

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

- Step -1 Installed nginx using command in gitbash - `yum install nginx`

```
total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
Running scriptlet: nginx-filesystem-1:1.28.0-1.amzn2023.0.2.noarch
Installing : nginx-filesystem-1:1.28.0-1.amzn2023.0.2.noarch
Installing : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch
Installing : libunwind-1.4.0-5.amzn2023.0.3.x86_64
Installing : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
Installing : nginx-core-1:1.28.0-1.amzn2023.0.2.x86_64
Installing : nginx-1:1.28.0-1.amzn2023.0.2.x86_64
Running scriptlet: nginx-1:1.28.0-1.amzn2023.0.2.x86_64
Verifying : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
Verifying : libunwind-1.4.0-5.amzn2023.0.3.x86_64
Verifying : nginx-1:1.28.0-1.amzn2023.0.2.x86_64
Verifying : nginx-core-1:1.28.0-1.amzn2023.0.2.x86_64
Verifying : nginx-filesystem-1:1.28.0-1.amzn2023.0.2.noarch
Verifying : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Installed:
gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64      libunwind-1.4.0-5.amzn2023.0.3.x86_64
nginx-1:1.28.0-1.amzn2023.0.2.x86_64            nginx-core-1:1.28.0-1.amzn2023.0.2.x86_64
nginx-filesystem-1:1.28.0-1.amzn2023.0.2.noarch  nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Complete!
root@ip-172-31-17-3-ec2-user:~#
```

- Step -2 Found the path using command - `find / -name nginx.conf`
- Step -3 Path - `/etc/nginx/nginx.conf`
- Step -3 To edit command – `vi /etc/nginx/nginx.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;

    error_page 404 /404.html;
    location = /404.html {
    }

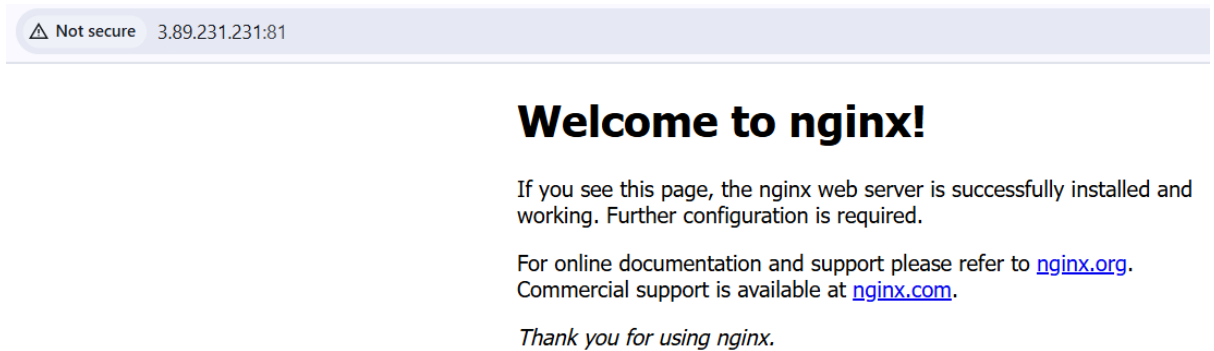
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}
```

- Settings for a TLS enabled server.
- Step 4 – search for server column and change port number from 80 to 81
- Step 5 changed inbound rules in EC2 in security and added port 81

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range
<input type="checkbox"/>	-	sgr-0300c21c0524daa90	IPv4	SSH	TCP	22
<input type="checkbox"/>	-	sgr-03a69f321545c54a3	IPv4	HTTP	TCP	80
<input type="checkbox"/>	-	sgr-0ecadd3db46b51bb2	-	All traffic	All	All
<input type="checkbox"/>	-	sgr-00ae591796a8e8439	IPv4	Custom TCP	TCP	81

- Step 6 – restart nginx command – `systemctl restart nginx`
- Step 7 – for confirmation check browser and entered public ip (from ec2) by adding port number 81 i.e (3.89.231.231:81)
- Step 8 – restart nginx in terminal using command – `systemctl restart nginx`

- Step 9 – output



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

- Step 1 – go to the location - /usr/share/nginx/html
- Step 2 – check for the file index.html by using command - ls

```
[root@ip-172-31-17-3 html]# cd /usr/share/nginx/html
[root@ip-172-31-17-3 html]# ls
404.html  icons  nginx-logo.png
50x.html  index.html  poweredby.png
```

- step 3 – to edit the index.html file use command – vi index.html
- step 4 – change the content in body ( Deployment of index.html in nginx)

```
</style>
</head>
<body>
<h1>!Deployment of index.html in nginx| </h1>
<p>If you see this page, the nginx web server is succes
sfully installed and
```

- Step 5 - restart the nginx command – systemctl restart nginx
- Step 6 – open browser refresh the page



-

### 3) Install Apache and run Apache on port number 82

- Step 1 - I'm using gitbash of I installed httpd , command – yum install httpd

```
[root@ip-172-31-17-3 ec2-user]# yum install httpd
Last metadata expiration check: 3:28:16 ago on Wed Sep 24 06:42:41 2025.
Package httpd-2.4.65-1.amzn2023.0.1.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
complete!
```

- Step 2- Find file httpd.conf by command - find / -name httpd.conf

```
[root@ip-172-31-17-3 ec2-user]# find / -name httpd.conf
/etc/httpd/conf/httpd.conf
/usr/lib/tmpfiles.d/httpd.conf
/usr/lib/sysusers.d/httpd.conf
[root@ip-172-31-17-3 ec2-user]#
```

- Step 3 – edit the file command – vi /etc/httpd/conf/httpd.conf

- Step 4 – changed port 80 to 82

```
Listen: Allows you to bind Apache to specific IP addresses and/or
ports, instead of the default. See also the <VirtualHost>
directive.

Change this to Listen on a specific IP address, but note that if
httpd.service is enabled to run at boot time, the address may not be
available when the service starts. See the httpd.service(8) man
page for more information.

Listen 12.34.56.78:80
Listen 82
```

- Step 5 – To save changes command – esc:wq! And press enter

- Step 6 – Add inbound rules in EC2 security group port number – 82

Inbound rules (4)								Manage tags	Edit inbound rules
Search								< 1 > ⚙	
<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range			
<input type="checkbox"/>	-	sgr-0300c21c0524daa90	IPv4	SSH	TCP	22			
<input type="checkbox"/>	-	sgr-03a69f321545c54a3	IPv4	HTTP	TCP	80			
<input type="checkbox"/>	-	sgr-0ecadd3db46b51bb2	-	All traffic	All	All			
<input type="checkbox"/>	-	sgr-00ae591796a8e8439	IPv4	Custom TCP	TCP	82			

- Step 7 – open browser and verify using public ip from EC2 and adding 82 port number 3.89.231.231:82

- Step 8 – refresh the browser page

← → ↻ ⚠ Not secure 3.89.231.231:82

## Deployment of index.html in nginx

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

Step 1- Find httpd using – find / -name httpd

Step 2 – path to location – cd /var/www/html

Step 3 – create index.html file using command touch index.html

```
root@ip-172-31-17-3 ec2-user]# touch index.html
```

Step 4 - enter content in the file command vi – vi index.html

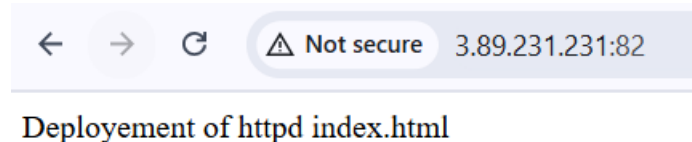
```
root@ip-172-31-17-3:/var/www/html
```

```
</h1> Deployment of httpd index.html </h1>
```

Step 5 – to save command – esc+:wq!

Step 6 – restart httpd by command – systemctl restart httpd

Step 6 – open browser search for public from ec2 by adding port 82 i.e  
(3.89.231.231:82)



#### 5 ) install Apache tomcat on port number 8082

Step 1 – install apache tomcat command – wget - <https://dlcdn.apache.org/tomcat/tomcat-11/v11.0.11/bin/apache-tomcat-11.0.11.tar.gz>

Step 2 – to extract files – tar -xzf apache-tomcat-11.0.11.tar.gz

Step 3 – download java – yum install java

Step 4 – setup service file - sudo nano /etc/systemd/system/tomcat.service

Step 5 – enter [Unit]

Description=Apache Tomcat Web Application Container

After=network.target

[Service]

Type=forking

User=root

Group=root

Environment="JAVA\_HOME=/usr/lib/jvm/java-24"

Environment="CATALINA\_HOME=/opt/tomcat"

Environment="CATALINA\_BASE=/opt/tomcat"

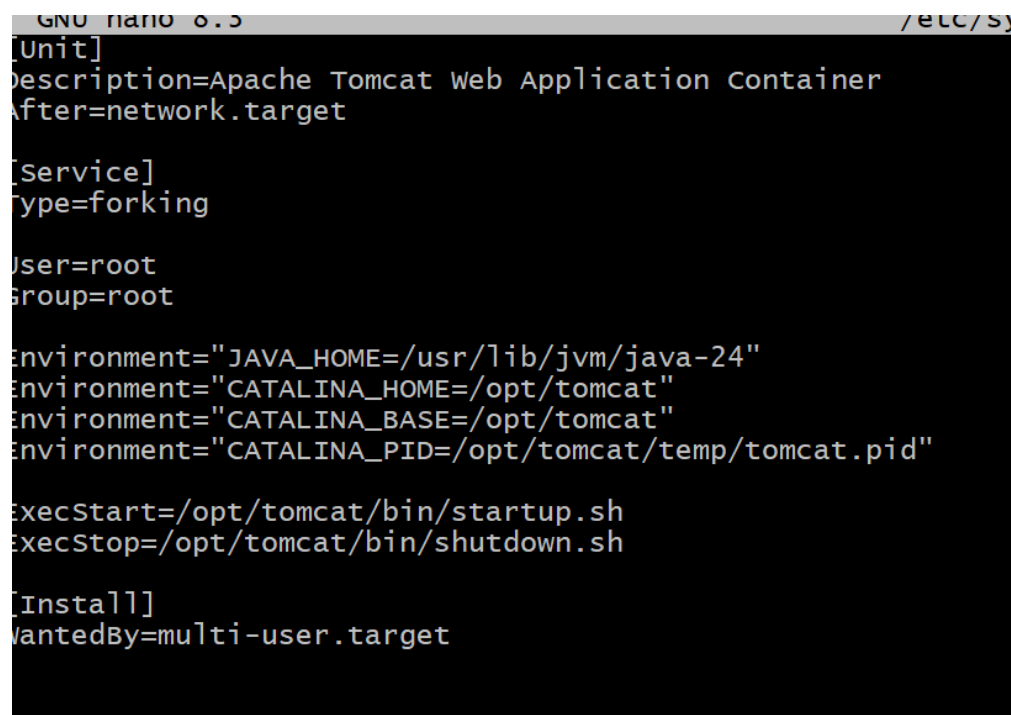
Environment="CATALINA\_PID=/opt/tomcat/temp/tomcat.pid"

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

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

[Install]

WantedBy=multi-user.target

A screenshot of a terminal window with a dark background. The terminal shows the configuration of a systemd unit file for Apache Tomcat. The text is as follows:

```
GNU nano 8.5 /etc/s
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target

[Service]
Type=forking

User=root
Group=root

Environment="JAVA_HOME=/usr/lib/jvm/java-24"
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

[Install]
WantedBy=multi-user.target
```

Step 6 -changed a line Envirnoment="JAVA\_HOMR=/usr/lib/jvm/java-24" to  
Environment=JAVA\_HOME=/usr/lib/jvm/java-24-amazon-corretto.x86\_64

```

lines 1-14/14 (END)...skipping...
tomcat.service - Apache Tomcat Web Application Container
Loaded: loaded (/etc/systemd/system/tomcat.service; enabled; preset: disabled)
Active: active (running) since Wed 2025-09-24 12:45:11 UTC; 83ms ago
Process: 45000 ExecStart=/opt/apache-tomcat-11.0.11/bin/startup.sh (code=exited, status=0/SUCCESS)
Main PID: 45007 (java)
Tasks: 11 (limit: 1057)
Memory: 16.4M
CPU: 81ms
CGroup: /system.slice/tomcat.service
└─45007 /usr/lib/jvm/java-24-amazon-corretto.x86_64/bin/java -Djava.util.logging.config.file=/opt/apach

Sep 24 12:45:11 ip-172-31-17-3.ec2.internal systemd[1]: Starting tomcat.service - Apache Tomcat Web Application Cont
Sep 24 12:45:11 ip-172-31-17-3.ec2.internal startup.sh[45000]: Tomcat started.
Sep 24 12:45:11 ip-172-31-17-3.ec2.internal systemd[1]: Started tomcat.service - Apache Tomcat Web Application Conta

```

Step 7 – open `/opt/apache-tomcat-11.0.11/conf/server.xml`

```

-->
Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
-->
<Connector port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
           port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
-->

```

Step 8 – changing port 8080 to 8082

```

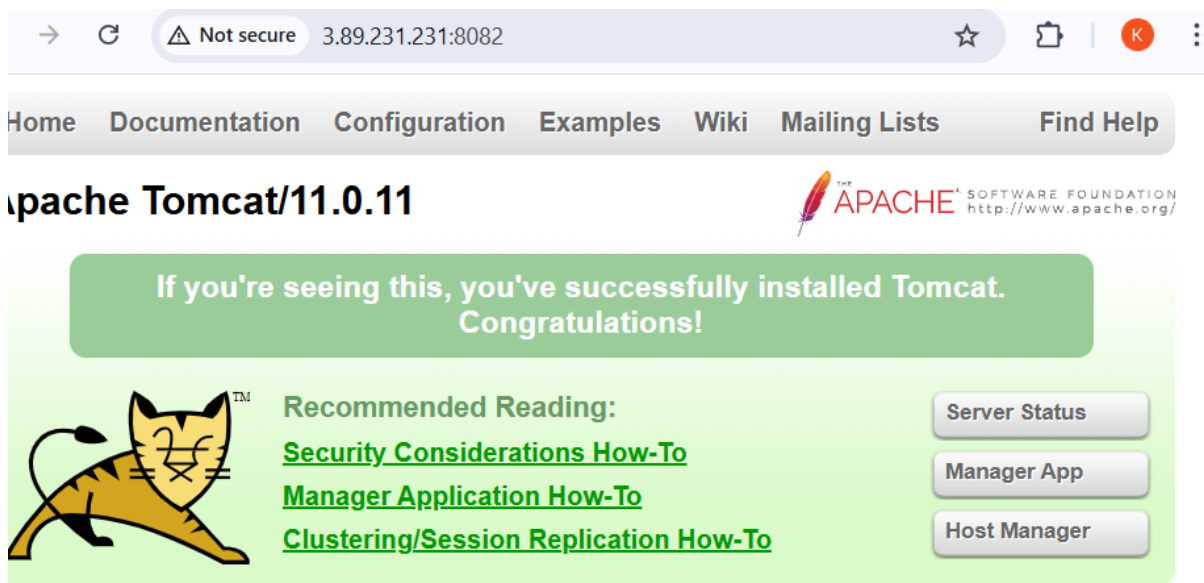
root@ip-172-31-17-3 ec2-user]# ^C
root@ip-172-31-17-3 ec2-user]# sudo nano /opt/apache-tomcat-11.0.11/conf/server.xml
root@ip-172-31-17-3 ec2-user]# ps -ef | grep 8082
root      45686   45418   0 13:03 pts/1    00:00:00 grep --color=auto 8082
root@ip-172-31-17-3 ec2-user]#

```

Step 9 – added inbound rules port 8082 in security groups EC2

Port range
22
8080
80
All
8082

Step 10 – verify in browser using public ip from ec2 and add 8082 to id and fetch



## 6) Deploy a sample app on webapps

Step 1 - path `cd /opt/apache-tomcat-11.0.11/webapps`

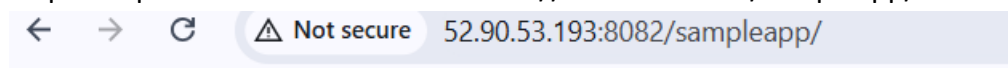
step 2 – created sample app - `sudo mkdir /opt/apache-tomcat-11.0.11/webapps/sampleapp`

`echo "<h1>Hello from Tomcat Sample App!</h1>" | sudo tee /opt/apache-tomcat-11.0.11/webapps/sampleapp/index.jsp`

```
rw-r--r--. 1 root root 0 Sep 26 07:06 index.jsp
[root@ip-172-31-17-3 webapps]# echo '<h1>Hello from Tomcat Sample App!</h1>' | sudo tee /
opt/apache-tomcat-11.0.11/webapps/sampleapp/index.jsp
<h1>Hello from Tomcat Sample App!</h1>
[root@ip-172-31-17-3 webapps]# vi index.jsp
```

step 3 – restart tomcat – `systemctl restart tomcat`

step 4 – open browser and enter – `HTTP://52.90.53.8082/sampleapp/`



# Hello from Tomcat Sample App!

## 7) Create a tomcat.service file for tomcat.

Step 1 – create service file

Command - `sudo nano /etc/systemd/system/tomcat.service`

Enter this command to rewrite and save it by using `ctrl+O` and hit enter

```
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target

[Service]
Type=forking

# Update JAVA_HOME and CATALINA_HOME as per your installation
Environment=JAVA_HOME=/usr/lib/jvm/java-24-amazon-corretto.x86_64
Environment=CATALINA_HOME=/opt/apache-tomcat-11.0.11
Environment=CATALINA_BASE=/opt/apache-tomcat-11.0.11
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'

ExecStart=/opt/apache-tomcat-11.0.11/bin/startup.sh
ExecStop=/opt/apache-tomcat-11.0.11/bin/shutdown.sh

User=ec2-user
Group=ec2-user

Restart=always

[Install]
WantedBy=multi-user.target
```

Step 2 – reload system - `sudo systemctl daemon-reexec`

`sudo systemctl daemon-reload`

step 3 – system start tomcat

step 4 – verify - <http://52.90.53193:8082/sampleapp/>

```
[root@ip-172-31-17-3 webapps]#
^[[201~[root@ip-172-31-17-3 websudo netstat -tulnp | grep java
# or
sudo lsof -i:8082
tcp6      0      0 :::8082                :::*                    LISTEN      2999/java
tcp6      0      0 127.0.0.1:8005         :::*                    LISTEN      2999/java

COMMAND  PID    USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
java     2999  ec2-user  44u  IPv6  21687    0t0  TCP *:us-cl (LISTEN)
```

## 8) Configure HA Proxy server

Step 1 – launched 3 instances ,server1 server2, HA proxy

<input type="checkbox"/>	server_1	i-027e27673edfb6f72	Running	t3.micro	3/3 checks p	3/3 checks p
<input type="checkbox"/>	server_2	i-036bbe1fdd27a3fdc	Running	t3.micro	3/3 checks p	3/3 checks p
<input checked="" type="checkbox"/>	HA_proxy	i-082a6084816416fa0	Running	t3.micro	3/3 checks p	3/3 checks p

Step -2 install httpd

Command – `yum install httpd -y`



```

(8/12): httpd-tools-2.4.65-1.amzn2023.0.1.x86_64 2.3 MB/s | 81 kB 00:00
(9/12): mailcap-2.1.49-3.amzn2023.0.3.noarch.rpm 1.4 MB/s | 33 kB 00:00
(10/12): mod_http2-2.0.27-1.amzn2023.0.3.x86_64 5.3 MB/s | 166 kB 00:00
(11/12): mod_lua-2.4.65-1.amzn2023.0.1.x86_64 2.6 MB/s | 60 kB 00:00
(12/12): libbrotli-1.0.9-4.amzn2023.0.2.x86_64 5.5 MB/s | 315 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
  Installing : apr-1.7.5-1.amzn2023.0.4.x86_64 1/12
  Installing : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 2/12
  Installing : apr-util-1.6.3-1.amzn2023.0.1.x86_64 3/12
  Installing : mailcap-2.1.49-3.amzn2023.0.3.noarch 4/12
  Installing : httpd-tools-2.4.65-1.amzn2023.0.1.x86_64 5/12
  Installing : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 6/12
  Running scriptlet: httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch 7/12
  Installing : httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch 7/12
  Installing : httpd-core-2.4.65-1.amzn2023.0.1.x86_64 8/12
  Installing : mod_http2-2.0.27-1.amzn2023.0.3.x86_64 9/12
  Installing : mod_lua-2.4.65-1.amzn2023.0.1.x86_64 10/12
  Installing : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 11/12
  Installing : httpd-2.4.65-1.amzn2023.0.1.x86_64 12/12
  Running scriptlet: httpd-2.4.65-1.amzn2023.0.1.x86_64 12/12
  Verifying : apr-1.7.5-1.amzn2023.0.4.x86_64 1/12
  Verifying : apr-util-1.6.3-1.amzn2023.0.1.x86_64 2/12
  Verifying : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 3/12
  Verifying : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 4/12
  Verifying : httpd-2.4.65-1.amzn2023.0.1.x86_64 5/12
  Verifying : httpd-core-2.4.65-1.amzn2023.0.1.x86_64 6/12
  Verifying : httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch 7/12
  Verifying : httpd-tools-2.4.65-1.amzn2023.0.1.x86_64 8/12
  Verifying : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 9/12
  Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch 10/12
  Verifying : mod_http2-2.0.27-1.amzn2023.0.3.x86_64 11/12
  Verifying : mod_lua-2.4.65-1.amzn2023.0.1.x86_64 12/12
Installed:
apr-1.7.5-1.amzn2023.0.4.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-2.4.65-1.amzn2023.0.1.x86_64
httpd-core-2.4.65-1.amzn2023.0.1.x86_64
httpd-filesystem-2.4.65-1.amzn2023.0.1.noarch
httpd-tools-2.4.65-1.amzn2023.0.1.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch
mod_http2-2.0.27-1.amzn2023.0.3.x86_64
mod_lua-2.4.65-1.amzn2023.0.1.x86_64
Complete!
[root@ip-172-31-21-213 ~]#

```

Step 3 – vi - /etc/hosts

Add ha\_proxy instance public id as (load balancer)

54.162.161.140 load\_balancer

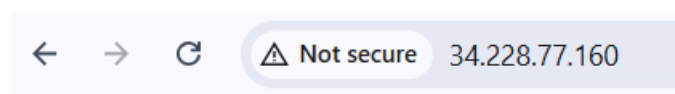
Step 4 – to check load balancer – command – ping load\_balancer -c 4

```

ping: balancer: Name or service not known
[root@ip-172-31-21-213 ~]# ping load_balancer -c 4
PING load_balancer (54.162.161.140) 56(84) bytes of data.

```

Step -5 - now browser with server 1 ip with port 80



**It works!**

## System status nginx

Step 6 - use command – ping load\_balancer1

```
root@ip-172-31-25-96 ~]# ping load_balancer1
PING load_balancer1 (54.162.161.140) 56(84) bytes of data.
64 bytes from 54.162.161.140: icmp_seq=1 ttl=64 time=0.046 ms
64 bytes from 54.162.161.140: icmp_seq=2 ttl=64 time=0.046 ms
64 bytes from 54.162.161.140: icmp_seq=3 ttl=64 time=0.046 ms
64 bytes from 54.162.161.140: icmp_seq=4 ttl=64 time=0.046 ms
64 bytes from 54.162.161.140: icmp_seq=5 ttl=64 time=0.046 ms
64 bytes from 54.162.161.140: icmp_seq=6 ttl=64 time=0.046 ms
64 bytes from 54.162.161.140: icmp_seq=7 ttl=64 time=0.046 ms
64 bytes from 54.162.161.140: icmp_seq=8 ttl=64 time=0.046 ms
64 bytes from 54.162.161.140: icmp_seq=9 ttl=64 time=0.046 ms
64 bytes from 54.162.161.140: icmp_seq=10 ttl=64 time=0.046 ms
--- load_balancer1 ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 17666ms

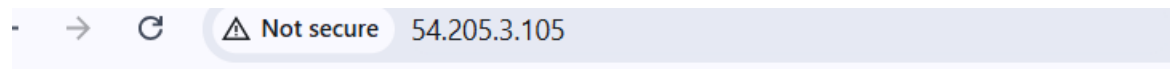
root@ip-172-31-25-96 ~]# systemctl status nginx
nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
   Active: active (running) since Fri 2025-09-26 10:51:14 UTC; 1min 20s ago
     Process: 26181 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
     Process: 26182 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 26183 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 26184 (nginx)
     Tasks: 3 (limit: 1057)
    Memory: 3.2M
       CPU: 58ms
    CGroup: /system.slice/nginx.service
            └─26184 "nginx: master process /usr/sbin/nginx"
              └─26185 "nginx: worker process"
                └─26186 "nginx: worker process"

Sep 26 10:51:14 ip-172-31-25-96.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server: [main]
Sep 26 10:51:14 ip-172-31-25-96.ec2.internal nginx[26182]: nginx: the configuration file /etc/nginx/nginx.conf is not supported by this version of nginx
Sep 26 10:51:14 ip-172-31-25-96.ec2.internal nginx[26182]: nginx: configuration file /etc/nginx/nginx.conf is not supported by this version of nginx
Sep 26 10:51:14 ip-172-31-25-96.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server: [main]

root@ip-172-31-25-96 ~]# ping load_balancer1
PING load_balancer1 (54.162.161.140) 56(84) bytes of data.
```

Step 7 – browse server 2 – public ip by adding port 80

(54.205.3.105:80)



## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](https://nginx.org).  
Commercial support is available at [nginx.com](https://nginx.com).

*Thank you for using nginx.*

## HA-proxy setting

Step 1 – ran this following command to access HA-proxy server

```

) to the list of known hosts.

#_
~\##### Amazon Linux 2023
~\#####\
~\###|
~\#/ https://aws.amazon.com/linux/amazon-linux-2023
~V~'~>
~
~.~.~
~/_/~/
~/_/~/

[ec2-user@ip-172-31-21-136 ~]$ sudo -i
[root@ip-172-31-21-136 ~]# yum update
Amazon Linux 2023 Kernel Livepatch repository 206 kB/s | 23 kB 00:00

Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-21-136 ~]#

```

step 3 – install haproxy command – `yum install haproxy -y`

Step 4 – add server1 public and server2 to haproxy command –

Vi /etc/hosts

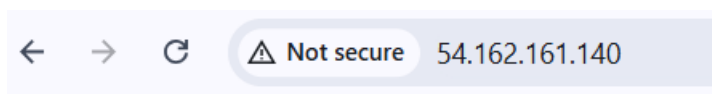
```
root@ip-172-31-21-136 ~]# vi /etc/hosts
root@ip-172-31-21-136 ~]# cat /etc/hosts
27.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
:1         localhost6 localhost6.localdomain6
4.228.77.160 server1
4.205.3.105 server2
root@ip-172-31-21-136 ~]# |
```

Step 5 – add server1 publicip with port80 and add server2 public ip with port 80

(34.228.77.160:80) – server 1

(54.205.3.105:80) – server 2

Step 6 – now browse with ha\_proxy public ip with port 80 , it will distribute to server1 , server 2



## It works!