Task – 2 – webservers

1) Install nginx and run nginx on port number 81.

2) Deploy a sample index.html file on nginx.

3) Install Apache and run Apache on port number 82

4) Deploy a sample index.html file on Apache.

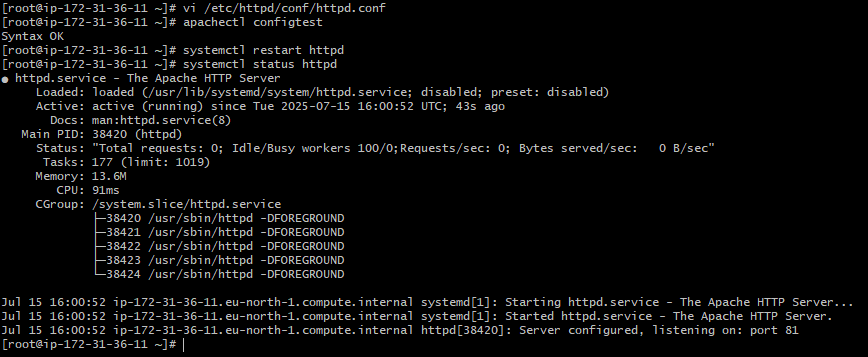
5) Install Apache tomcat on port number 8082

6) Deploy a sample app on webapps

7) Create a tomcat.service file for tomcat.

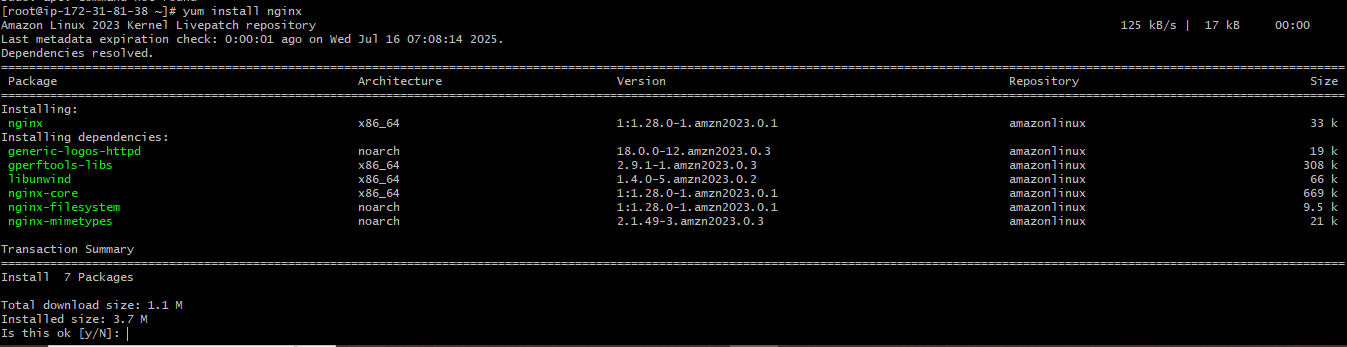
8) Configure HA Proxy server

1) Install nginx and run nginx on port number 81.

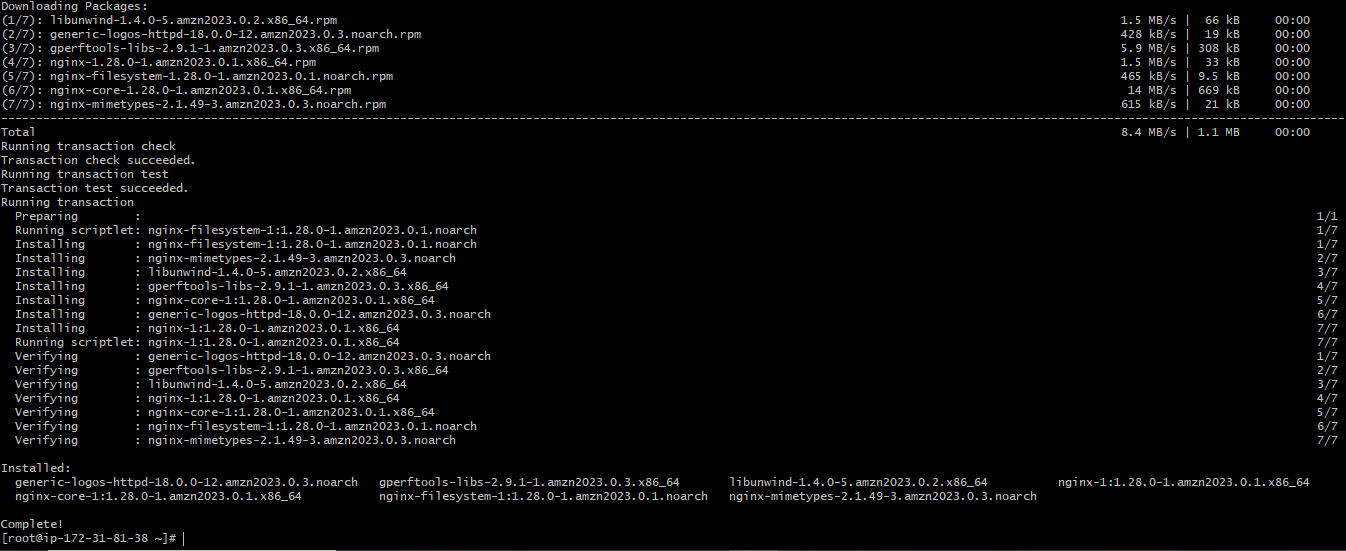


First install nginx command is

Switching to root bcz don’t want un necessary issues.



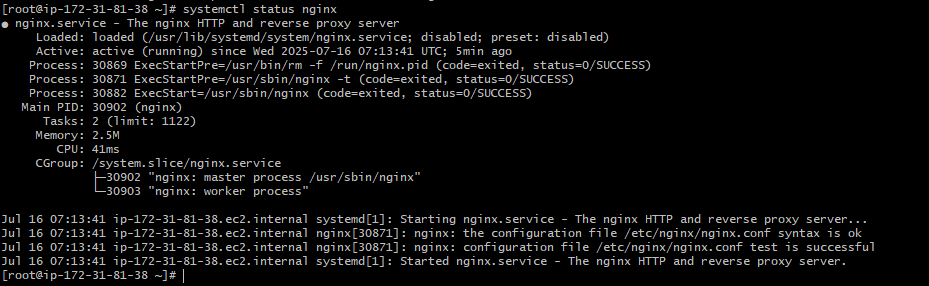
Then press yes



Now start the nginx



Now check the status of nginx



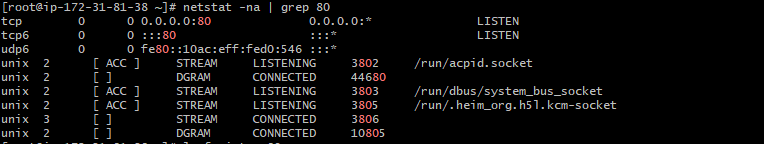
If it is showing …… as log ile then use command

Systemctl ststus nginx –l

There is another way to check the status command is

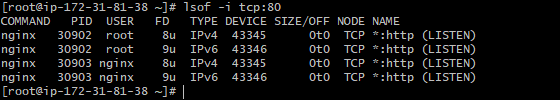
Netstart –na |grep 80

It shows listen port 80 only



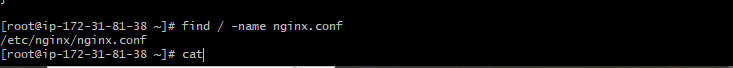
But to check what is running now then

Lsof –i tcp:80



Now to find the nginx file in which path it is we use below command because we can read the file and we can ind the port.

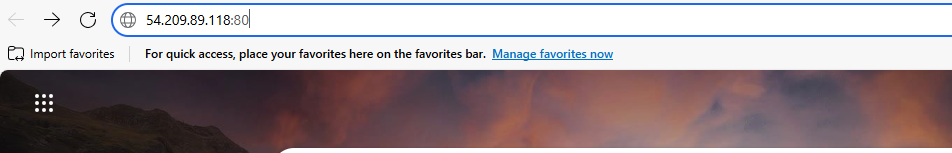
Find / -name nginx.conf

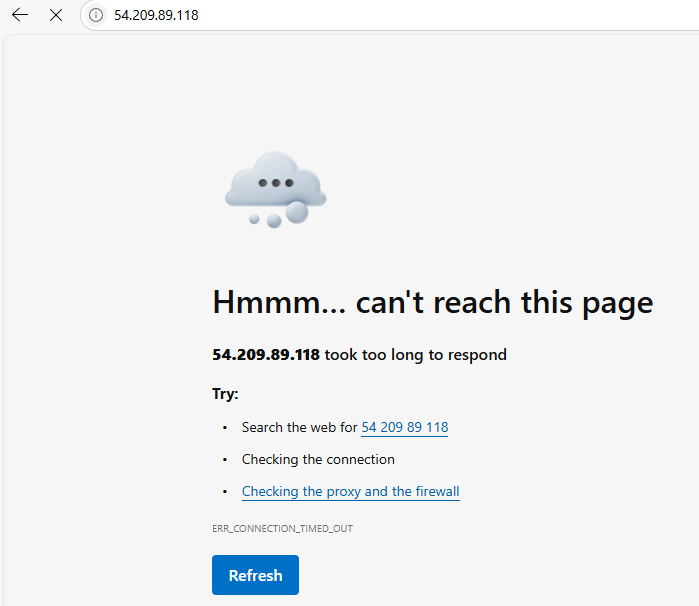


Here path is /etc/nginx/nginx.conf

Now read the file

Cat /etc/nginx/nginx.conf

It will open the complete just read and ind on which port nginx is running if it’s showing 80 then I you copy the public IP Id from EC2 along with keep :80 in the browser then it will show the page is not opening lie below.



To run the page just change the port rom 80 to 81 bcz its already running on port 80

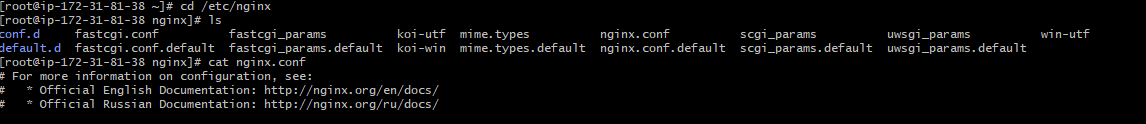
Or just try in 81 (why we need to change port researh )

Now to add/edit the 81 in the ile nginx.conf you use command

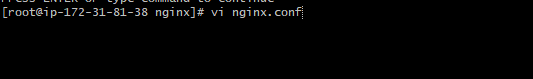
Nginx is in nginx.conf file. So first come into the directory nginx



Then see here files ls



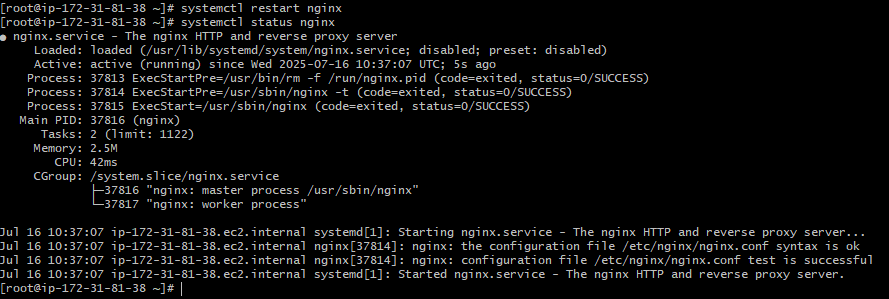
Vi nginx.conf



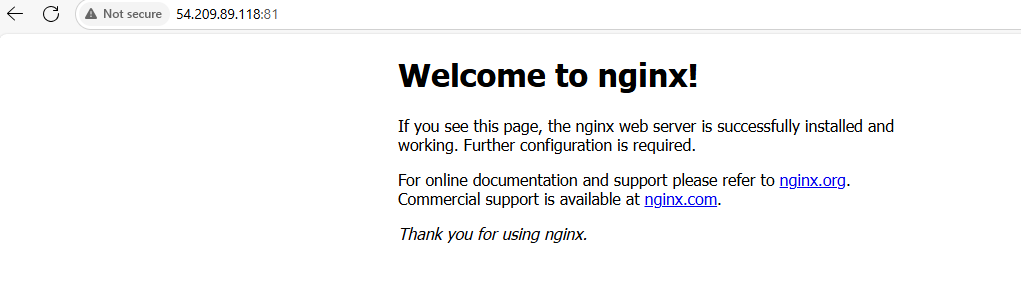
Here just change the port beside listen 80 to 81

Then restart the nginx that is

Systemctl restart nginx



Now go to browser an check whether its working on port 81 or not



Yes done

2) Deploy a sample index.html file on nginx.

To deploy index.html nginx file first

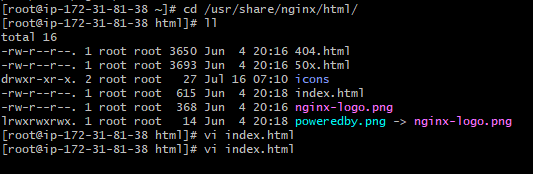
Find the path of html file command is

Find / -name html

(this index html already created in the nginx. So just edit that and change the text then see is it appearing on web browser or not. We can careate our own sample index.html file but again we need edit the nginx software with the same details. Do research on that)

Once you find the path in the find just change the directory to html because their only we can see the ll ling list of the file and it is showing index.html file or not.

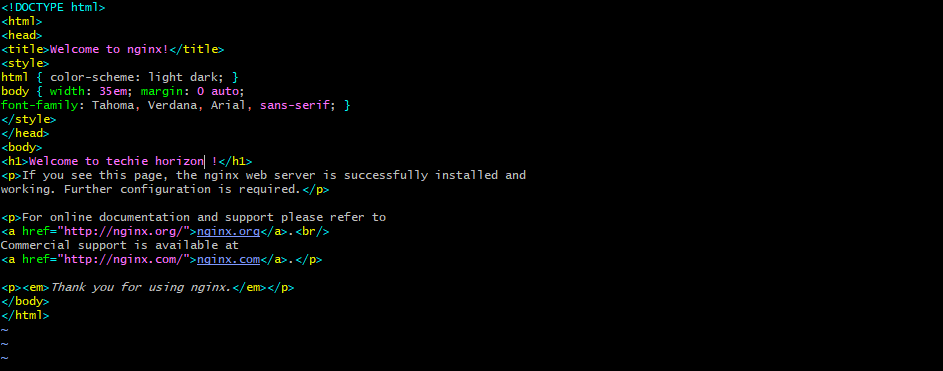
cd /usr/share/nginx/html/



Then read by using cat /usr/share/nginx/html/

And just see the content and just edit as per need in this index.html file

vi index.html



Here I just changed <h1>Welcome to nginx !</h1>

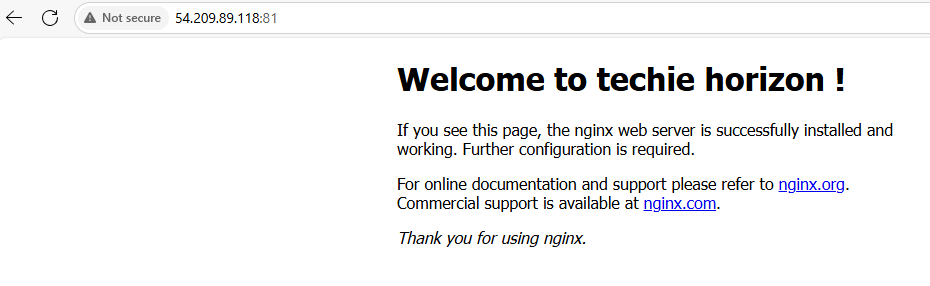
To

<h1>Welcome to techie horizon !</h1>

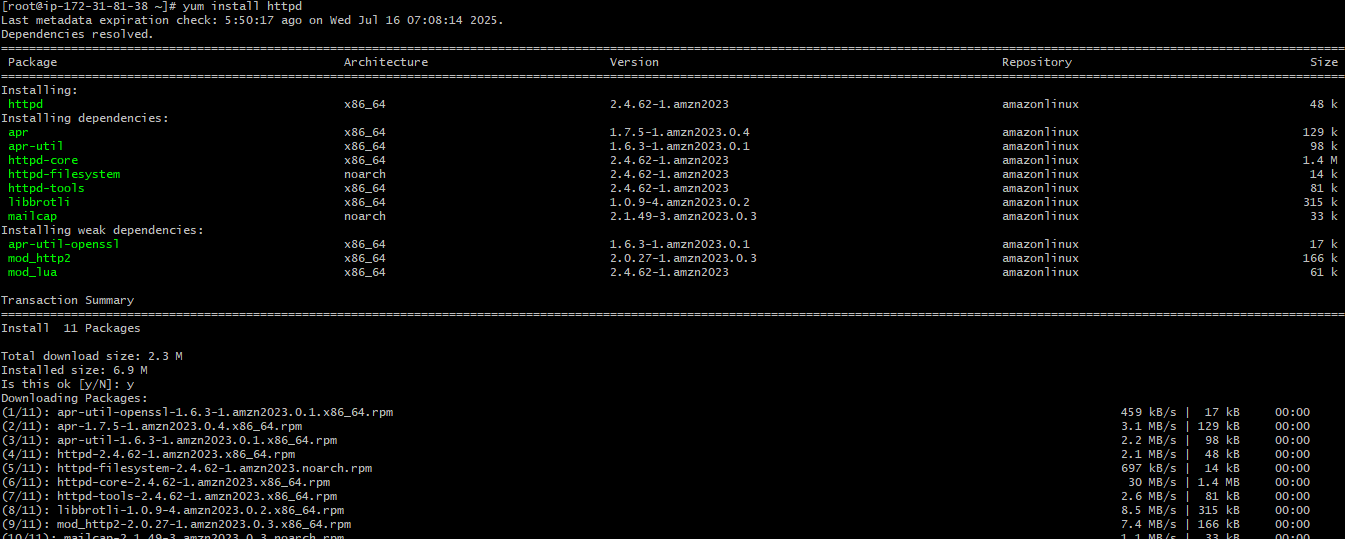
Then save it

Then go to browser the changed html file is showing as per change or not

Here it is showing



3) Install Apache and run Apache on port number 82

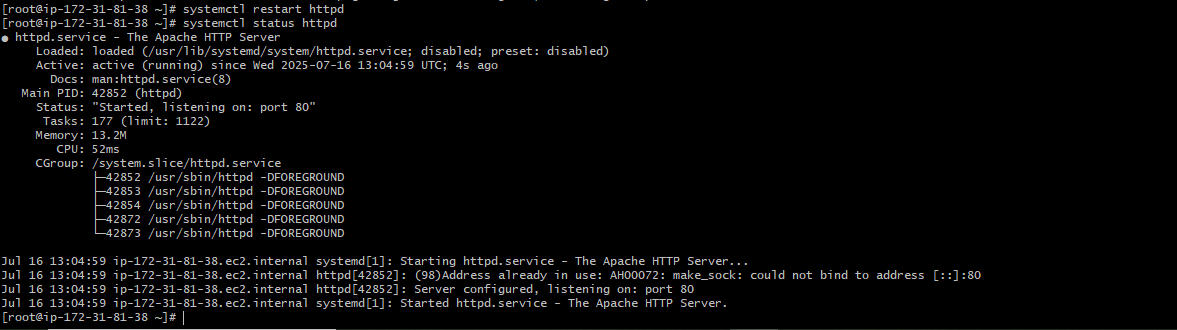


Systemctl status httpd

Or

Systemctl status httpd –l (if shows logs .....then use this command)

Othervise no need

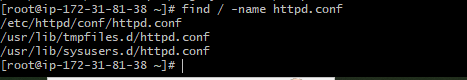


As it showing listening on port 80

No change this to port 82 as per the task

fInd the path

find / -name httpd.conf



Now read to check the port

Cat /etc/httpd/conf/httpd.conf

Yes it is on port 80

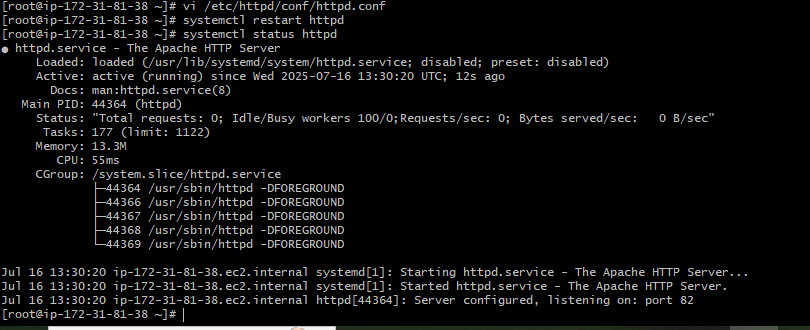
Now edit/write httpd.conf file to change the port id to 82

vi /etc/httpd/conf/httpd.conf

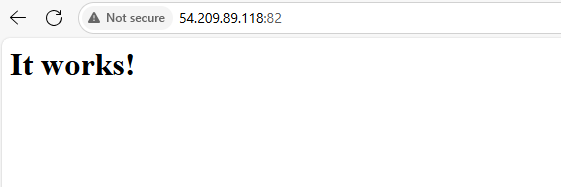
save it after keeping 82

then restart the httpd

check staus



Now check browser



4) Deploy a sample index.html file on Apache.

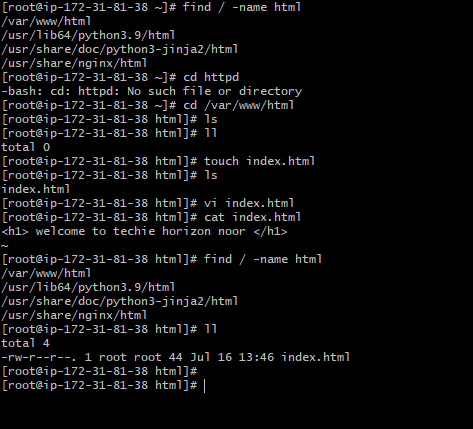
First we need to find the path of html

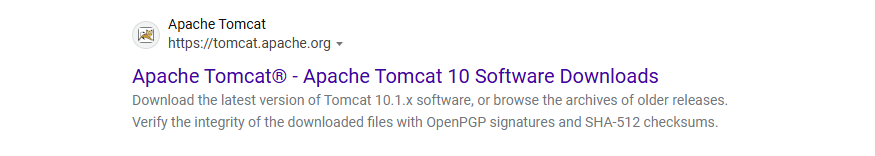
Find / -name html

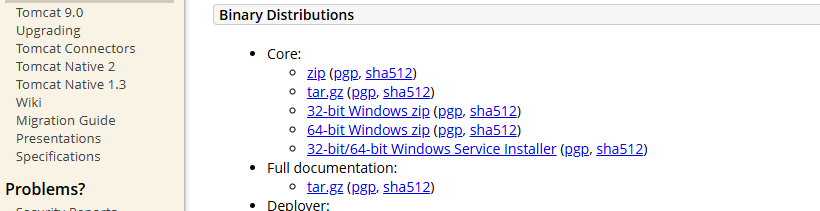


Then come to directory o html by entering

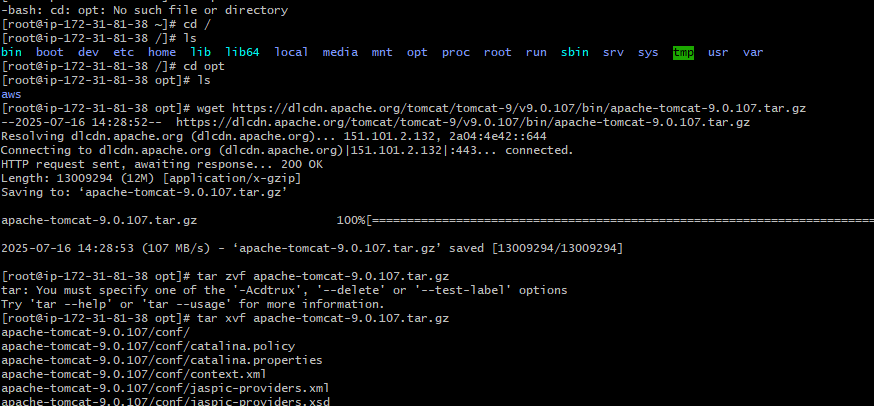
Here in the above we can create html file (index.html) as it was not created in the hhtpd earlier.

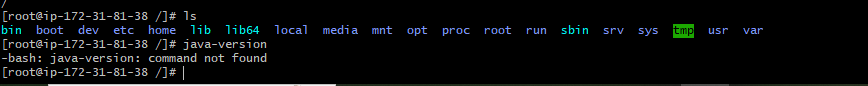






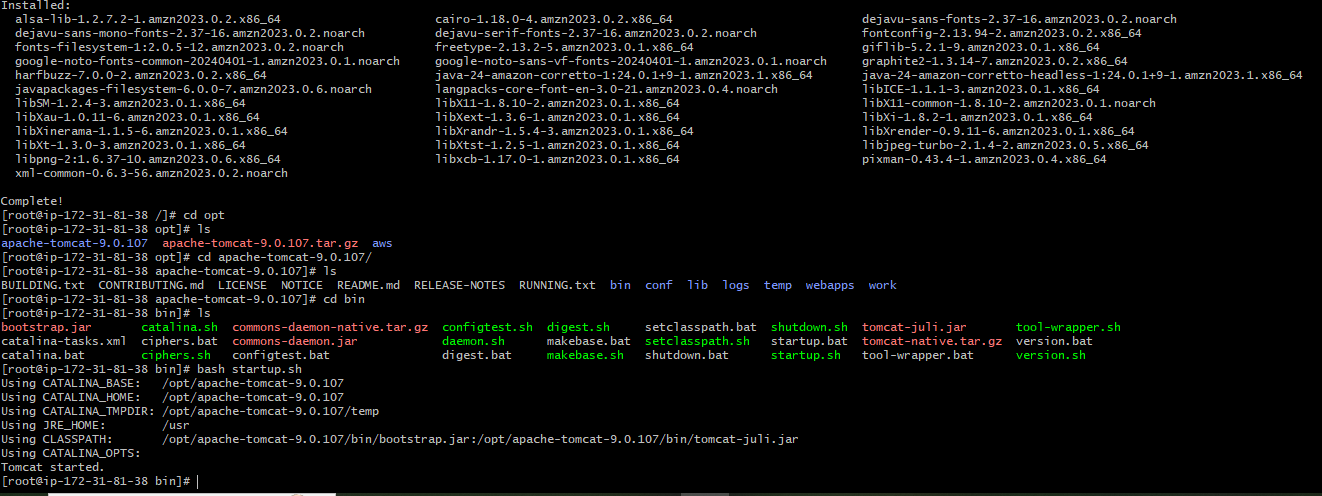
Left click on tar and copy link

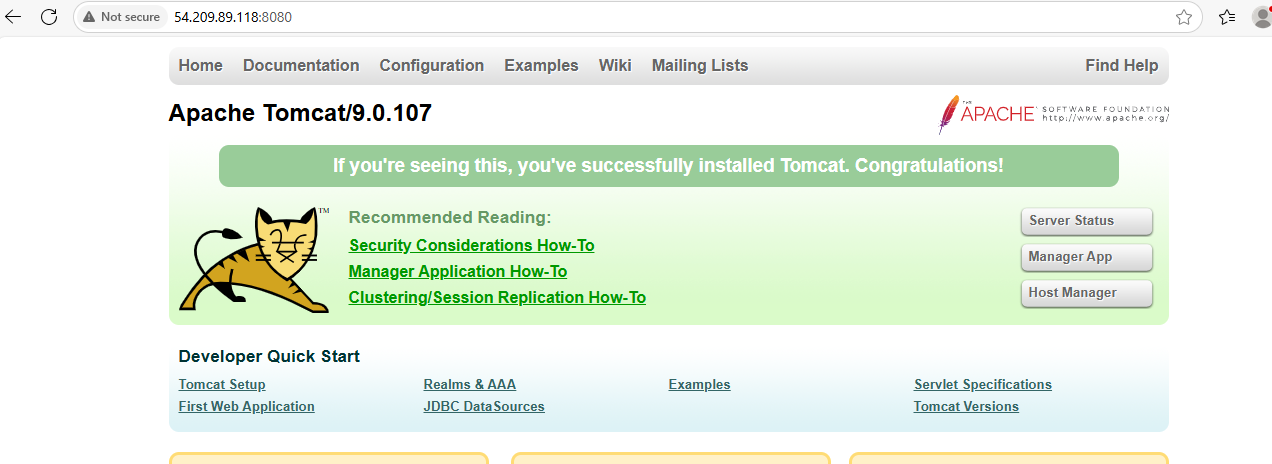




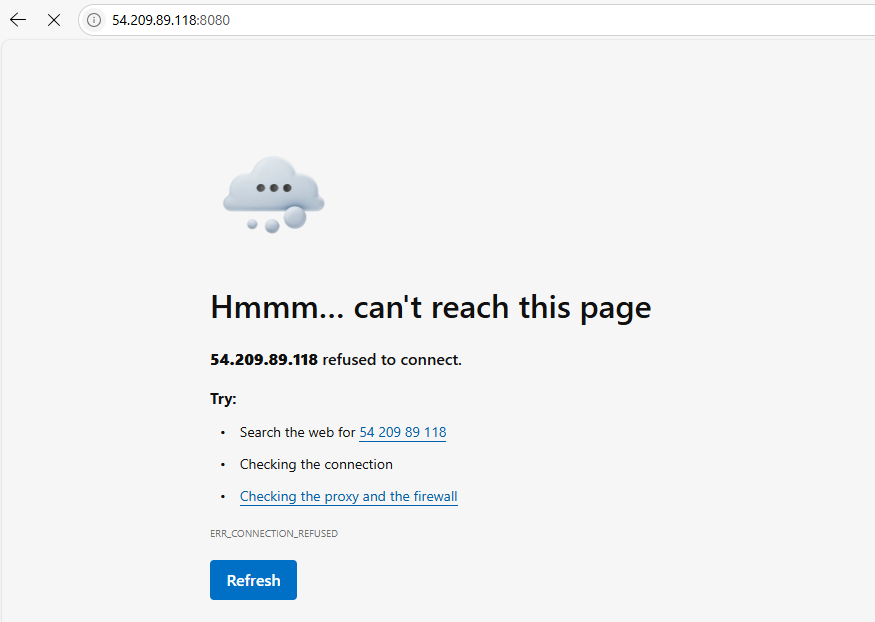
To check java version command is Java –version (check this correct or not)

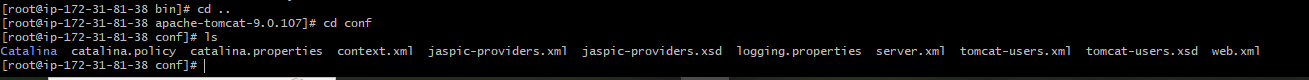
Yum Install Java

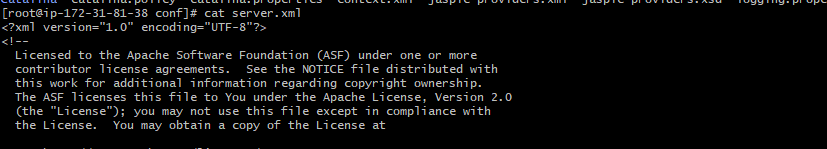






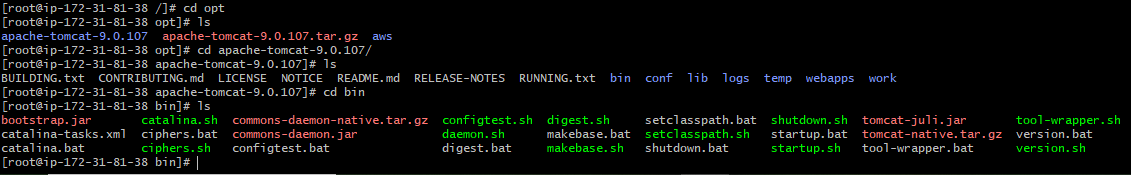






8080

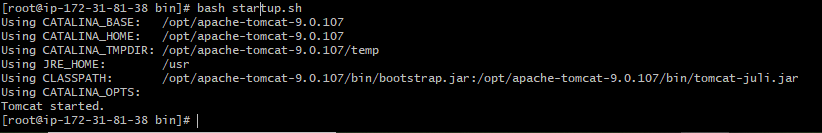
8082



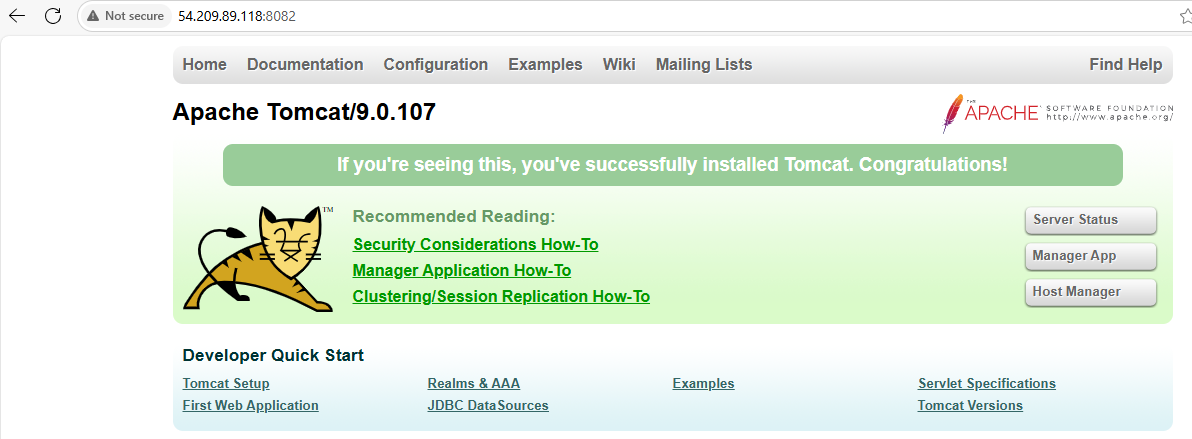
[root@ip-172-31-81-38 bin]# bash startup.sh

To restart bcz we don’t have systemctl here or apache tomcat

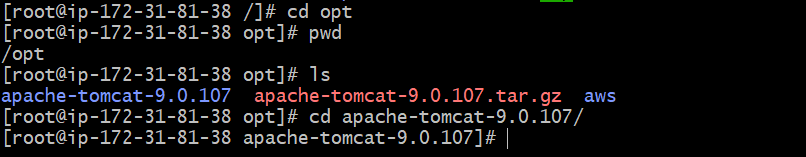
So first need to start and shutdown and start again after any port changes

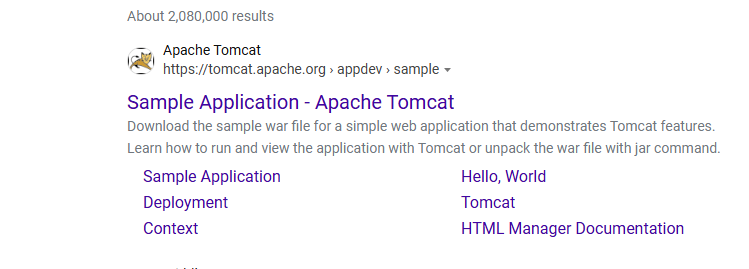


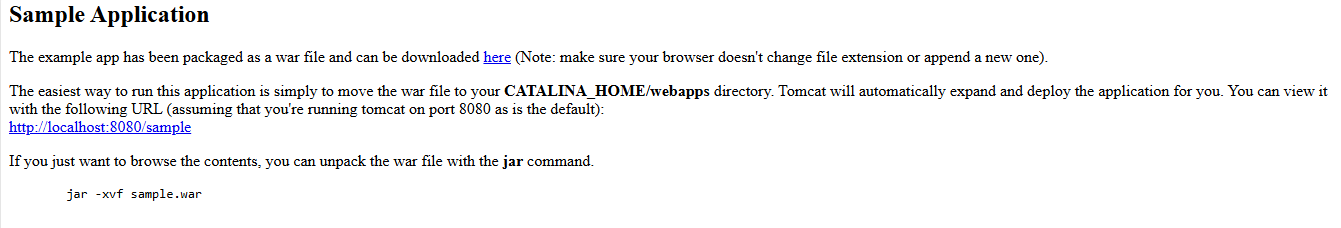
Now started

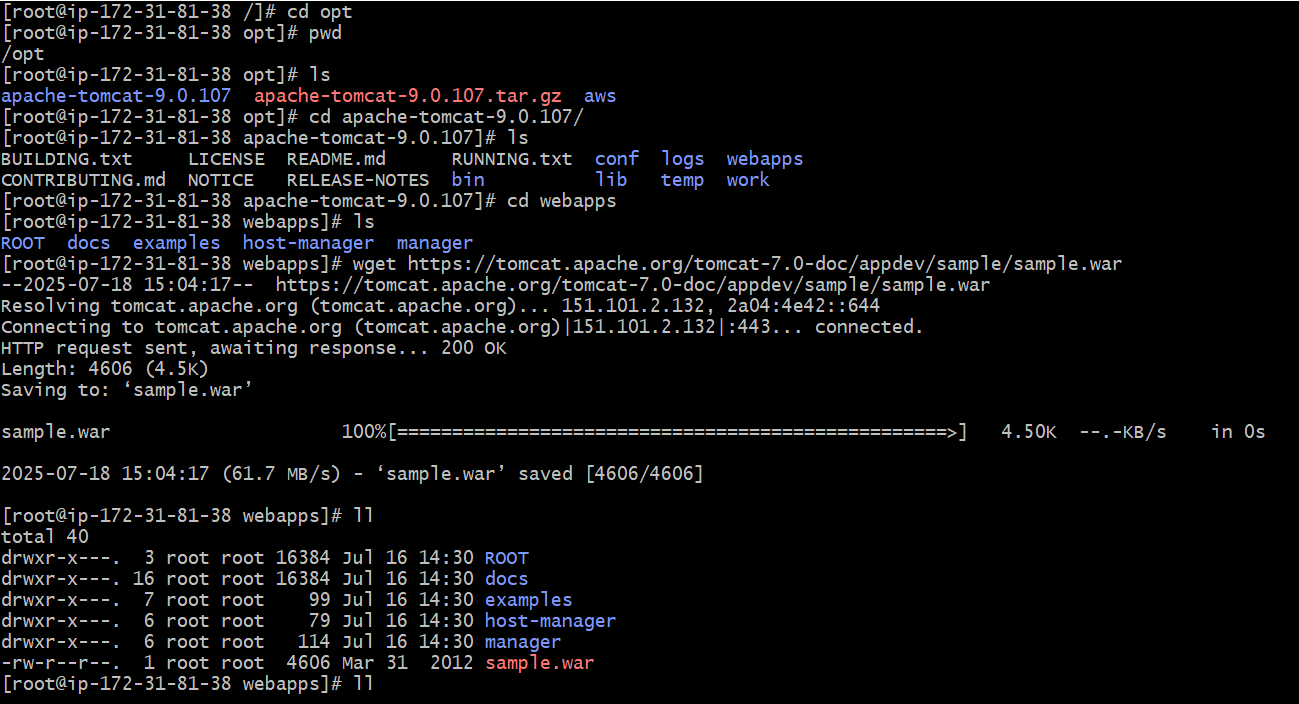


6) Deploy a sample app on webapps

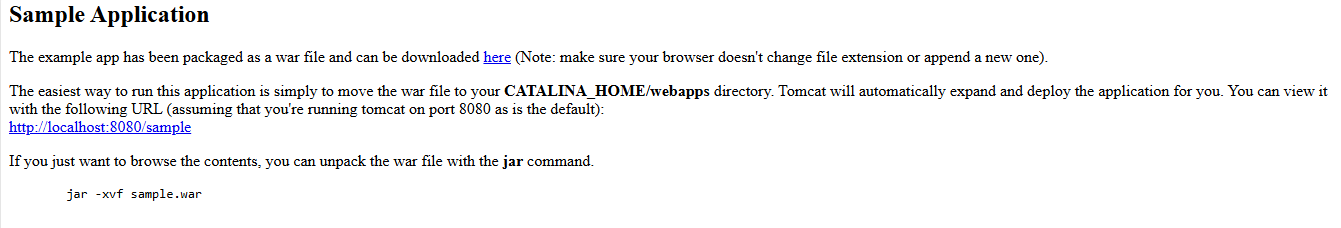


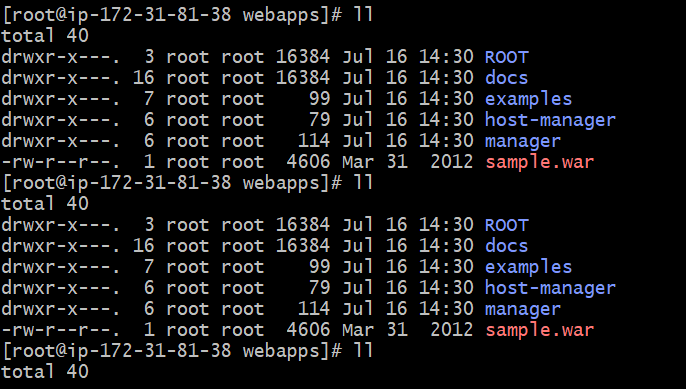






Wget http: web page copy from here



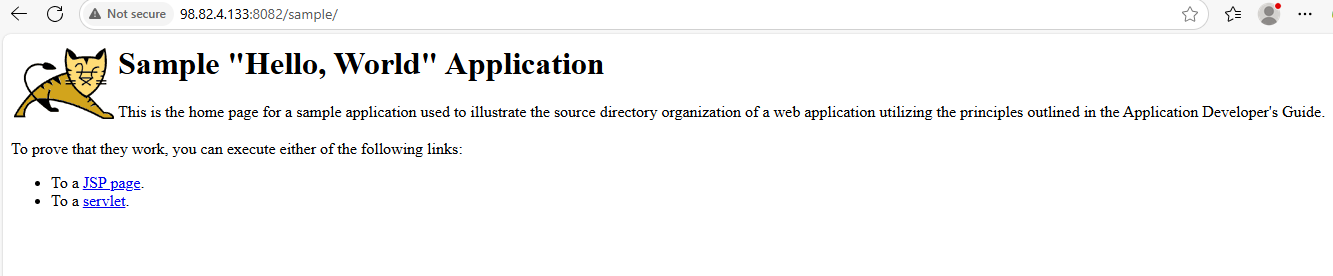


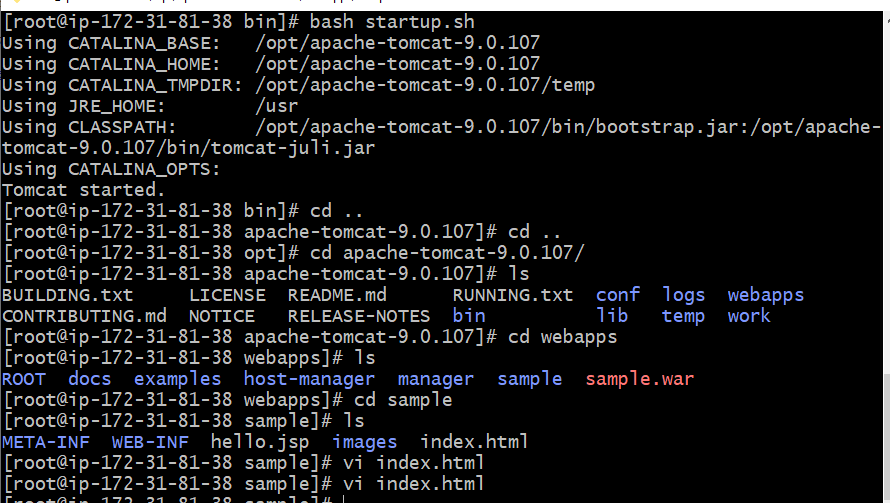
Here in the above sample.war has to change into sample file blue.

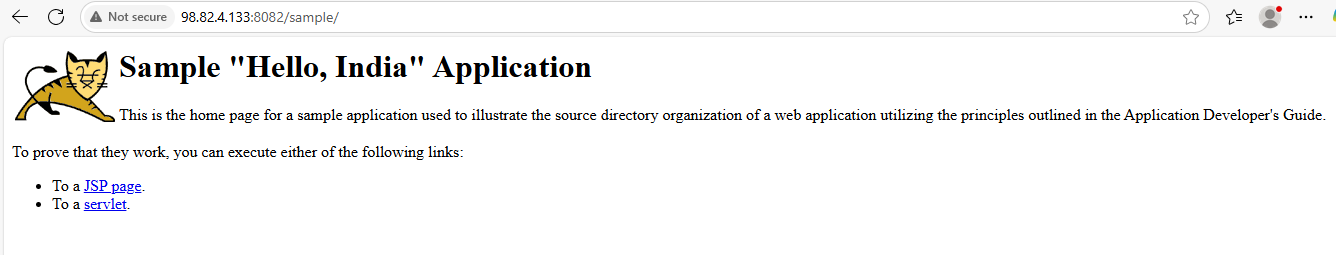
After getting the sample file you need check in the webbrowser by entering ipaddress along with port no of tomcat.

But for me it is not coming so check the doubt may be need to shutdown tomcat and start tomcat again and execute command wget along link of sample file .

Then again check in webbrowser by entering ipaddress along with port no of tomcat.







7 question

ini

CopyEdit

[Unit]

Description=Apache Tomcat Web Application Container

After=network.target

[Service]

Type=forking

Environment=JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk # Adjust based on your Java path

Environment=CATALINA\_PID=/opt/tomcat/temp/tomcat.pid

Environment=CATALINA\_HOME=/opt/tomcat

Environment=CATALINA\_BASE=/opt/tomcat

Environment='CATALINA\_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'

Environment='JAVA\_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'

ExecStart=/opt/tomcat/bin/startup.sh

ExecStop=/opt/tomcat/bin/shutdown.sh

User=tomcat

Group=tomcat

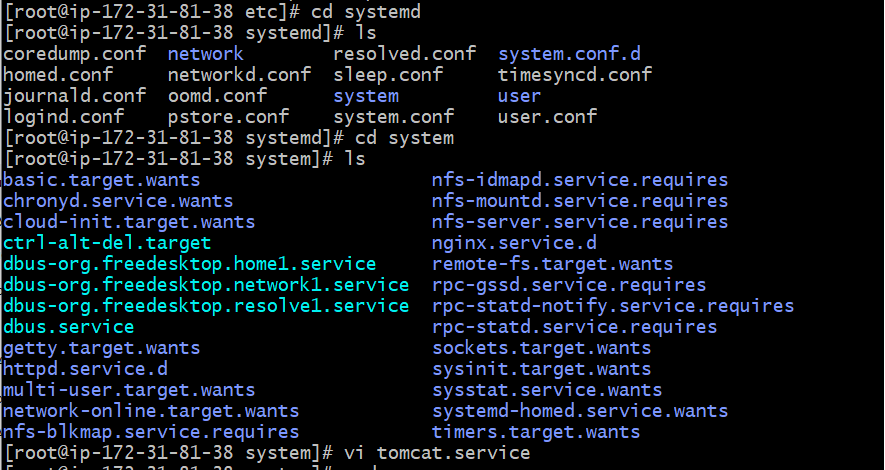
UMask=0007

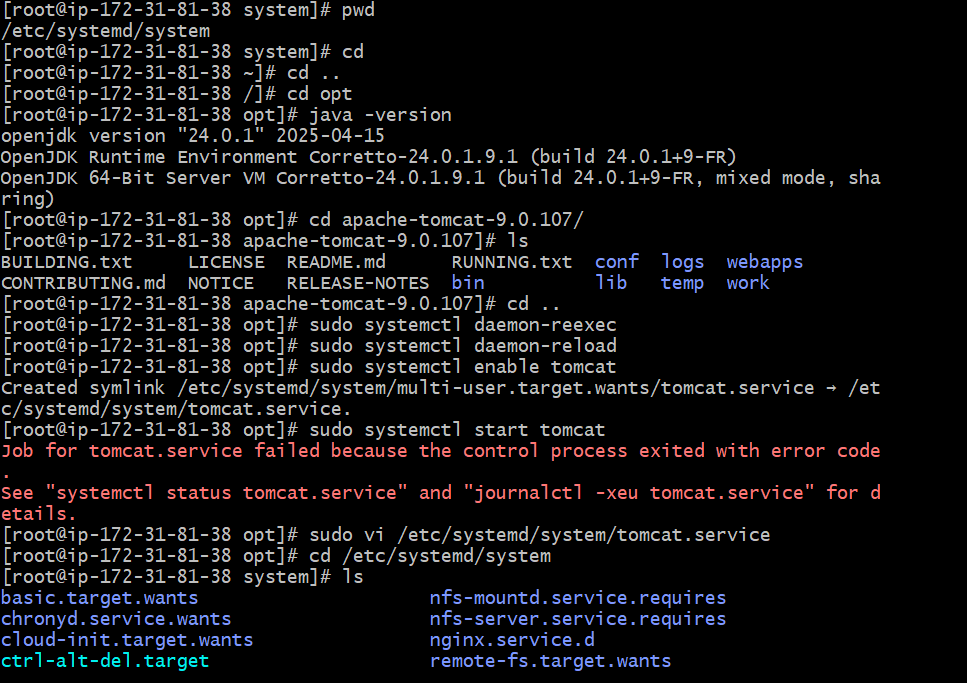
RestartSec=10

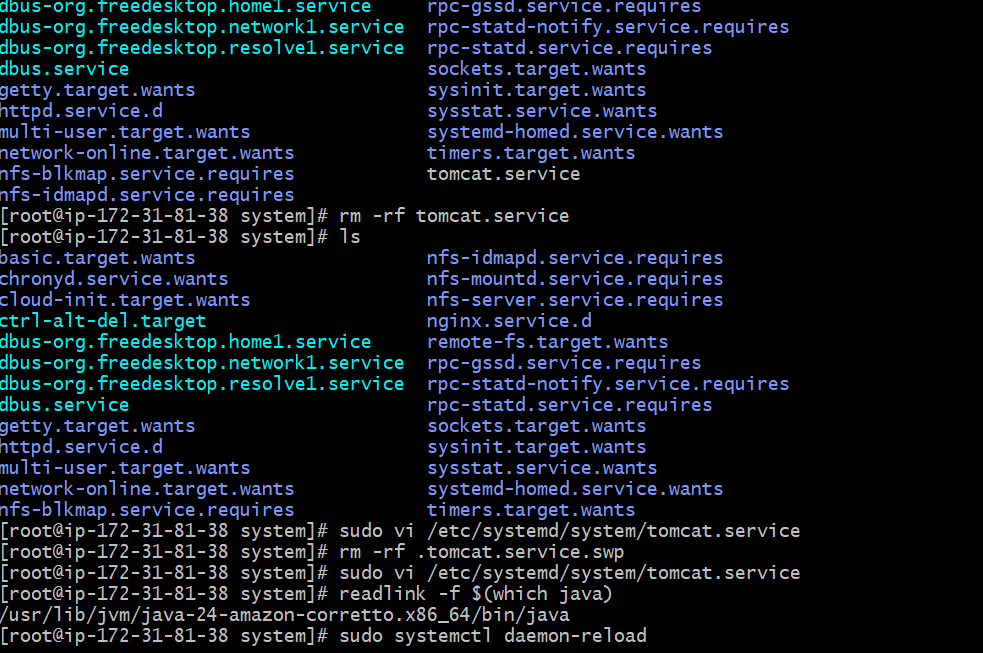
Restart=always

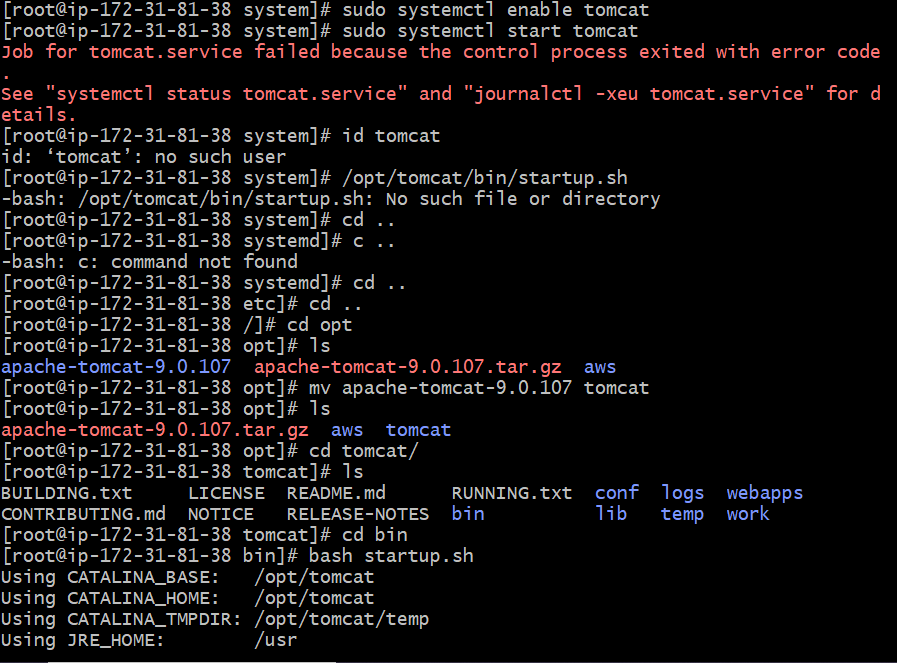
[Install]

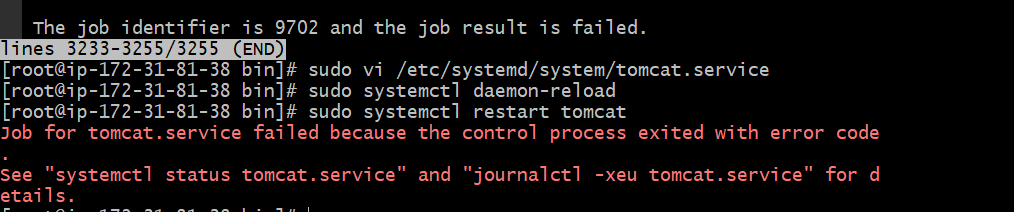
WantedBy=multi-user.target











To all the servers list of the user or my consisting of ports



**🔹 What is HAProxy?**

**HAProxy** stands for **High Availability Proxy**. It is a powerful, free, and open-source **TCP/HTTP load balancer** and **reverse proxy** software used to improve the performance, scalability, and availability of web applications.

**✅ What HAProxy Does:**

1. **Distributes client requests** across multiple backend servers (load balancing).
2. **Performs health checks** to detect down/unhealthy servers.
3. **Acts as a single entry point** for your applications.
4. **Improves availability** and **reduces downtime**.
5. Can **secure** services with SSL termination, rate limiting, and DDoS protection.
6. Used as a **reverse proxy**, forwarding traffic to internal services while hiding backend IPs.

**🔧 Example Use Case:**

You have 4 servers running a website. HAProxy can:

* Listen on port 80 (HTTP)
* Distribute incoming traffic equally across those 4 servers
* Automatically remove any server from rotation if it becomes unresponsive

**🌐 Who Uses HAProxy?**

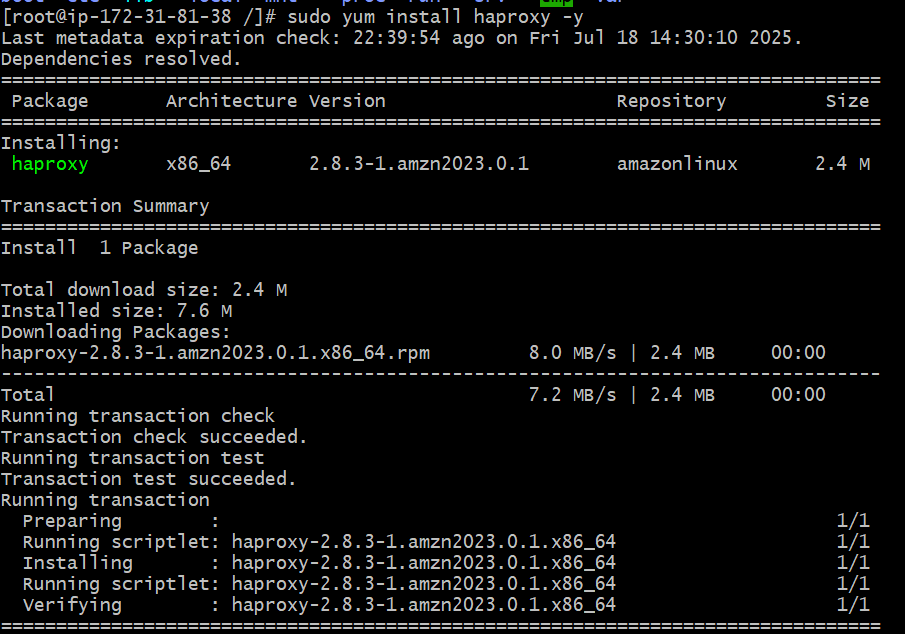
* Companies like **GitHub**, **Airbnb**, **Reddit**, **Trello**, and **Stack Overflow**
* Often used in **cloud**, **microservices**, and **container environments**

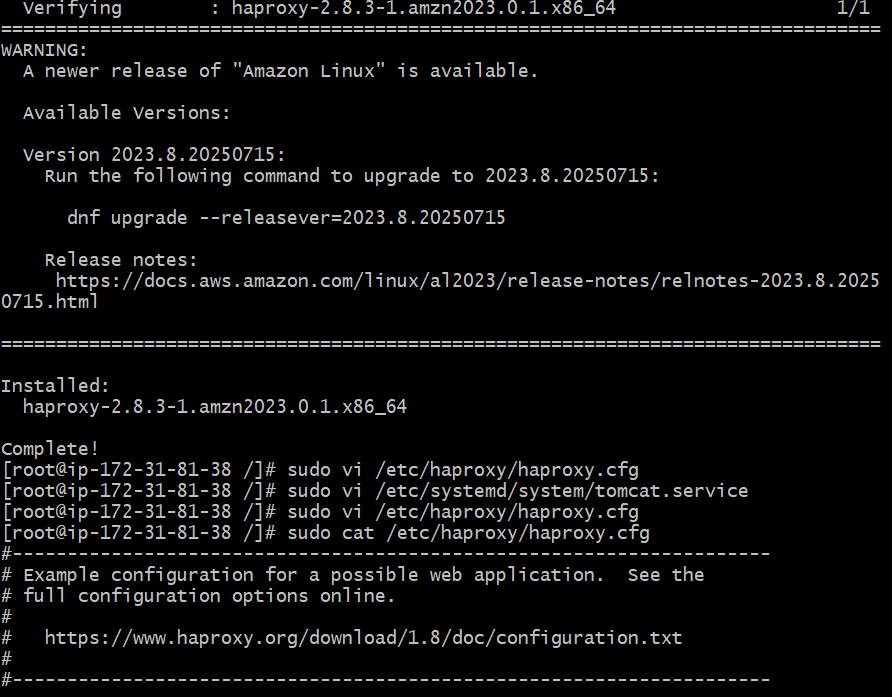
**🔥 Features:**

| **Feature** | **Description** |
| --- | --- |
| Load Balancing | Round-robin, least connections, source hashing, etc. |
| Health Checks | Ensures traffic only goes to healthy servers |
| SSL Termination | Handles HTTPS and decrypts before passing to backend |
| Logging & Stats | Detailed request logs and real-time monitoring |
| Layer 4 & Layer 7 | Works on both TCP (Layer 4) and HTTP (Layer 7) |

**📌 Common Ports**

| **Protocol** | **Port** | **Purpose** |
| --- | --- | --- |
| HTTP | 80 | Public web access |
| HTTPS | 443 | Secure web access |
| Custom | any | Forward to backends |
|  |  |  |



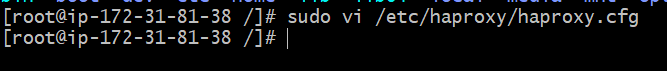


HAPROXY has installed

To configure a HAPROXY goto the below command

sudo vi /etc/haproxy/haproxy.cfg

and copy the below configuration of chatgpt into file.



global

log /dev/log local0

log /dev/log local1 notice

daemon

defaults

log global

mode http

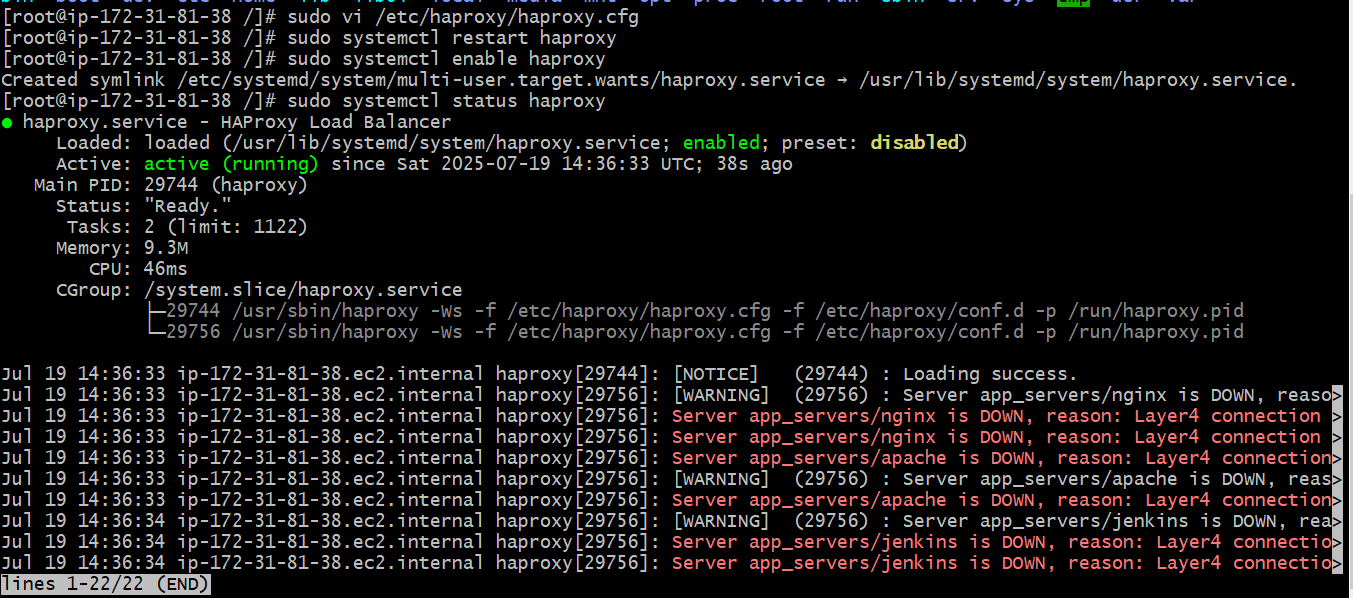
option httplog

option dontlognull

timeout connect 50000

timeout client 50000

timeout server 50000



frontend http\_front

bind \*:80

default\_backend app\_servers

backend app\_servers

balance roundrobin

option httpchk

# Pointing to your various services

server nginx 127.0.0.1:81 check

server apache 127.0.0.1:82 check

server tomcat 127.0.0.1:8082 check

server jenkins 127.0.0.1:8080 check