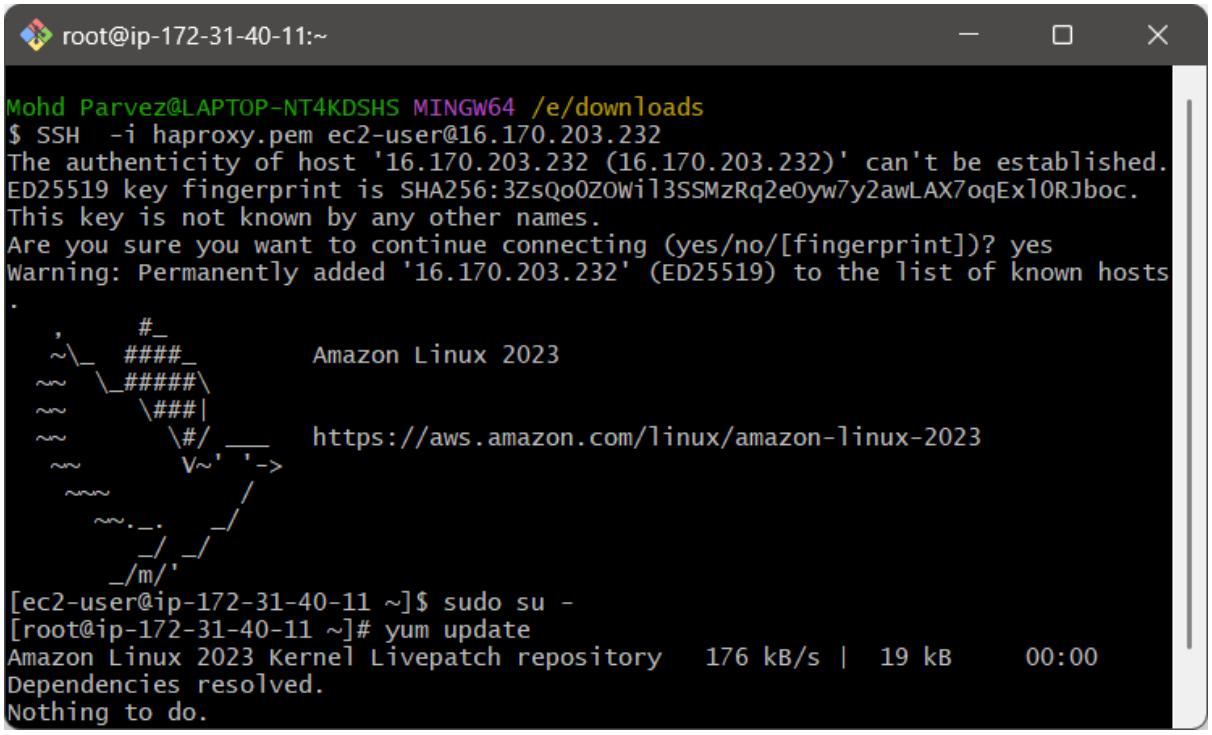
- Go to : Vi /etc/hosts



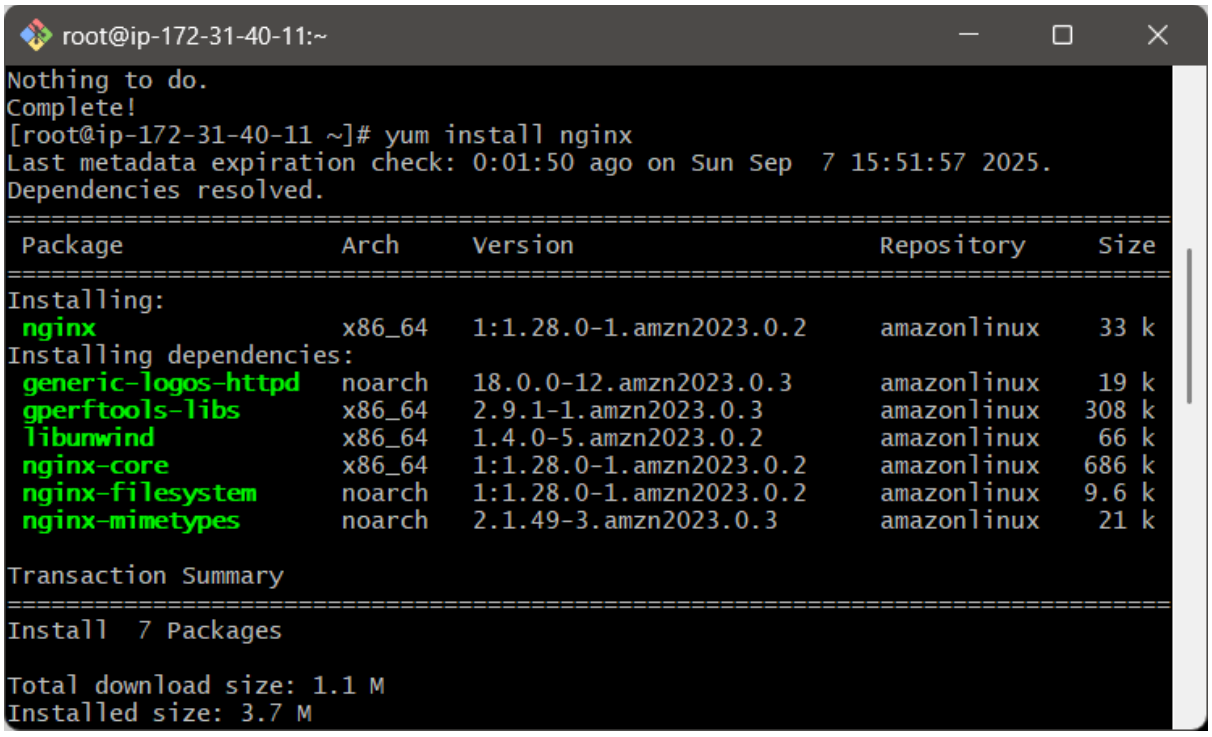
- Add HA-Proxy-Server : Pubic IP Address
- Save and exit the file : “esc”(button) :wq!



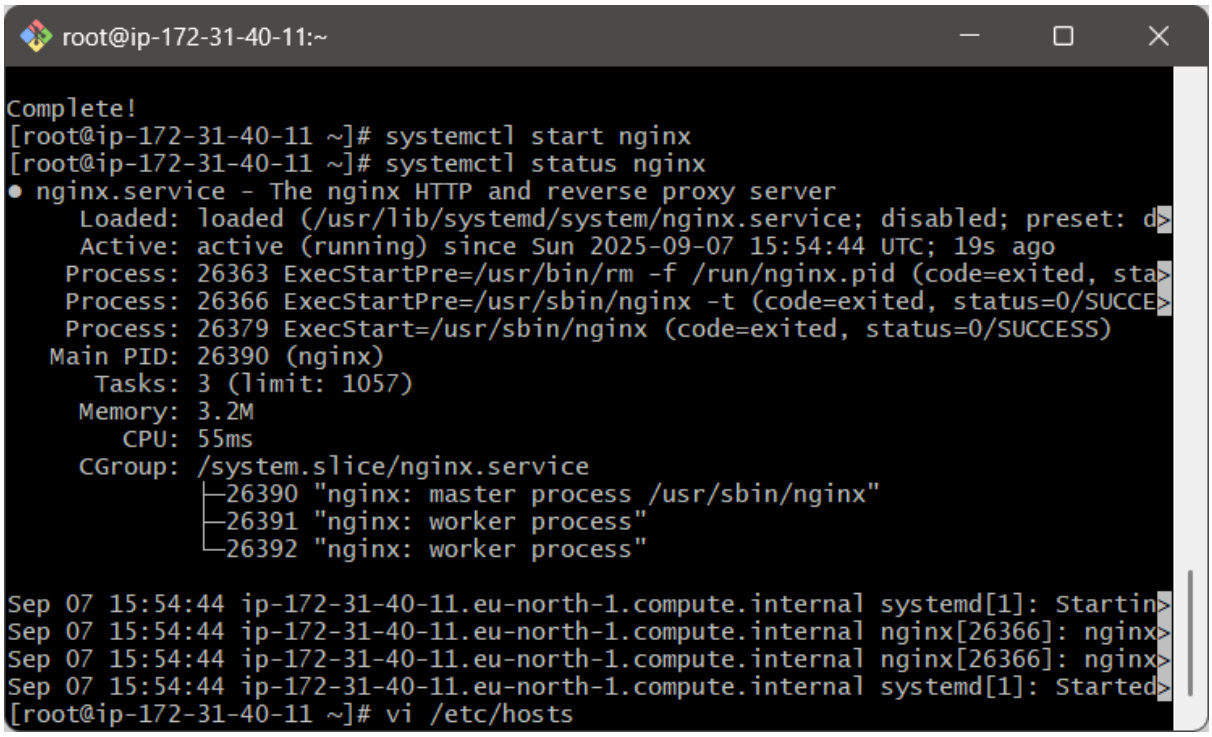
- Restart and check the status of httpd



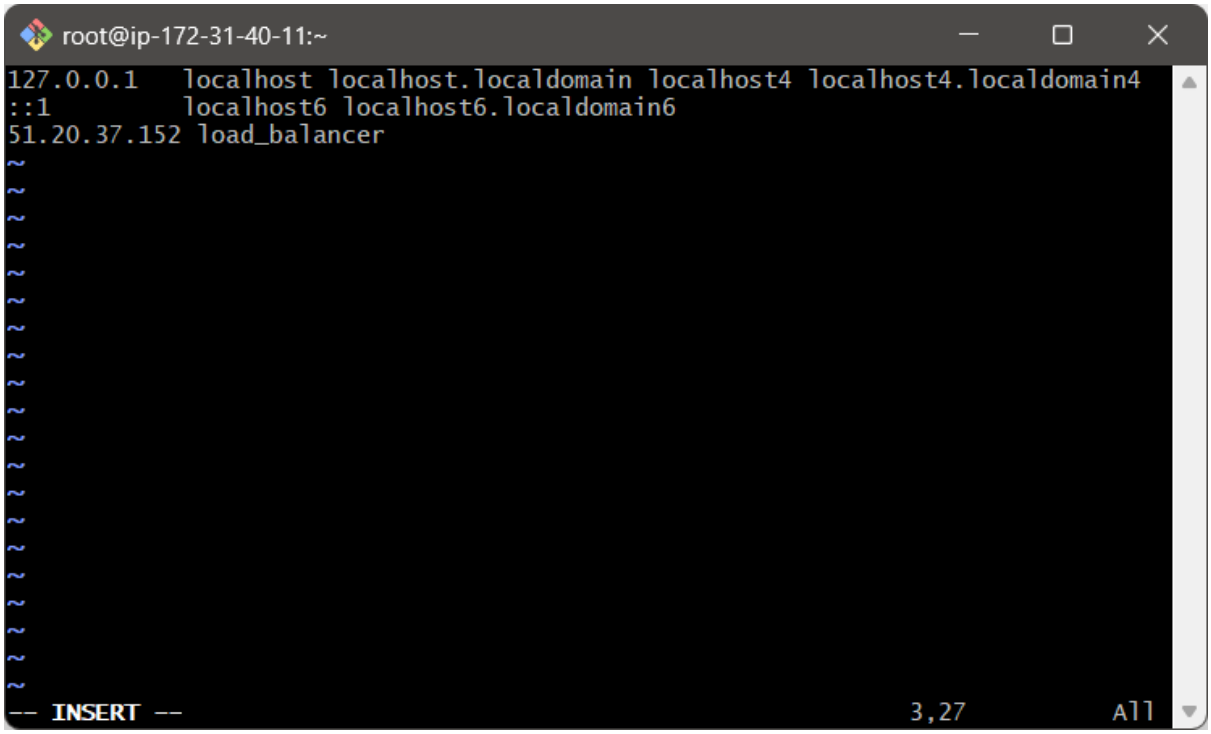
- Connect to the server-2
- Switch to root user



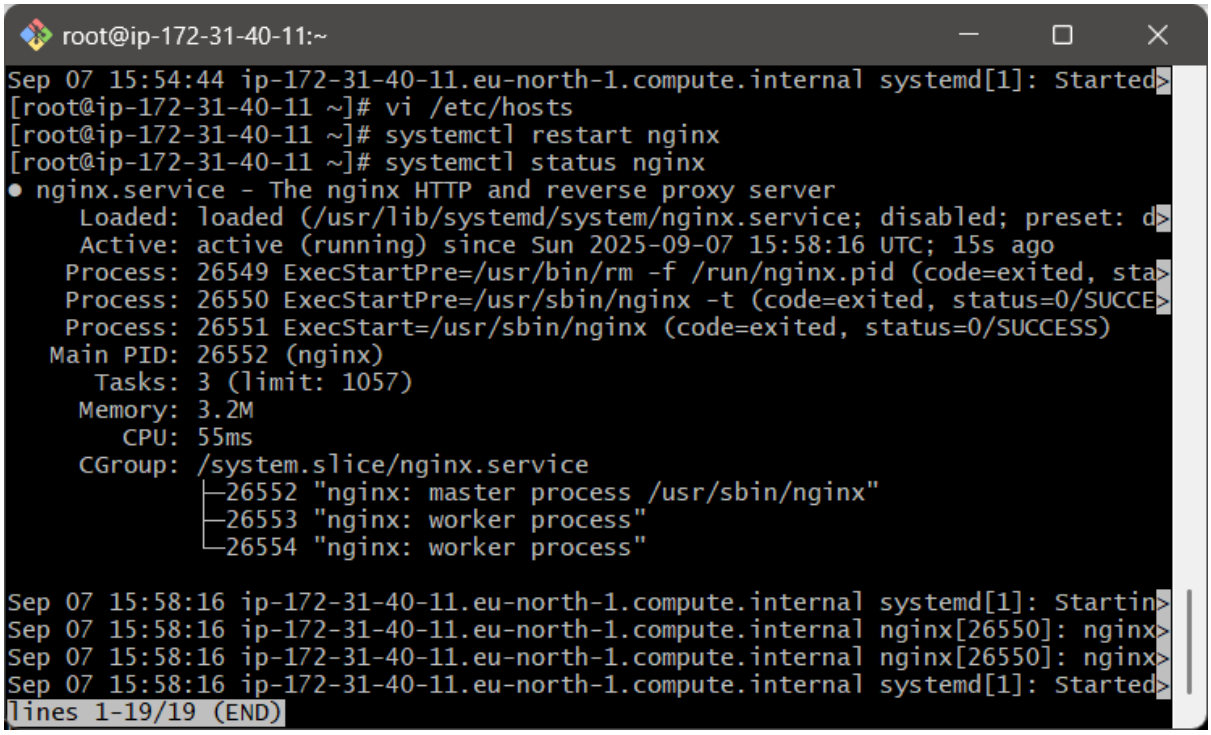
- Install nginx on server-2



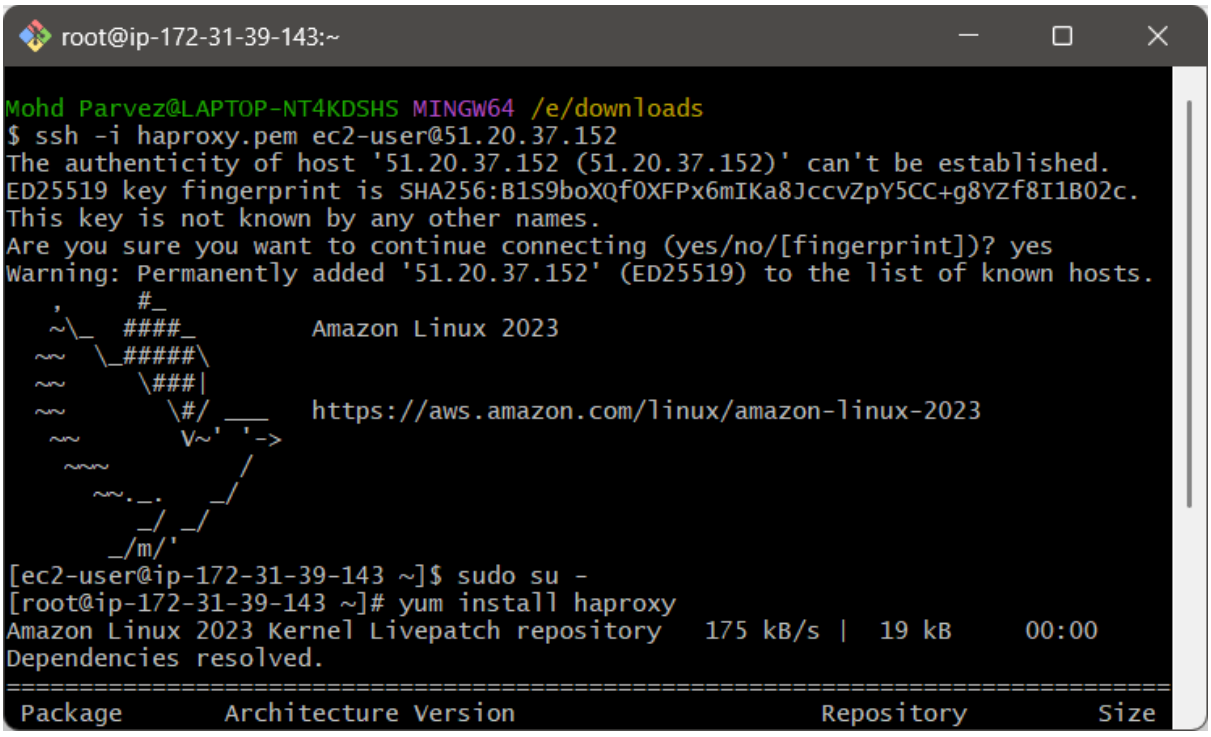
- start the nginx service
- Check the status
- Go the hosts file



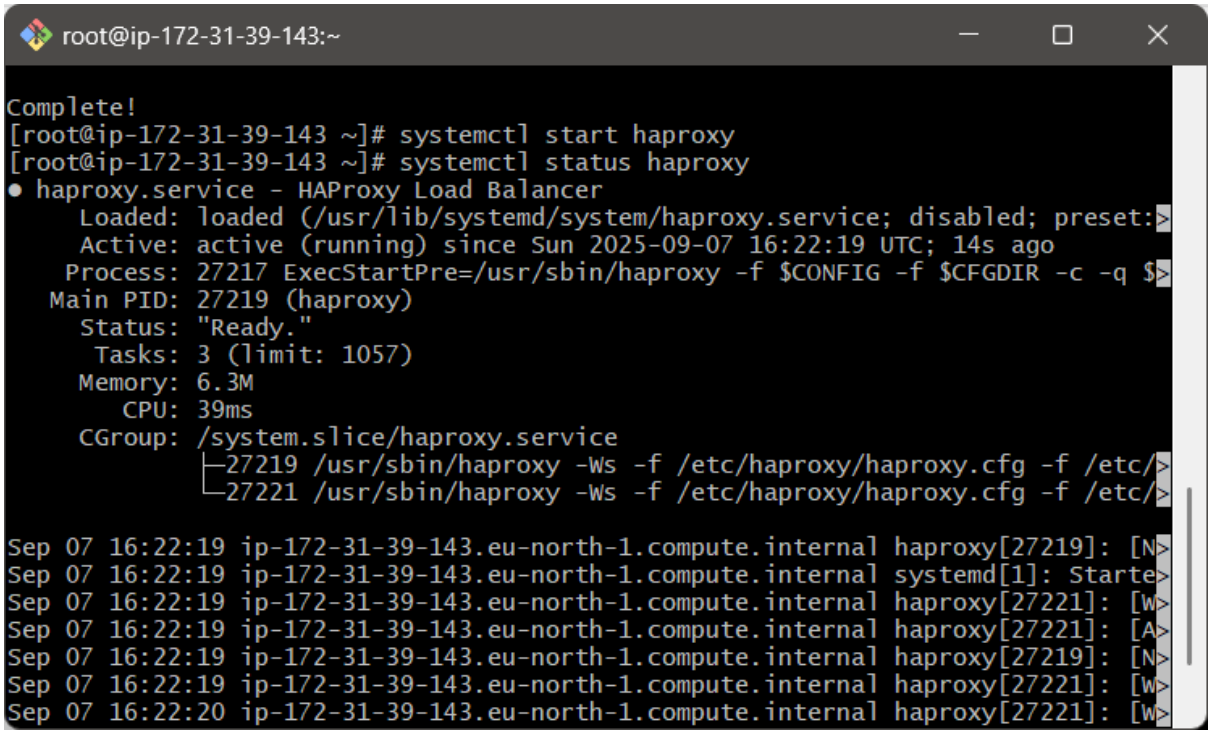
- Add ha proxy i[address



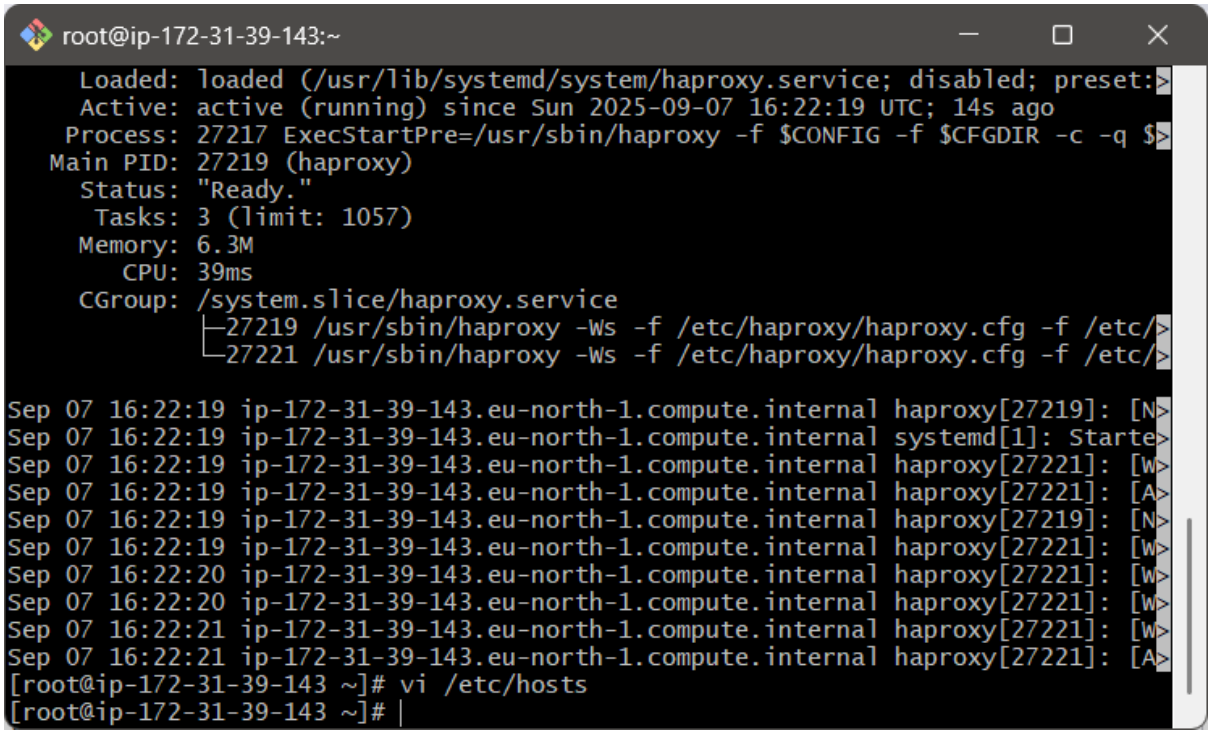
- Restart and check the status



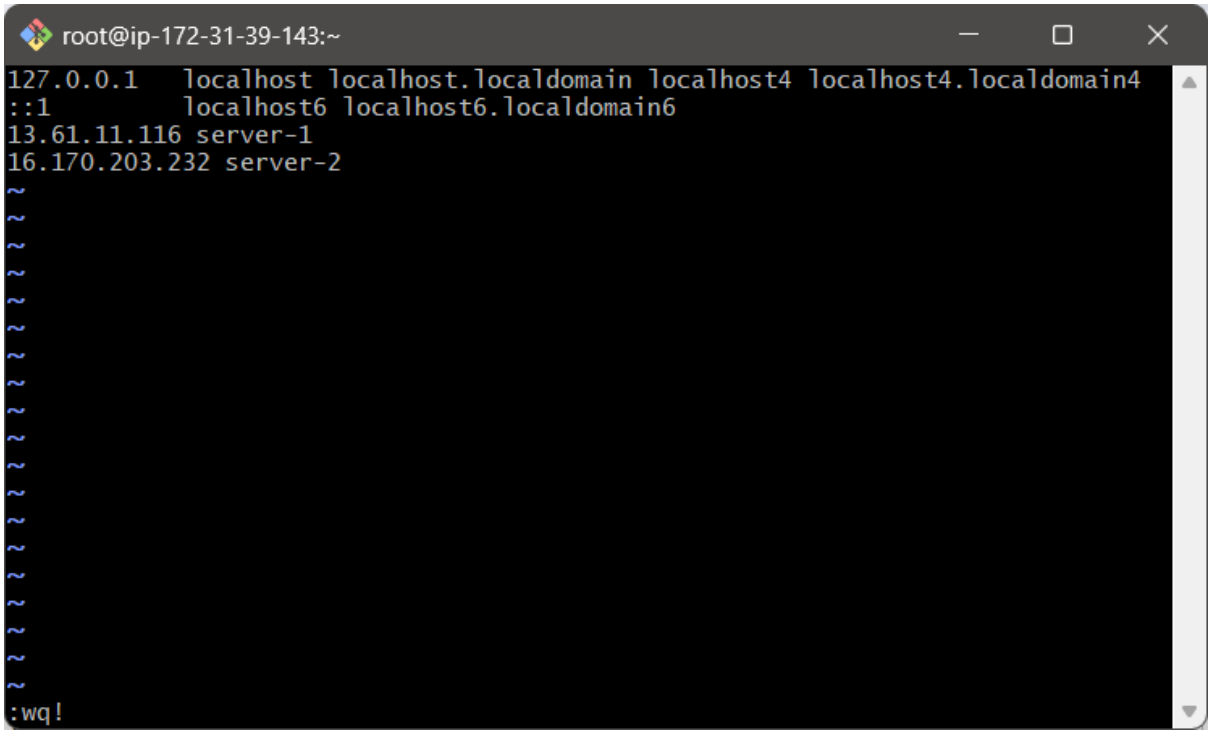
- Connect to haproxy instance
- Switch to root user
- install haproxy service



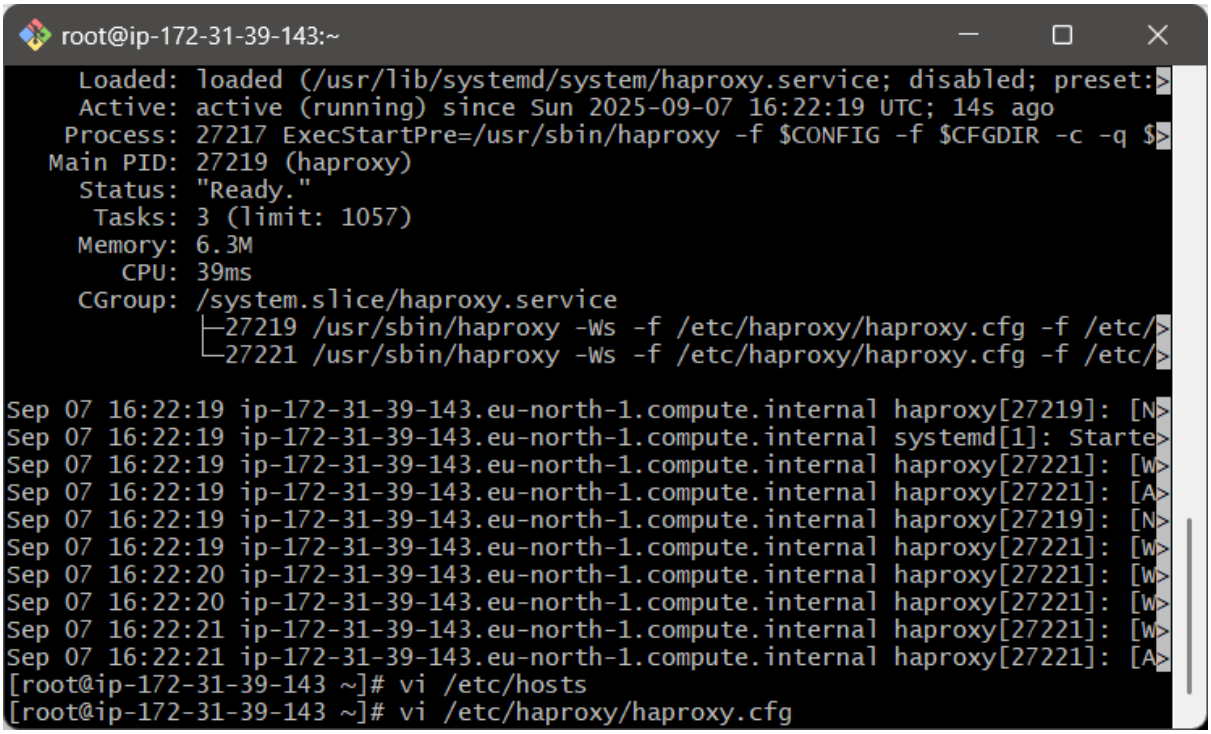
- start the haproxy service and check the status.



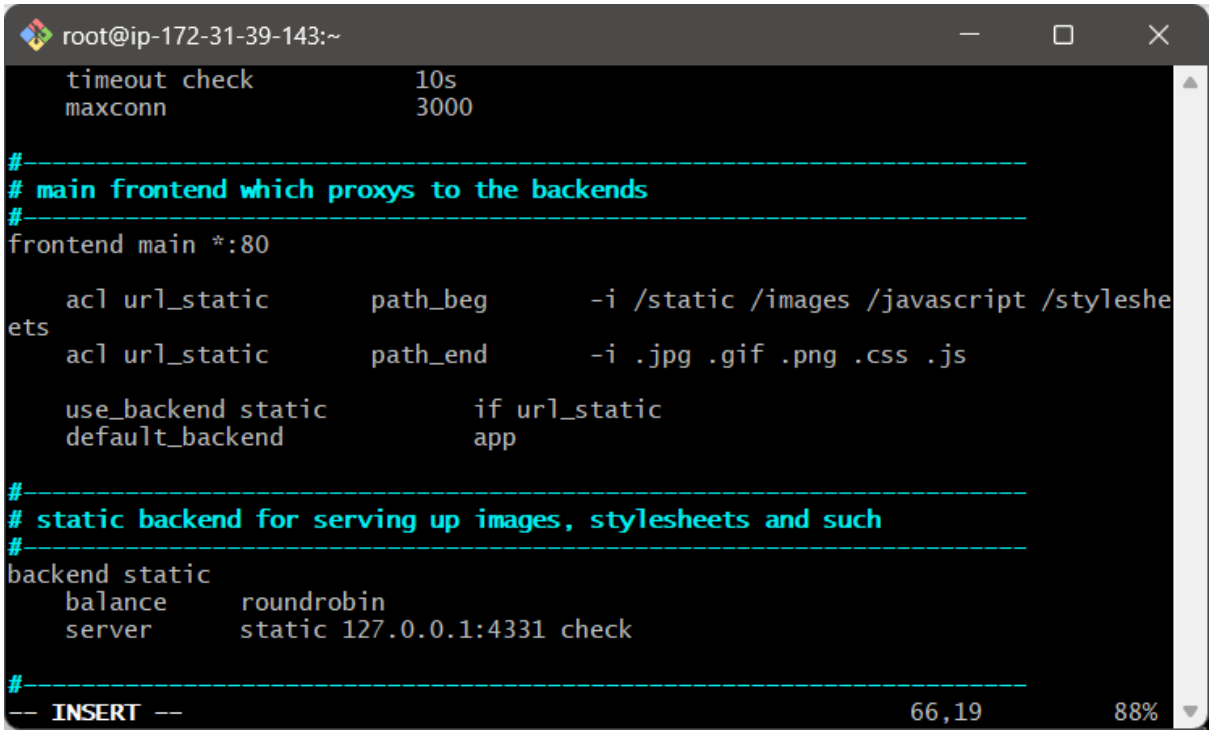
- go to the hosts file and add ip address of server-1 and server-2

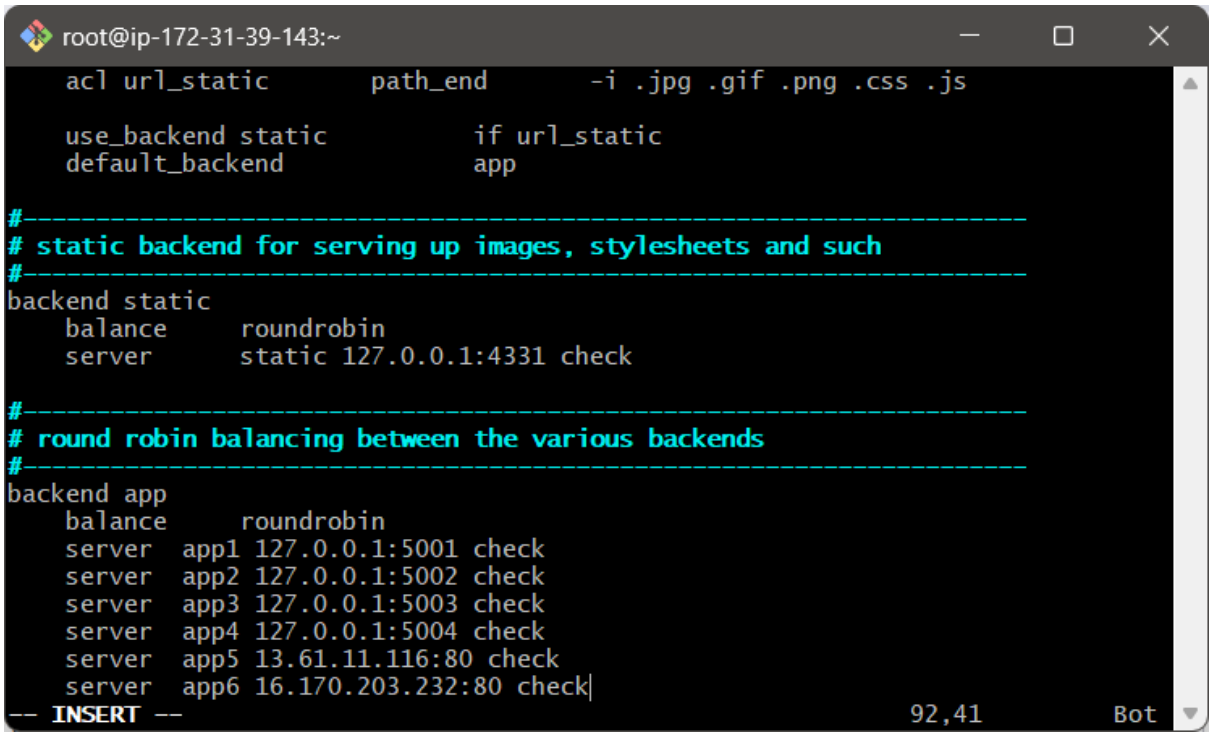


- add ip address of server-1 and server-2 and save it

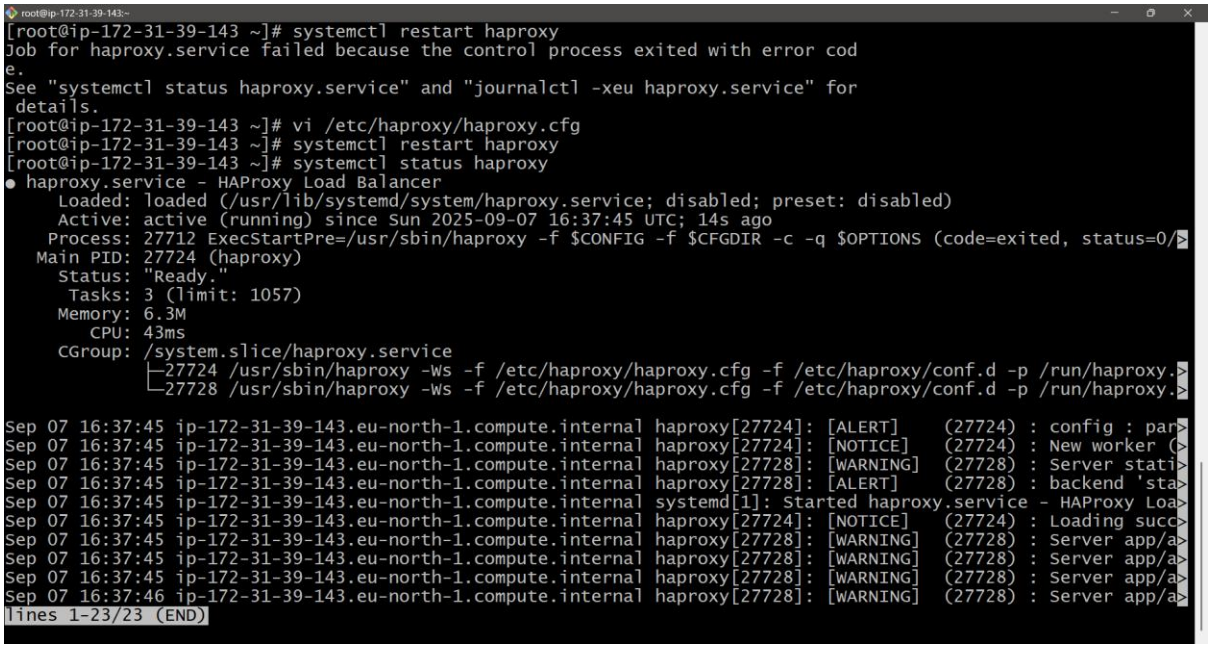


- Go to the haproxy.cfg file





- Add server-1 and server-2 ip adrees and port to the file
- Save and exit



- Restart and check the status of the service

- Go to browser and copy the haproxy server ip and enter : ip adress port number
- You can see first httpd service and refresh you see the nginx is running on the port 80