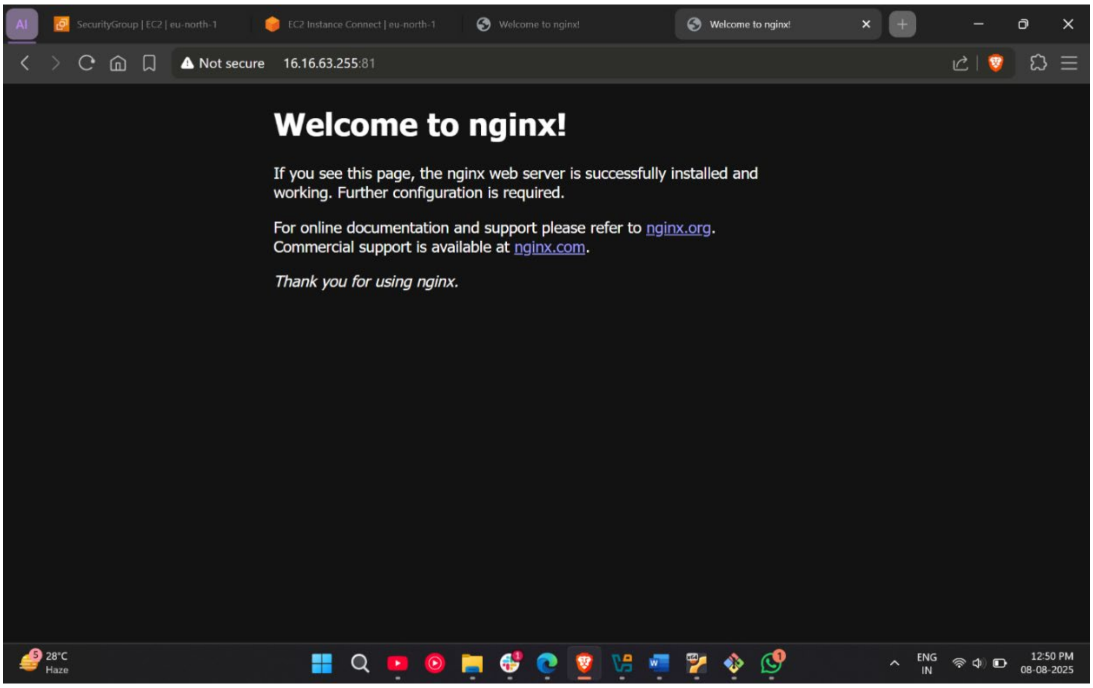
Web Server and Application Server

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



Welcome to nginx!

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

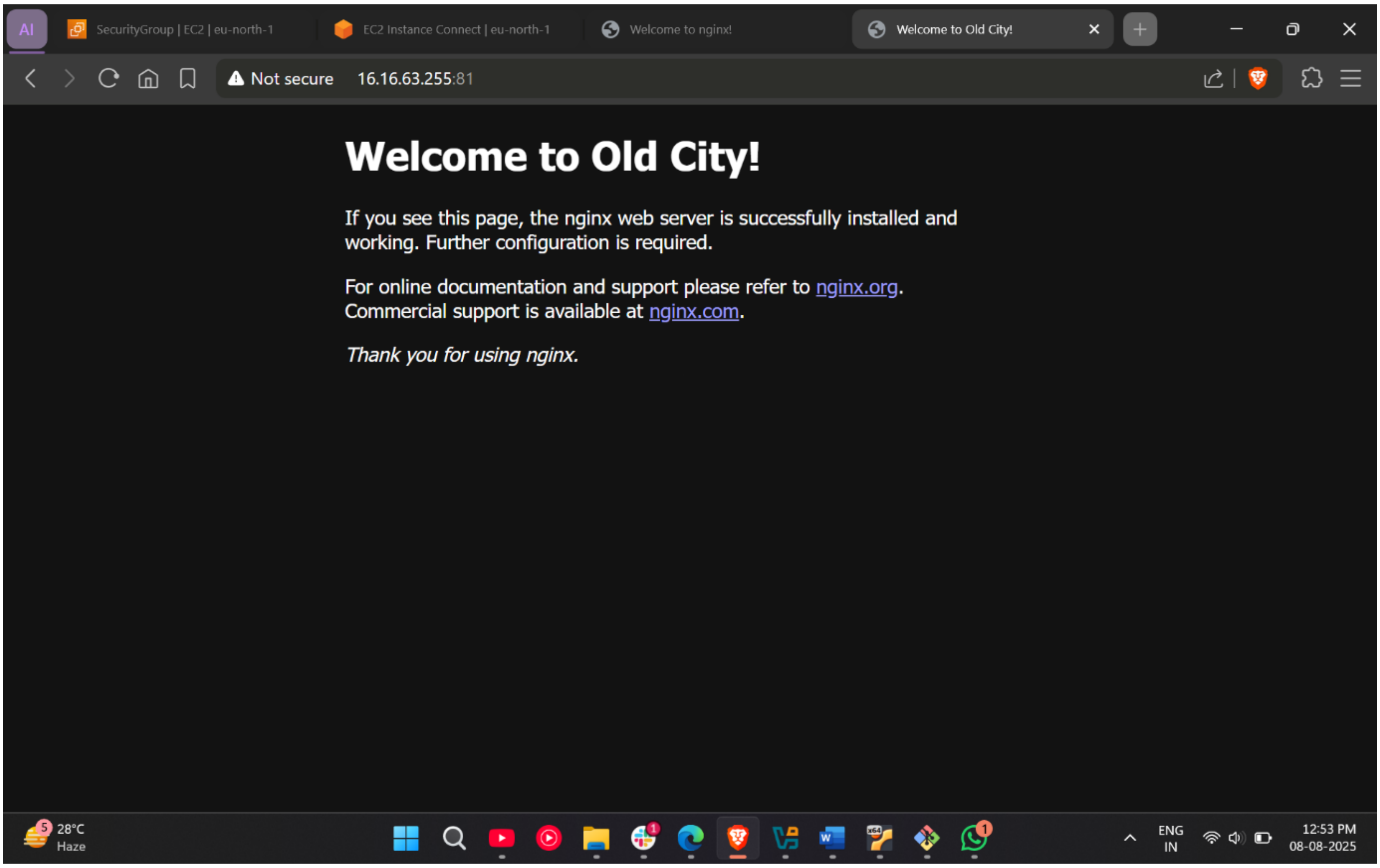
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Steps to install the nginx web server:

1) Launch an instance on AWS and select any Amazon Linux OS
2) Update and upgrade the machine before installing nginx. To do that, use:
   ```
   #  yum update & # yum upgrade
   ```
3) Install nginx using this command:
   ```
   #  yum install nginx -y
   ```
4) Next step is to start nginx and enable it. Do that using:
   ```
   #  systemctl start nginx
   #  systemctl enable nginx
   ```
5) The next step is to make a copy of the conf file. The conf file is located at /etc/nginx/nginx.conf. It is always best practice to make a backup of the file in case the original file gets corrupted.
   ```
   # cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf.backup
   ```
6) Edit the “nginx.conf” file and change the port number to 81 from 80.
   ```
   # vi nginx.conf or vim nginx.conf or nano nginx.conf
   ```
   While editing, we look for the port we need to change. It will be in the line:
   ```
   server {
           listen       80;
           listen       [::]:80;
           server_name  _;
   ```
7) Change the port to 81. Then restart nginx with:
   ```
   #  systemctl stop nginx or systemctl restart nginx
   ```
8) Check if the configuration is done right with:
   ```
   #  nginx -t
   ```
9) Copy the public IP from the machine and open a new tab in the browser. Paste the IP with the port number and press Enter. The nginx server will successfully run on port 81. Eg: 16.16.63.255:81

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



SecurityGroup | EC2 | eu-north-1
EC2 Instance Connect | eu-north-1
Welcome to nginx!
Welcome to Old City!
Not secure
16.16.63.255:81

Welcome to Old City!

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

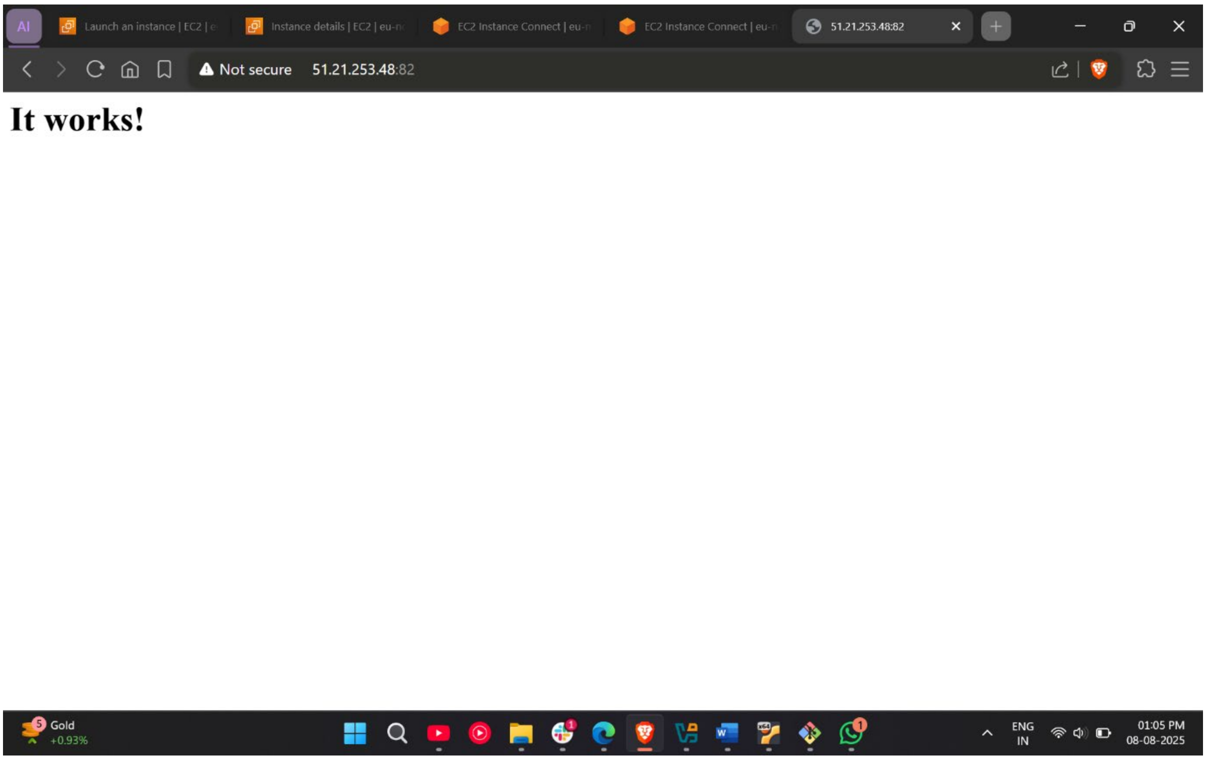
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Steps to deploy a sample HTML file on nginx:

1) Navigate to "/usr/share/nginx/html" directory and look for an index.html file.
2) Create a backup of the index file for safety
   # cp index.html index.html.backup
3) Open the index.html file and make the desired changes.
   # vi index.html
4) Look for the head and the body where you need to make changes, which appear in the output. Changes made in the head can be viewed on the browser's tab title, and changes made in the body can be noticed on the main heading of the page.
5) Once changes are made, save and exit the HTML file.
6) Check the changes by opening a new tab and pasting the public IP on port 81.
   Eg: 51.21.243.49:81

3) Install Apache HTTPD and run it on port number 82

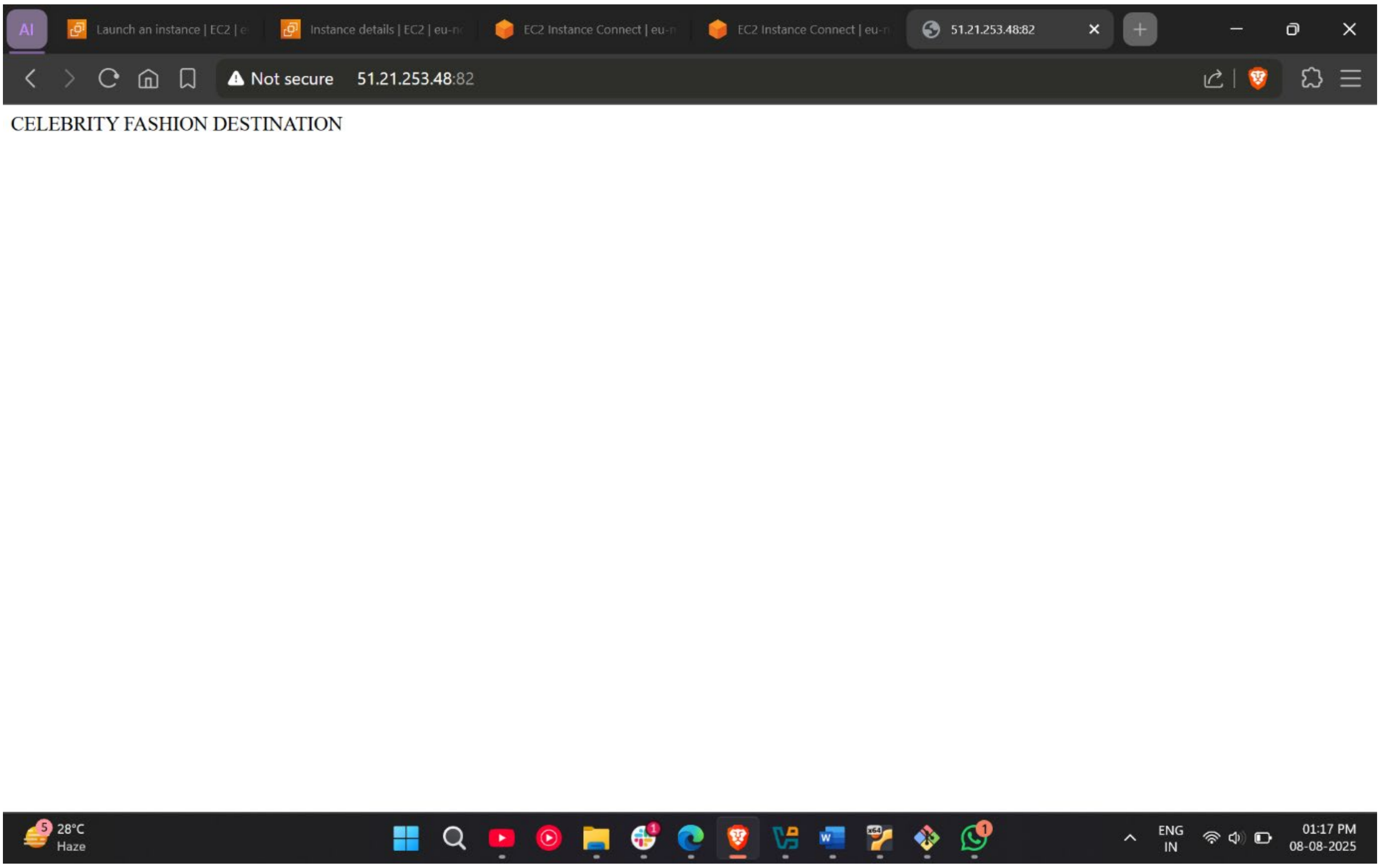

Launch an instance | EC2 | e
Instance details | EC2 | eu-n
EC2 Instance Connect | eu-n
EC2 Instance Connect | eu-n
51.21.253.48:82
Not secure
51.21.253.48:82
Gold
+0.93%
ENG
IN
01:05 PM
08-08-2025

It works!

Steps to install Apache HTTPD and run it:

1) Update and upgrade the machine.
   # yum update
   # yum upgrade
2) Install HTTPD.
   # yum install httpd -y
   (Use -y to say yes before installation, so you don't have to do it later)
3) Start and enable HTTPD
   # systemctl start httpd
   # systemctl enable httpd
4) Navigate to - /etc/httpd/conf/ and look for httpd.conf file. Make a copy for backup in case the original file gets corrupted.
   # cd /etc/httpd/conf
   # cp httpd.conf httpd.conf.backup
5) Open the file and change the port from 80 to 82. Save the file and provide the ownership to the httpd user.
   # vi httpd.conf – Look for port 80 and change it to 82
   (To easily find the keyword, type: esc -> :/+keyword to search. Eg: esc -> :/80)
   # chown -R httpd:httpd httpd.conf
6) Restart HTTPD and start again. Copy the public IP and open it on port 82.
   # systemctl restart httpd (or) #systemctl stop httpd
   # systemctl start httpd
7) Check for the results.

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

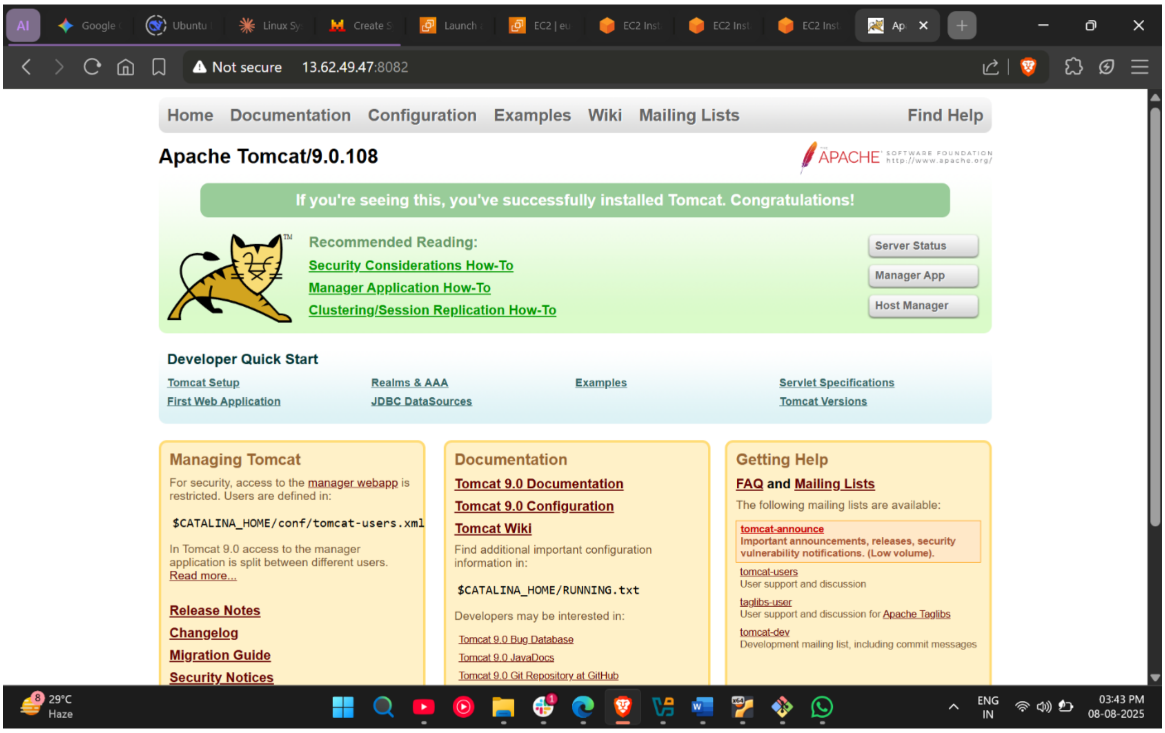


CELEBRITY FASHION DESTINATION

Steps to edit index.html file:

1) Find the index.html file using either the find command or navigate to /usr/share/httpd/noindex directory.
2) Give appropriate permissions and ownership to the /usr/share/httpd directory to the apache user.
   # chown -R apache: /usr/share/httpd/noindex or # chown -R apache: /var/www/html/index.html
3) In case of /var/www/html, create a file with the name index.html and add any text to check the output.
4) Open the index.html file with any editor.
   # vi index.html
5) Make the desired changes to the file and save it.
6) Restart the https service. If stop command is use, make sure to start the httpd service again.
   # systemctl restart httpd or # systemctl stop httpd
   # systemctl start httpd
7) Open a browser and enter the public IP with the port number 82.
8) Check the deployed index.html file.

5) Install Apache Tomcat on port number 8082



Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/9.0.108

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To

Server Status
Manager App
Host Manager

Developer Quick Start

Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 9.0 access to the manager application is split between different users. Read more...

Release Notes
Changelog
Migration Guide
Security Notices

Documentation

Tomcat 9.0 Documentation
Tomcat 9.0 Configuration
Tomcat Wiki

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

Tomcat 9.0 Bug Database
Tomcat 9.0 JavaDocs
Tomcat 9.0 Git Repository at GitHub

Getting Help

FAQ and Mailing Lists

The following mailing lists are available:

tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).

tomcat-users
User support and discussion

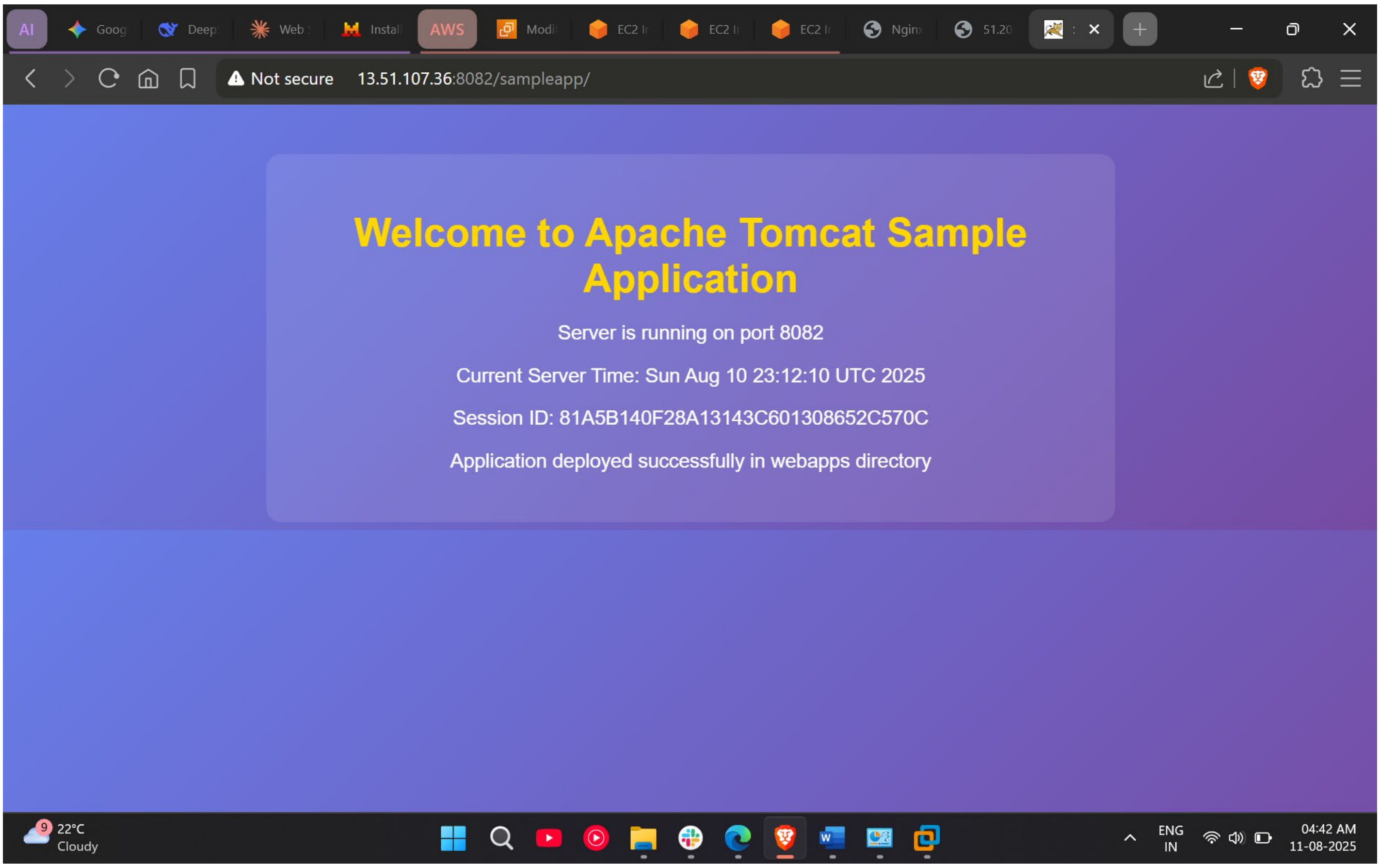
taglibs-user
User support and discussion for Apache Taglibs

tomcat-dev
Development mailing list, including commit messages

Steps to install and run Apache Tomcat:

1) Create an instance and name it tomcat. Select All traffic and use source as: 0.0.0.0/0.
2) Update and upgrade the system.
3) Install the appropriate version of Java into the system.
   # dnf install openjdk-21-amazon-corretto-devel -y
4) Add a user with the name tomcat and make it a system user with a home directory in /opt/tomcat and give it a /bin/false shell. This is a good practice that helps protect the server even after a cyberattack.
   # useradd -r -m -U -d /opt/tomcat -s /bin/false tomcat
   r – system user
   m – to create a home directory
   U – to create a group with the same username
   d – path/to/directory
   s – providing shell, which in this case is /bin/false
5) Download Tomcat from the internet into the "/tmp" directory. Click on the tar file and copy the link. To download, use:
   # wget <link_address>
6) Extract the file in the "/opt/tomcat" directory.
   # tar xvf <file_name> -C /opt/tomcat –strip-components=1. To complete the name of the file or directory, press the Tab key.
7) Change the ownership of the directory "/opt/tomcat", and edit the "server.xml" file. Change the port number from the default 8080 to 8082. Make a backup before editing.
   # chown -R tomcat: /opt/tomcat
   # cp /opt/tomcat/conf/server.xml /opt/tomcat/conf/server.xml.backup
   # vi server.xml -> Change port number from 8080 to 8082 wherever needed.
8) Restart Tomcat and start again from the bin directory.
   # ./shutdown.sh && # ./startup.sh
9) Open a new tab in the browser and check the output with the public IP on port 8082

6) Deploy a sample app on webapps



Welcome to Apache Tomcat Sample Application

Server is running on port 8082

Current Server Time: Sun Aug 10 23:12:10 UTC 2025

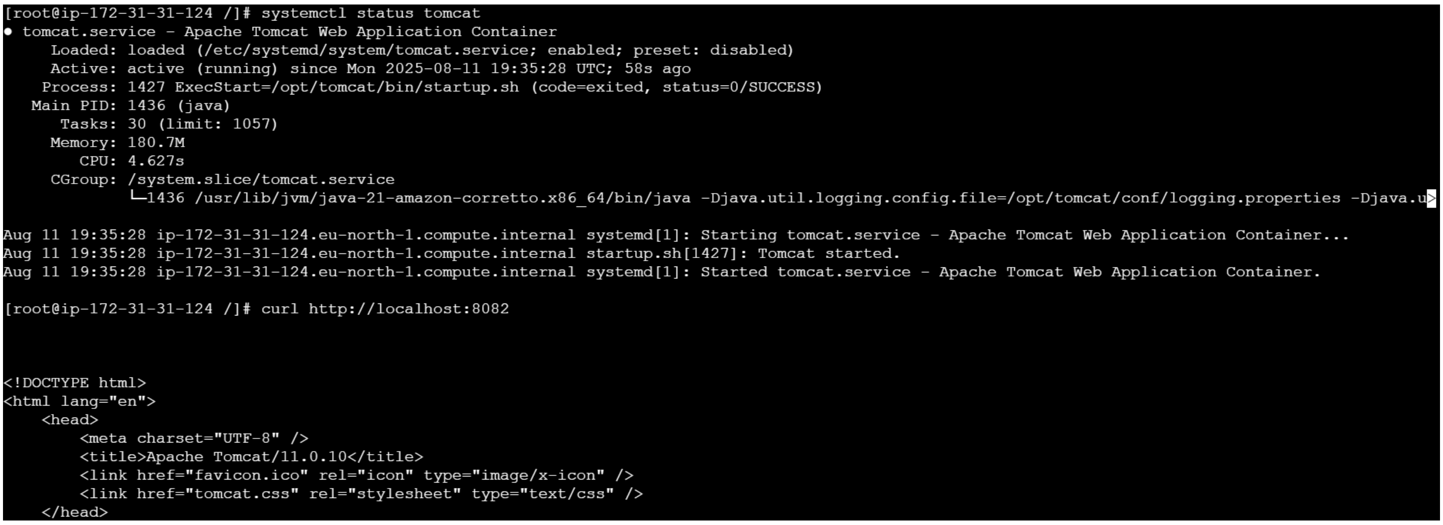
Session ID: 81A5B140F28A13143C601308652C570C

Application deployed successfully in webapps directory

Steps:

1) Create a new directory with the name “sampleapp” in the “/opt/tomcat/webapps” directory and add two files with the names “web.xml” where we will define how the web app will look like, and another file “index.jsp” where we add the content which appears of the web app.
   # mkdir /opt/tomcat/webapps/sampleapp
   # vi web.xml -> Define the web app
   # vi index.jsp -> Content of the web app
2) Give the ownership of the directory to tomcat
   # chown -R tomcat: /opt/tomcat/webapps/sampleapp
3) Restart Tomcat to deploy the application. Go to the “/opt/tomcat/conf” directory and use the commands:
   # ./shutdown.sh (or) # sh shutdown.sh
   # ./startup.sh (or) sh startup.sh
4) Check if the deployment is working with curl command
   # curl http://localhost:8082/sampleapp/
5) Open a new tab and copy the public IP from the instance and open it on port “8082/sampleapp/” to check the deployment.
   http:// 13.51.107.36:8082/sampleapp

7) Create a “tomcat.service” file for Tomcat.



```
[root@ip-172-31-31-124 /]# systemctl status tomcat
● tomcat.service - Apache Tomcat Web Application Container
     Loaded: loaded (/etc/systemd/system/tomcat.service; enabled; preset: disabled)
     Active: active (running) since Mon 2025-08-11 19:35:28 UTC; 58s ago
    Process: 1427 ExecStart=/opt/tomcat/bin/startup.sh (code=exited, status=0/SUCCESS)
   Main PID: 1436 (java)
      Tasks: 30 (limit: 1057)
     Memory: 180.7M
        CPU: 4.627s
     CGroup: /system.slice/tomcat.service
             └─1436 /usr/lib/jvm/java-21-amazon-corretto.x86_64/bin/java -Djava.util.logging.config.file=/opt/tomcat/conf/logging.properties -Djava.u>

Aug 11 19:35:28 ip-172-31-31-124.eu-north-1.compute.internal systemd[1]: Starting tomcat.service - Apache Tomcat Web Application Container...
Aug 11 19:35:28 ip-172-31-31-124.eu-north-1.compute.internal startup.sh[1427]: Tomcat started.
Aug 11 19:35:28 ip-172-31-31-124.eu-north-1.compute.internal systemd[1]: Started tomcat.service - Apache Tomcat Web Application Container.

[root@ip-172-31-31-124 /]# curl http://localhost:8082



<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <title>Apache Tomcat/11.0.10</title>
        <link href="favicon.ico" rel="icon" type="image/x-icon" />
        <link href="tomcat.css" rel="stylesheet" type="text/css" />
    </head>
```

Steps:

1) Create a “tomcat.service” file in the directory - /etc/systemd/system/tomcat.service.
2) Edit the file using vi command and enter the following content:

```
[Unit]

Description=Apache Tomcat Web Application Container

After=network.target



[Service]

Type=forking



Environment=JAVA_HOME=/usr/lib/jvm/”Check and enter the correct file/java version”

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

Environment=CATALINA_HOME=/opt/tomcat #make sure to check the directory name and enter as it is

Environment=CATALINA_BASE=/opt/tomcat #same as above

Environment='CATALINA_OPTS=-Xms256M -Xmx512M -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

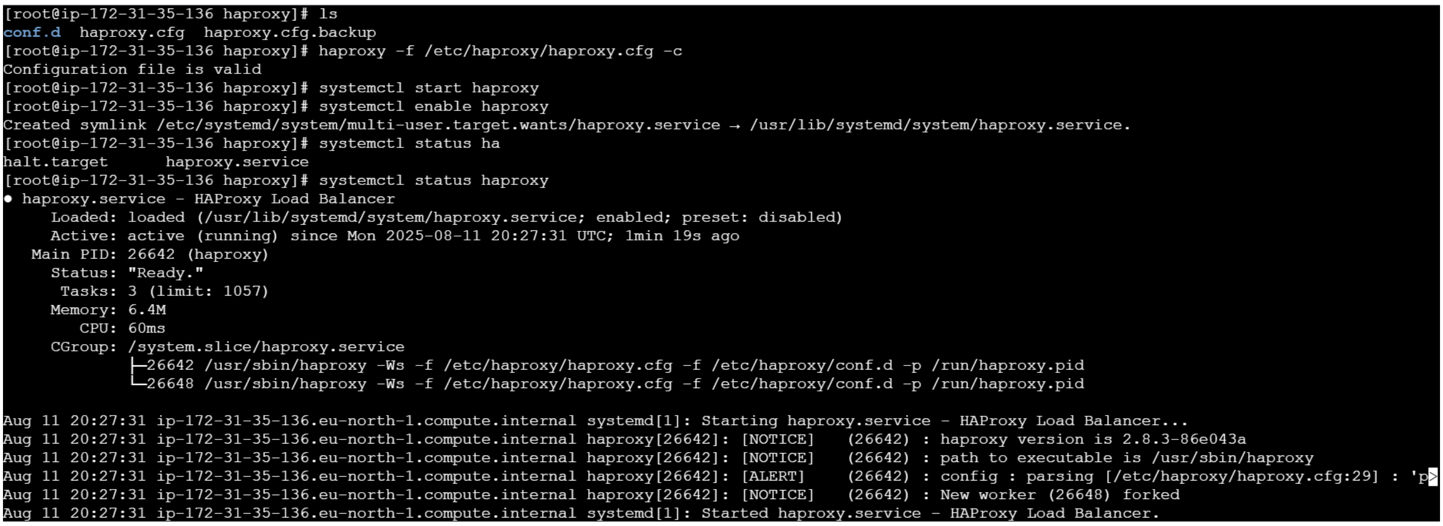
Restart=always

[Install]

WantedBy=multi-user.target

3) Save the file and go back to the directory "/opt/tomcat/bin" and manually shut down Tomcat.
4) Reload systemd and start the service. Use the following commands:
   # systemctl daemon-reload – Reload systemd configuration files
   # systemctl start tomcat – Start the Apache Tomcat service
   # systemctl enable tomcat – Enable automatic startup on boot
5) Verify if the service is running properly.
   # sudo systemctl status tomcat – Shows the status of the service
   # curl http://localhost:8082 – Shows whether the service is opening with the IP and Port number.

9) Configure HA Proxy server



```
[root@ip-172-31-35-136 haproxy]# ls
conf.d  haproxy.cfg  haproxy.cfg.backup
[root@ip-172-31-35-136 haproxy]# haproxy -f /etc/haproxy/haproxy.cfg -c
Configuration file is valid
[root@ip-172-31-35-136 haproxy]# systemctl start haproxy
[root@ip-172-31-35-136 haproxy]# systemctl enable haproxy
Created symlink /etc/systemd/system/multi-user.target.wants/haproxy.service → /usr/lib/systemd/system/haproxy.service.
[root@ip-172-31-35-136 haproxy]# systemctl status ha
halt.target      haproxy.service
[root@ip-172-31-35-136 haproxy]# systemctl status haproxy
● haproxy.service - HAProxy Load Balancer
     Loaded: loaded (/usr/lib/systemd/system/haproxy.service; enabled; preset: disabled)
     Active: active (running) since Mon 2025-08-11 20:27:31 UTC; 1min 19s ago
   Main PID: 26642 (haproxy)
     Status: "Ready."
      Tasks: 3 (limit: 1057)
     Memory: 6.4M
        CPU: 60ms
     CGroup: /system.slice/haproxy.service
             ├─26642 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -f /etc/haproxy/conf.d -p /run/haproxy.pid
             └─26648 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -f /etc/haproxy/conf.d -p /run/haproxy.pid

Aug 11 20:27:31 ip-172-31-35-136.eu-north-1.compute.internal systemd[1]: Starting haproxy.service - HAProxy Load Balancer...
Aug 11 20:27:31 ip-172-31-35-136.eu-north-1.compute.internal haproxy[26642]: [NOTICE]   (26642) : haproxy version is 2.8.3-86e043a
Aug 11 20:27:31 ip-172-31-35-136.eu-north-1.compute.internal haproxy[26642]: [NOTICE]   (26642) : path to executable is /usr/sbin/haproxy
Aug 11 20:27:31 ip-172-31-35-136.eu-north-1.compute.internal haproxy[26642]: [ALERT]    (26642) : config : parsing [/etc/haproxy/haproxy.cfg:29] : 'p>
Aug 11 20:27:31 ip-172-31-35-136.eu-north-1.compute.internal haproxy[26642]: [NOTICE]   (26642) : New worker (26648) forked
Aug 11 20:27:31 ip-172-31-35-136.eu-north-1.compute.internal systemd[1]: Started haproxy.service - HAProxy Load Balancer.
```

Steps to install and run HA-Proxy:

1) Create a new instance and install HA-Proxy
   # yum install haproxy -y
2) Make a copy of the configuration file and then edit it. The configuration file is located in the directory "/etc/haprocy/haprocy.cfg"
3) Add nginx, httpd, and tomcat IPs with port numbers
   # vi /etc/haproxy/haprocy.cfg – #Add the below text

```
global
   log        127.0.0.1:514 local0
   chroot     /var/lib/haproxy
   stats socket /run/haproxy/admin.sock mode 660 level admin
   stats timeout 30s
```

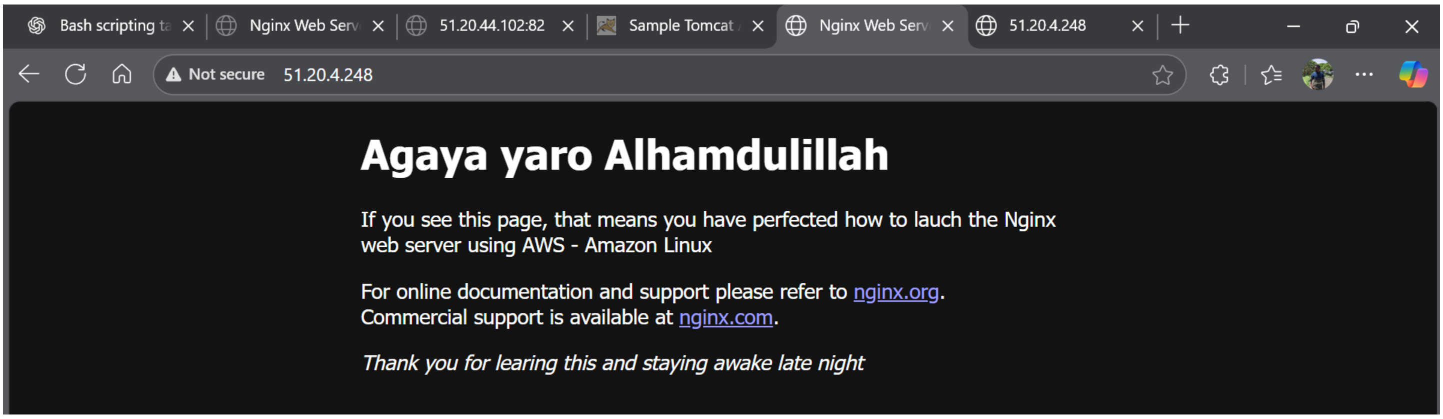
```
    user       haproxy
    group      haproxy
    daemon

defaults
    mode                   http
    log                    global
    option                 httplog
    option                 dontlognull
    option                 http-server-close
    option                 forwardfor       except 127.0.0.0/8
    option                 redispatch
    retries                3
    timeout http-request    10s
    timeout queue           1m
    timeout connect         10s
    timeout client          1m
    timeout server          1m
    timeout http-keep-alive 10s
    timeout check           10s
    maxconn                3000

# HAProxy Statistics
frontend stats
    bind *:8404
    stats enable
    stats uri /stats
    stats refresh 30s
    stats admin if TRUE

# Frontend for web servers (Nginx and Apache)
frontend web_frontend
    bind *:80
    default_backend web_servers

# Backend for web servers
backend web_servers
    balance roundrobin
    option httpchk GET /
    server nginx_server 127.0.0.1:81 check
    server apache_server 127.0.0.1:82 check

# Frontend for Tomcat applications
frontend tomcat_frontend
    bind *:8080
    default_backend tomcat_servers
```

```
# Backend for Tomcat servers
backend tomcat_servers
   balance roundrobin
   option httpchk GET /
   server tomcat_server 127.0.0.1:8082 check
```

4) Test HA-Proxy configuration using:
   # haproxy -f /etc/haproxy/haproxy.cfg -c
5) Start and Enable HA-Proxy
   # sudo systemctl start haproxy
   # sudo systemctl enable haproxy
6) Verify the status of the service using:
   sudo systemctl status haproxy
7) On the “haproxy” server, open the file “hosts” which is located in the “/etc” directory and add the public IPs of “nginx”, “httpd”, and “tomcat”.
   # vi /etc/hosts – Add the following IPs below the already existing data

```
13.60.224.33   nginx
51.20.105.32   httpd
51.20.83.104   tomcat
```

8) On the “nginx” server and “httpd” server, open the “hosts” file and add the “haproxy” server’s public IP, as in the above step.
9) Open a two new tabs and copy paste the public IP of “haproxy” to check the result of load balancer.

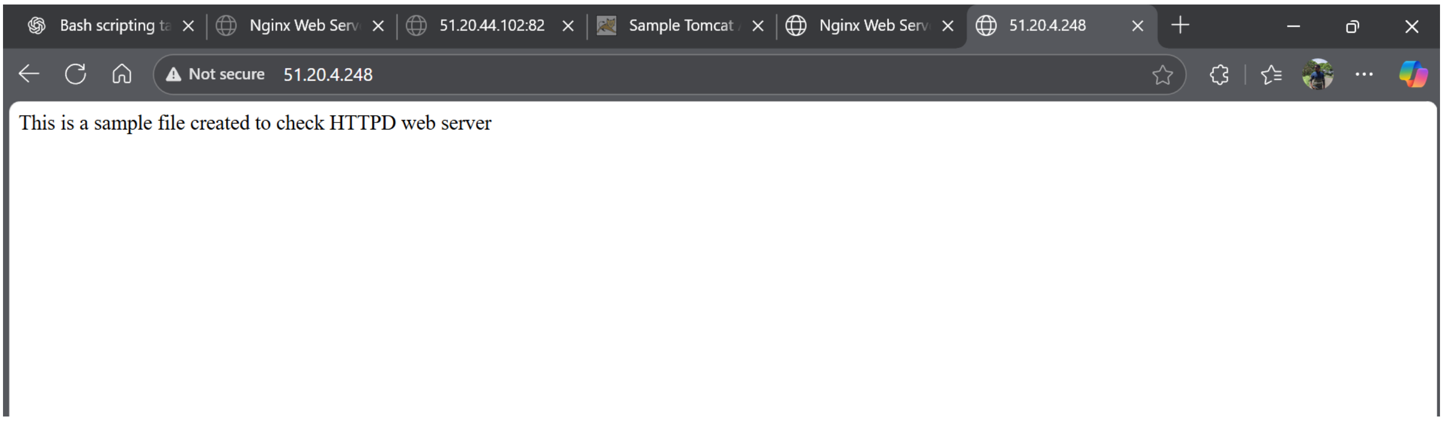


Agaya yaro Alhamdulillah

If you see this page, that means you have perfected how to lauch the Nginx web server using AWS - Amazon Linux

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for learing this and staying awake late night



This is a sample file created to check HTTPD web server

Shaik Omer Farooq
Batch 14