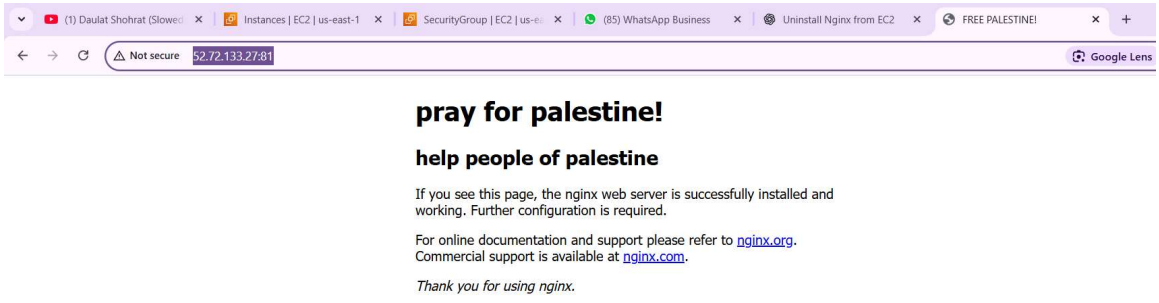
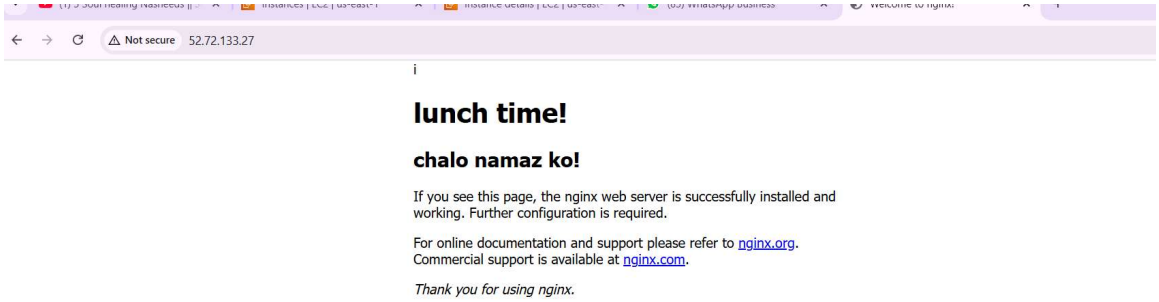
1) Install nginx and run nginx on port number 81.



yum insatll nginx -y
cd /etc/nginx
vi nginx.conf
listen port:81

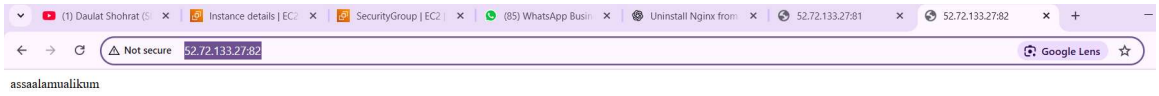
2) Deploy a sample index.html file on nginx.



yum install nginx -y
systemctl start nginx
cd /usr/share/nginx/html/

echo "welcome to techie horizon:>index.html

3) Install Apache and run Apache on port number 82

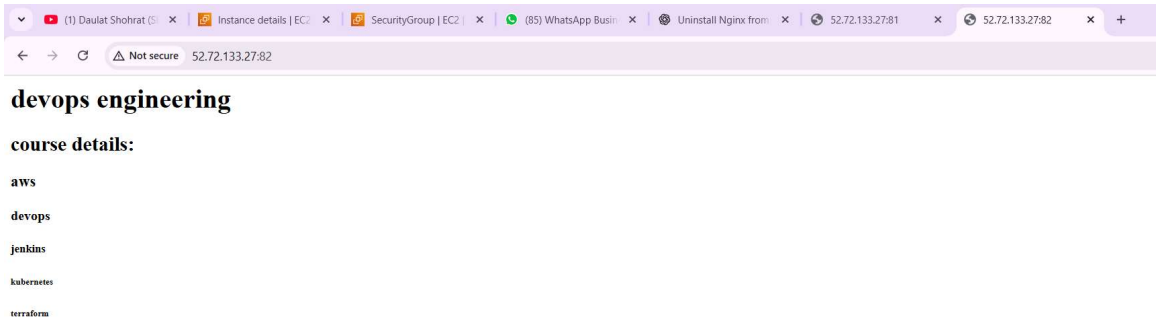


yum install httpd -y

systemctl start httpd

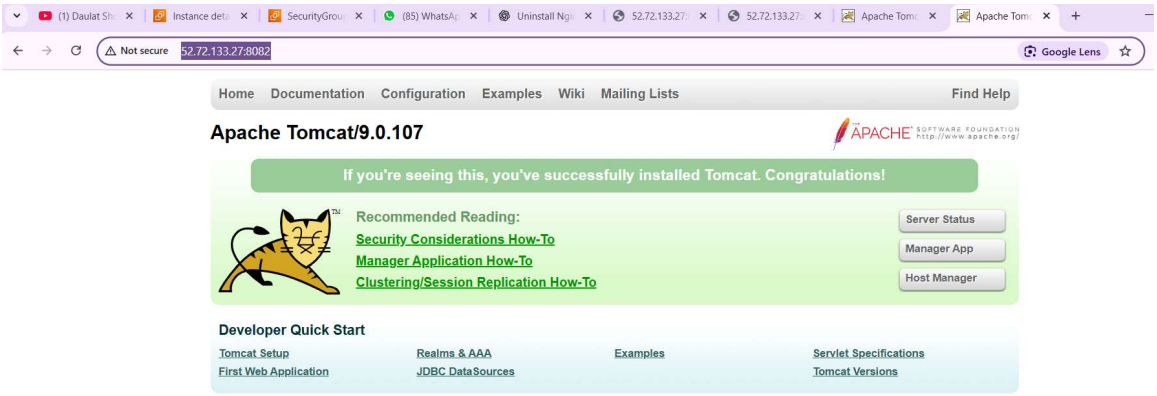
cd /etc/httpd/conf/
vi httpd.conf
change port @ listen port:80 to 82

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



yum install httpd -y
systemctl start httpd
cd /var/www/html
echo "hello everyone from MOHD SAIF UDDIN" > index.html

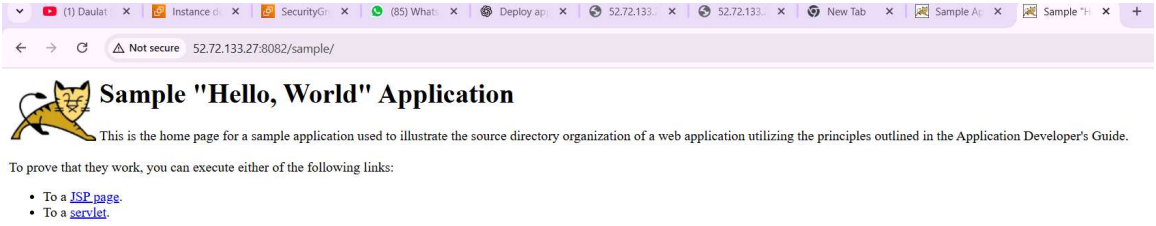
5) Install Apache tomcat on port number 8082



cd /opt/tomcat/conf/

vi server.xml
connector port :8080 to 8082
allow port access in security group as tcp:8082 and
restart the tomcat and able to access the tomcat with different port

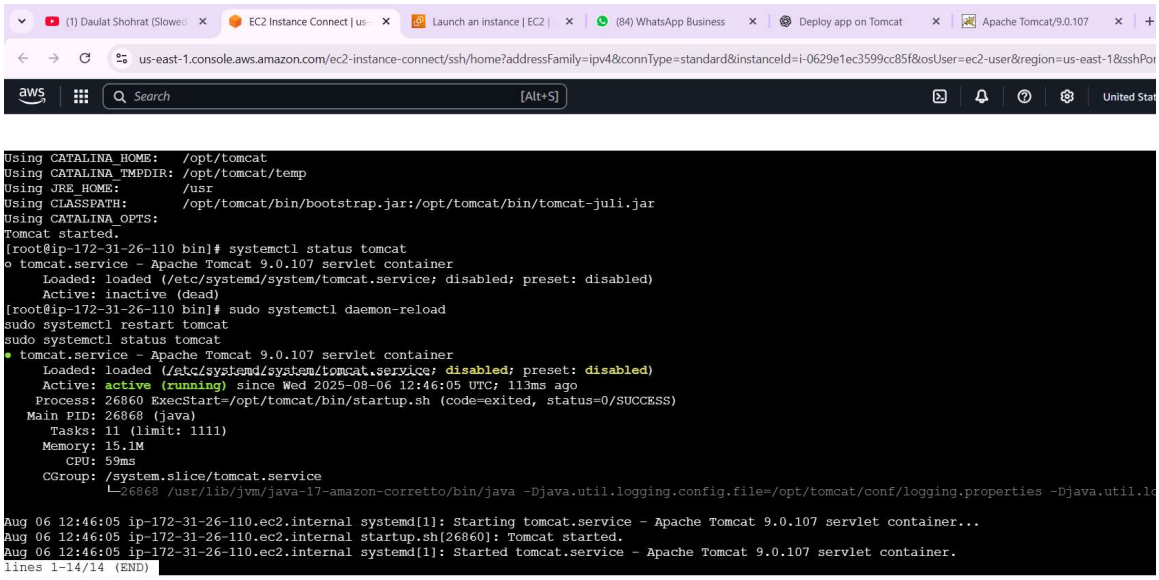
6) Deploy a sample app on webapps



cd /tomcat/webapps
wget <sample file url>

cd unzip sample file
vi index.html

7) Create a tomcat.service file for tomcat.



commands
yum update
yum install java-17 -y

cd opt

wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.107/bin/apache-tomcat-9.0.107.tar.gz
tar xvf tomcat.tar(extract file)
mv tomcat-file-name tomcat
chown -R ec2-user:ec2-user tomcat
chmod 777 /opt/tomcat/bin/startup.sh shutdown.sh
touch /etc/systemd/system/tomcat.service
vi tomcat.service(paste the code)

[Unit]

Description=Apache Tomcat 9.0.107 servlet container

After=network.target

[Service]

Type=forking

User=ec2-user

Group=ec2-user

Environment=JAVA_HOME=/usr/lib/jvm/java-17-amazon-corretto

Environment=CATALINA_HOME=/opt/tomcat

Environment=CATALINA_BASE=/opt/tomcat

Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid

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

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

Restart=on-failure

TimeoutSec=30

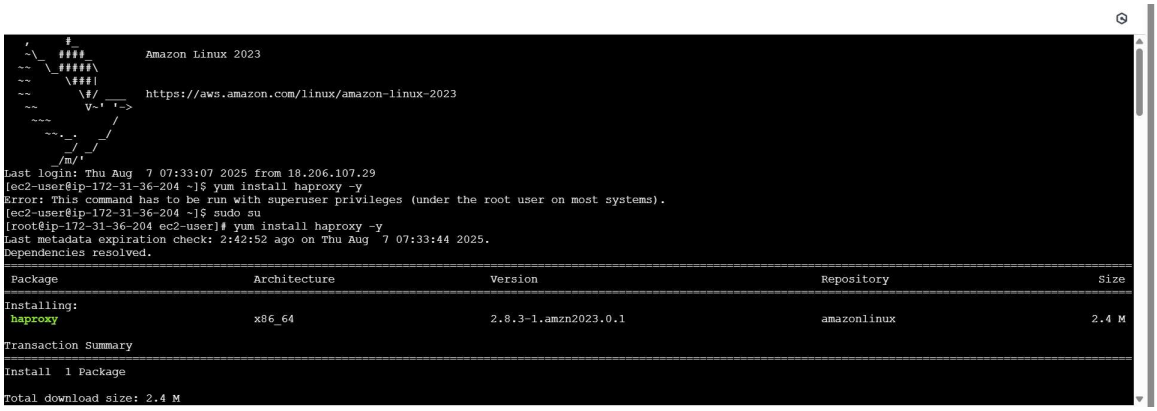
[Install]

WantedBy=multi-user.target

---

sh startup.sh
systemctl daemon-reload(to give daemon permissions)
systemctl status tomcat
systemctl start tomcat
(hence we can access the tomcat by using systemctl command)

8) Configure HA Proxy server



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Thu Aug  7 07:33:07 2025 from 18.206.107.29
[ec2-user@ip-172-31-36-204 ~]$ yum install haproxy -y
Error: This command has to be run with superuser privileges (under the root user on most systems).
[ec2-user@ip-172-31-36-204 ~]$ sudo su
[root@ip-172-31-36-204 ec2-user]# yum install haproxy -y
Last metadata expiration check: 2:42:52 ago on Thu Aug  7 07:33:44 2025.
Dependencies resolved.
================================================================================
 Package              Architecture          Version                  Repository      Size
================================================================================
Installing:
 haproxy              x86_64                2.8.3-1.amzn2023.0.1     amazonlinux     2.4 M

Transaction Summary
================================================================================
Install  1 Package

Total download size: 2.4 M
```

```
[root@ip-172-31-42-127 haproxy]# ping server1
PING server1 (52.87.210.206) 56(84) bytes of data.
64 bytes from server1 (52.87.210.206): icmp_seq=1 ttl=126 time=0.450 ms
64 bytes from server1 (52.87.210.206): icmp_seq=2 ttl=126 time=0.401 ms
64 bytes from server1 (52.87.210.206): icmp_seq=3 ttl=126 time=0.376 ms
64 bytes from server1 (52.87.210.206): icmp_seq=4 ttl=126 time=0.385 ms
^C
--- server1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3093ms
rtt min/avg/max/mdev = 0.376/0.403/0.450/0.028 ms
[root@ip-172-31-42-127 haproxy]# ping server2
PING server2 (13.217.27.172) 56(84) bytes of data.
64 bytes from server2 (13.217.27.172): icmp_seq=1 ttl=126 time=0.215 ms
64 bytes from server2 (13.217.27.172): icmp_seq=2 ttl=126 time=0.252 ms
64 bytes from server2 (13.217.27.172): icmp_seq=3 ttl=126 time=0.286 ms
^C
--- server2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2069ms
rtt min/avg/max/mdev = 0.215/0.251/0.286/0.029 ms
[root@ip-172-31-42-127 haproxy]# pws
bash: pws: command not found
[root@ip-172-31-42-127 haproxy]# pwd
/etc/haproxy
[root@ip-172-31-42-127 haproxy]# ls
conf.d  haproxy.cfg
[root@ip-172-31-42-127 haproxy]# vi haproxy.cfg
[root@ip-172-31-42-127 haproxy]# systemctl restart haproxy
[root@ip-172-31-42-127 haproxy]#
```

```
See "systemctl status haproxy.service" and "journalctl -xeu haproxy.service" for details.
[root@ip-172-31-42-127 haproxy]# vi haproxy.cfg
[root@ip-172-31-42-127 haproxy]# systemctl start haproxy
[root@ip-172-31-42-127 haproxy]# systemctl status haproxy
● haproxy.service - HAProxy Load Balancer
     Loaded: loaded (/usr/lib/systemd/system/haproxy.service; enabled; preset: disabled)
     Active: active (running) since Thu 2025-08-07 12:47:16 UTC; 9s ago
    Process: 25841 ExecStartPre=/usr/sbin/haproxy -f $CONFIG -f $CFGDIR -c -q $OPTIONS (code=exited, status=0/SUCCESS)
   Main PID: 25843 (haproxy)
     Status: "Ready."
      Tasks: 3 (limit: 1057)
     Memory: 6.3M
        CPU: 38ms
     CGroup: /system.slice/haproxy.service
             ├─25843 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -f /etc/haproxy/conf.d -p /run/haproxy.pid
             └─25845 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -f /etc/haproxy/conf.d -p /run/haproxy.pid
```

```
#---------------------------------------------------------------------
frontend main
    bind *:80
    acl url_static       path_beg       -i /static /images /javascript /stylesheets
    acl url_static       path_end       -i .jpg .gif .png .css .js

    use_backend static          if url_static
    default_backend             app

#---------------------------------------------------------------------
# static backend for serving up images, stylesheets and such
#---------------------------------------------------------------------
backend static
    balance     roundrobin
    server      static 127.0.0.1:4331 check

#---------------------------------------------------------------------
# round robin balancing between the various backends
#---------------------------------------------------------------------
backend app
    balance     roundrobin
    server  app1 127.0.0.1:5001 check
    server  app2 127.0.0.1:5002 check
    server  app3 127.0.0.1:5003 check
    server  app4 127.0.0.1:5004 check
    server  app5 35.168.18.53:81 check
    server  app6 184.73.86.137:82 check
```

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost6 localhost6.localdomain6
3.84.78.79      haproxy
~
```

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost6 localhost6.localdomain6
3.84.78.79      haproxy
~
```

steps:
deploy a httpd server
deploy a nginx server
deploy a tomcat server
install a HAPROXY server
yum install haproxy
systemctl enable haproxy
cat /etc/hosts (edit file and paste public ips of two servers with names httpdserver1:public ip nginxserver:public ip)
:wq!

vi /etc/haproxy/haproxy.cfg (paste public ips with port 80 extension of two servers also change listen port to 80 also change bind port to 80)

bind:80

server app5 nginxpublicip:port check
server app6 httpdpublicip:port check

systemctl restart haproxy
systemctl status haproxy

go to nginx server
vi /etc/hosts (paste public ip of haproxy with name)

public ip haproxy

go to apache server
vi /etc/hosts(paste public ip with name)

public ip haproxy