Java Web Greeter and Quote App

Overview:

A simple Java application that greets the user using a given name and also displays a random motivational quote. Also using Maven and Tomcat to deploy in two different servers one for building and another for deploying in AWS EC2.

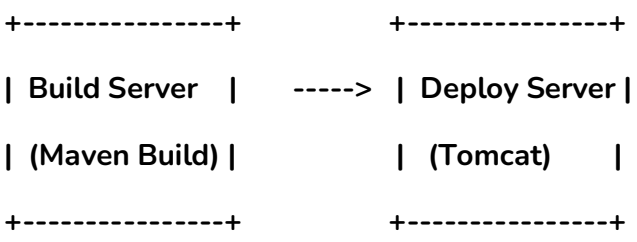
Tech Stack:

- **Language:** Java
- **Build Tool:** Maven
- **Server:** Apache Tomcat
- **Deployment:** Two servers – Build Server & Deploy Server

Prerequisites:

- **Java JDK 17** on both servers
- **Maven** installed on **Build Server**
- **Tomcat** installed on **Deploy Server**
- Both servers **must be in the same network**, typically the same **VPC** in AWS, to **copy files internally using private IPs**

Architecture:



Step-by-Step Guide: Two-Server Java Web Greeter and Quote App Deployment

1. Create the Servers:

Build Server

1. **Launch a Ubuntu EC2.**
    - **Amazon Machine Image: Ubuntu 24.04 LTS**
    - **Instance type: t3 micro(2 vCPU, RAM: 1 GB)**
2. **Create a Key pair(Build-Key.pem) for Secure Login.**
3. **Create a Security group(SSH-Build) or select an existing group(SSH-Build).**
    - **Add SSH port(22) to the inbound rule (If creating a Security group).**

Deploy Server

1. **Launch a Ubuntu EC2.**
    - **Amazon Machine Image: Ubuntu 24.04 LTS**
    - **Instance type: t3 micro(2 vCPU, RAM: 1 GB)**
2. **Create a Key pair(Deploy-Key.pem) for Secure Login.**
3. **Create a Security group(Tomcat-Deploy) or select an existing group(Tomcat-Deploy).**
    - **Add SSH port(22) to the inbound rule (If creating a Security group).**
    - **Add Custom port(8080) to the inbound rule(Tomcat runs on this Port).**

Tip: Make sure both servers can communicate (or at least Build Server can SCP files to Deploy Server).

2. Install Prerequisites:

a. On Both Servers:

sudo apt -y update (To Update the Server)

git (to check whether the git is installed or not)

java –version (Shows the Java Versions to Download)

sudo apt install openjdk-17-jre-headless (To install the Java 17 version)

Build-Server:

```
ubuntu@ip-172-31-24-227:~$ sudo apt -y update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
```

```
ubuntu@ip-172-31-24-227:~$ git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--config-env=<name>=<envvar>] <command> [<args>]
```

```
ubuntu@ip-172-31-24-227:~$ java --version
Command 'java' not found, but can be installed with:
sudo apt install openjdk-17-jre-headless  # version 17.0.16+8~us1-0ubuntu1~24.04.1, or
sudo apt install openjdk-21-jre-headless  # version 21.0.8+9~us1-0ubuntu1~24.04.1
sudo apt install default-jre              # version 2:1.17-75
sudo apt install openjdk-11-jre-headless  # version 11.0.28+6-1ubuntu1~24.04.1
sudo apt install openjdk-8-jre-headless   # version 8u462-ga~us1-0ubuntu2~24.04.2
sudo apt install openjdk-19-jre-headless  # version 19.0.2+7-4
sudo apt install openjdk-20-jre-headless  # version 20.0.2+9-1
sudo apt install openjdk-22-jre-headless  # version 22~22ea-1
```

```
ubuntu@ip-172-31-24-227:~$ sudo apt install openjdk-17-jre-headless
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf ca-certificates-java fontconfig-config fonts-dejavu-core fonts
  java-common libasound2-data libasound2t64 libavahi-client3 libavahi-common-data libavahi-commor
  libfontconfig1 libgraphite2-3 libharfbuzz0b libjpeg-turbo8 libjpeg8 liblcms2-2 libpcsclite1
Suggested packages:
  default-jre alsa-utils libasound2-plugins cups-common liblcms2-utils pcscd libnss-mdns fonts-de
```

Deploy-Server:

```
ubuntu@ip-172-31-19-39:~$ sudo apt -y update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1213 kB]
```

```
ubuntu@ip-172-31-19-39:~$ sudo apt install openjdk-17-jre-headless
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf ca-certificates-java fontconfig-config fonts-dejavu-core fonts-dejavu-mono
  java-common libasound2-data libasound2t64 libavahi-client3 libavahi-common-data libavahi-common3 libcups2t64
  libfontconfig1 libgraphite2-3 libharfbuzz0b libjpeg-turbo8 libjpeg8 liblcms2-2 libpcsclite1
```

b. Build Server:

- Install **Maven**:
    - Sudo apt -y install **maven**

```
ubuntu@ip-172-31-24-227:~$ sudo apt install -y maven
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libaopalliance-java libapache-pom-java libatinject-jsr330-api-java libcdi-api-java libcommon
  libcommons-io-java libcommons-lang3-java libcommons-parent-java liberror-prone-java
```

c. Deploy Server:

- Install **Tomcat**:
    - Using **wget** command and URL from **Tomcat** tar.gz.

```
ubuntu@ip-172-31-19-39:~$ wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.110/bin/apache-tomcat-9.0.110.tar.gz
--2025-10-08 05:03:20--  https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.110/bin/apache-tomcat-9.0.110.tar.gz
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... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13036068 (12M) [application/x-gzip]
Saving to: 'apache-tomcat-9.0.110.tar.gz'

apache-tomcat-9.0.110.tar.gz  100%[=================================================>]  12.43M  --.-KB/s    in 0.04s

2025-10-08 05:03:20 (333 MB/s) - 'apache-tomcat-9.0.110.tar.gz' saved [13036068/13036068]
```

3. Setup Project on Build Server:

A. Clone the repository:

a. Using the command git clone URL of the repository.

b. Change directory to JavaWebGreeterQuoteApp.

```
ubuntu@ip-172-31-24-227:~$ git clone https://github.com/Charantej-afk/JavaWebGreeterQuoteApp.git
Cloning into 'JavaWebGreeterQuoteApp'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 22 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (22/22), 5.37 KiB | 1.79 MiB/s, done.
```

```
ubuntu@ip-172-31-24-227:~$ ls
JavaWebGreeterQuoteApp
ubuntu@ip-172-31-24-227:~$ cd JavaWebGreeterQuoteApp/
ubuntu@ip-172-31-24-227:~/JavaWebGreeterQuoteApp$ ls
pom.xml  src
```

B. Validate and Build with Maven:

a. Validate with using the command - “mvn validate”

```
ubuntu@ip-172-31-24-227:~/JavaWebGreeterQuoteApp$ mvn validate
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------< mypackage:JavaWebGreeterQuoteApp >------------------
[INFO] Building Java Web Greeter Quote App 1.0
[INFO] --------------------------------[ war ]---------------------------------
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.098 s
[INFO] Finished at: 2025-10-08T05:01:40Z
[INFO] ------------------------------------------------------------------------
```

b. Build with using the command - “mvn package”

i. Use the command to compile and build the java webapp at once.

```
ubuntu@ip-172-31-24-227:~/JavaWebGreeterQuoteApp$ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------< mypackage:JavaWebGreeterQuoteApp >------------------
[INFO] Building Java Web Greeter Quote App 1.0
[INFO] --------------------------------[ war ]---------------------------------
```

ii. This is a situation in which we will get errors during packaging because the pom.xml file is build for the older version of Java and Maven. So we need to modify the pom.xml file to the correct version of Java and Maven. Normally this is done by the Code Developers.

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  14.289 s
[INFO] Finished at: 2025-10-07T08:45:53Z
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-war-plugin:2.2:war (default-war) on project webapp: Execut
ion default-war of goal org.apache.maven.plugins:maven-war-plugin:2.2:war failed: Unable to load the mojo 'war' in the p
lugin 'org.apache.maven.plugins:maven-war-plugin:2.2' due to an API incompatibility: org.codehaus.plexus.component.repos
itory.exception.ComponentLookupException: Cannot access defaults field of Properties
[ERROR] -----------------------------------------------------
[ERROR] realm =    plugin>org.apache.maven.plugins:maven-war-plugin:2.2
[ERROR] strategy = org.codehaus.plexus.classworlds.strategy.SelfFirstStrategy
[ERROR] urls[0] = file:/home/ubuntu/.m2/repository/org/apache/maven/plugins/maven-war-plugin/2.2/maven-war-plugin-2.2.ja
```

iii. But here we will get no errors because the pom.xml is in line with the newer version or whatever the Java version we are using for building and deploying the Java webapp.

```
[INFO] Packaging webapp
[INFO] Assembling webapp [JavaWebGreeterQuoteApp] in [/home/ubuntu/JavaWebGreeterQuoteApp/target/JavaWebGreeterQuoteApp]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/ubuntu/JavaWebGreeterQuoteApp/src/main/webapp]
[INFO] Building war: /home/ubuntu/JavaWebGreeterQuoteApp/target/JavaWebGreeterQuoteApp.war
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  17.077 s
[INFO] Finished at: 2025-10-08T05:02:08Z
[INFO] ------------------------------------------------------------------------
```

This is a Situation where we get an error while packaging the following Steps need to be run by the "Code Developer" to edit so that the pom.xml file is in line with the Newer version of the Java and Maven.

c. **If there are errors in the pom.xml file then edit & upload it to the Build server.**

d. **To read the pom.xml file we use the command:**

   i. Cat pom.xml

```
ubuntu@ip-172-31-27-77:~/JavaWebCalculator$ cat pom.xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
```

e. **Clear and Edit the pom.xml file using the command:**

   i. > pom.xml

   ii. vi pom.xml (vi is the text editor).

```
ubuntu@ip-172-31-27-77:~/JavaWebCalculator$ >pom.xml
ubuntu@ip-172-31-27-77:~/JavaWebCalculator$ vi pom.xml
ubuntu@ip-172-31-27-77:~/JavaWebCalculator$
```

f. **Clean maven package:**

   i. mvn clean

```
ubuntu@ip-172-31-27-77:~/JavaWebCalculator$ mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< com.web.cal:webapp >------------------------
[INFO] Building WebAppCal Maven Webapp 0.1.3
[INFO] --------------------------------[ war ]---------------------------------
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.pom (3.9 kB at 9.7 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.jar (25 kB at 432 kB/s)
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ webapp ---
[INFO] Deleting /home/ubuntu/JavaWebCalculator/target
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.033 s
[INFO] Finished at: 2025-10-07T09:00:55Z
[INFO] ------------------------------------------------------------------------
```

---

g. Build the Webapp:

i. mvn package

```
ubuntu@ip-172-31-27-77:~/JavaWebCalculator$ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< com.web.cal:webapp >------------------------
[INFO] Building WebAppCal Maven Webapp 0.1.3
[INFO] --------------------------------[ war ]---------------------------------
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-resources-plugin/2.6/maven
[INFO] Packaging webapp
[INFO] Assembling webapp [webapp] in [/home/ubuntu/JavaWebCalculator/target/webapp]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/ubuntu/JavaWebCalculator/src/main/webapp]
[INFO] Building war: /home/ubuntu/JavaWebCalculator/target/webapp.war
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  10.040 s
[INFO] Finished at: 2025-10-07T09:06:36Z
[INFO] ------------------------------------------------------------------------
```

C. Result:

a. After Successful Building of webapp you will get a folder target in it you will have the Artifact file or WAR file.

```
ubuntu@ip-172-31-24-227:~/JavaWebGreeterQuoteApp$ ls
pom.xml  src  target
ubuntu@ip-172-31-24-227:~/JavaWebGreeterQuoteApp$ cd target/
ubuntu@ip-172-31-24-227:~/JavaWebGreeterQuoteApp/target$ ls
JavaWebGreeterQuoteApp  JavaWebGreeterQuoteApp.war  classes  generated-sources  maven-archiver  maven-status
```

4.Setup project on Deploy server:

A. Tomcat Zipped Tar file:

```
ubuntu@ip-172-31-19-39:~$ ls
apache-tomcat-9.0.110.tar.gz
```

B. **Extract the tar file content:**

```
ubuntu@ip-172-31-19-39:~$ tar -xvf apache-tomcat-9.0.110.tar.gz
apache-tomcat-9.0.110/conf/
apache-tomcat-9.0.110/conf/catalina.policy
apache-tomcat-9.0.110/conf/catalina.properties
apache-tomcat-9.0.110/conf/context.xml
apache-tomcat-9.0.110/conf/jaspic-providers.xml
apache-tomcat-9.0.110/conf/jaspic-providers.xsd
apache-tomcat-9.0.110/conf/logging.properties
apache-tomcat-9.0.110/conf/server.xml
apache-tomcat-9.0.110/conf/tomcat-users.xml
apache-tomcat-9.0.110/conf/tomcat-users.xsd
```

```
ubuntu@ip-172-31-19-39:~$ ls
apache-tomcat-9.0.110  apache-tomcat-9.0.110.tar.gz
```

C. **Change Directory to tomcat file:**

```
ubuntu@ip-172-31-19-39:~$ cd apache-tomcat-9.0.110/
ubuntu@ip-172-31-19-39:~/apache-tomcat-9.0.110$ ls
BUILDING.txt     LICENSE  README.md      RUNNING.txt  conf  logs  webapps
CONTRIBUTING.md  NOTICE   RELEASE-NOTES  bin          lib   temp  work
```

D. **Start Tomcat:**

a. Go to bin directory using cd bin/

```
ubuntu@ip-172-31-19-39:~/apache-tomcat-9.0.110$ cd bin/
ubuntu@ip-172-31-19-39:~/apache-tomcat-9.0.110/bin$ ls
bootstrap.jar       ciphers.sh                  daemon.sh     setclasspath.bat  startup.sh           version.bat
catalina-tasks.xml  commons-daemon-native.tar.gz  digest.bat    setclasspath.sh   tomcat-juli.jar      version.sh
catalina.bat        commons-daemon.jar          digest.sh     shutdown.bat      tomcat-native.tar.gz
catalina.sh         configtest.bat              makebase.bat  shutdown.sh       tool-wrapper.bat
ciphers.bat         configtest.sh               makebase.sh   startup.bat       tool-wrapper.sh
```

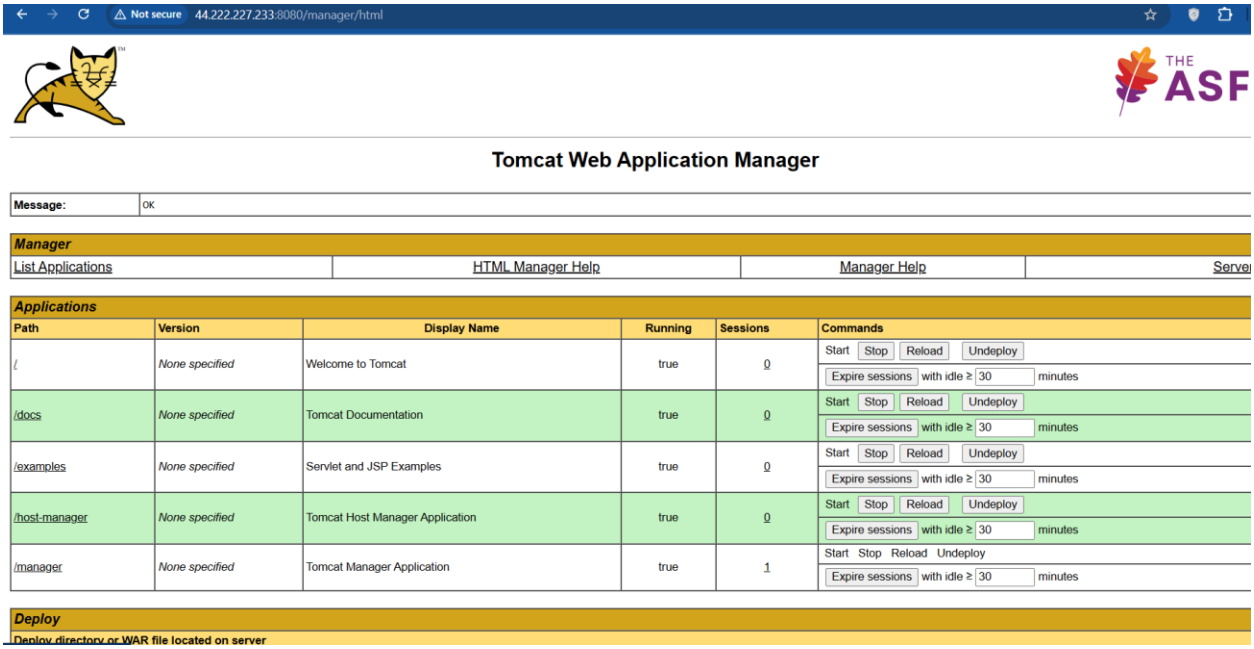
b. Run the Script startup.sh: using ./startup.sh

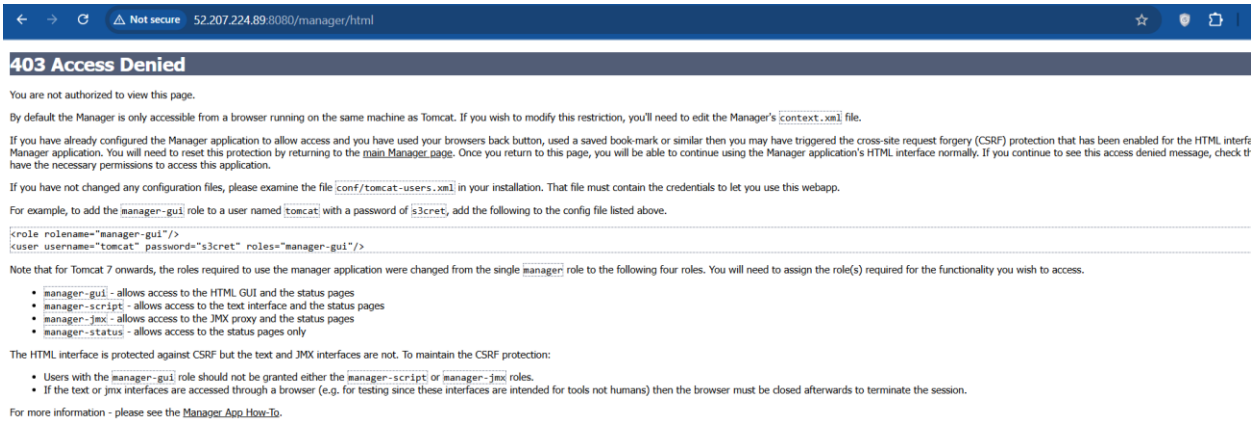
```
ubuntu@ip-172-31-19-39:~/apache-tomcat-9.0.110/bin$ ./startup.sh
Using CATALINA_BASE:   /home/ubuntu/apache-tomcat-9.0.110
Using CATALINA_HOME:   /home/ubuntu/apache-tomcat-9.0.110
Using CATALINA_TMPDIR: /home/ubuntu/apache-tomcat-9.0.110/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/ubuntu/apache-tomcat-9.0.110/bin/bootstrap.jar:/home/ubuntu/apache-tomcat-9.0.110/bin/tomca
t-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

c. Using the public ip of the Deploy server and port number of tomcat we can observe tomcat.

i. <PUBLIC-IP-OF-THE-DEPLOY-SERVER>:8080

ii. Access Manager App on the website. We will be Access Denied.





E. Edit Manager's context.xml files:

a. Search Manager's context.xml files using command

Find -name context.xml

```
ubuntu@ip-172-31-19-39:~/apache-tomcat-9.0.110$ find -name context.xml
./conf/context.xml
./webapps/manager/META-INF/context.xml
./webapps/host-manager/META-INF/context.xml
./webapps/docs/META-INF/context.xml
./webapps/examples/META-INF/context.xml
```

b. Edit the files that contain manager words or remove the highlighted lines.

vi ./webapps/manager/META_INF/context.xml
vi ./webapps/host-manager/META-INF/context.xml

```
ubuntu@ip-172-31-19-39:~/apache-tomcat-9.0.110$ vi ./webapps/manager/META-INF/context.xml
ubuntu@ip-172-31-19-39:~/apache-tomcat-9.0.110$ vi ./webapps/host-manager/META-INF/context.xml
```

- This is the content that will be in the both file with some difference but we need to change only one line that is in the next point.

```
<Context antiResourceLocking="false" privileged="true" >
  <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
                   sameSiteCookies="strict" />
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
         allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|String)|org\.apache\.catalin
a\.filters\.CsrfPreventionFilter\$LruCache(?:\$1)?|java\.util\.(?:Linked)?HashMap"/>
</Context>
~
```

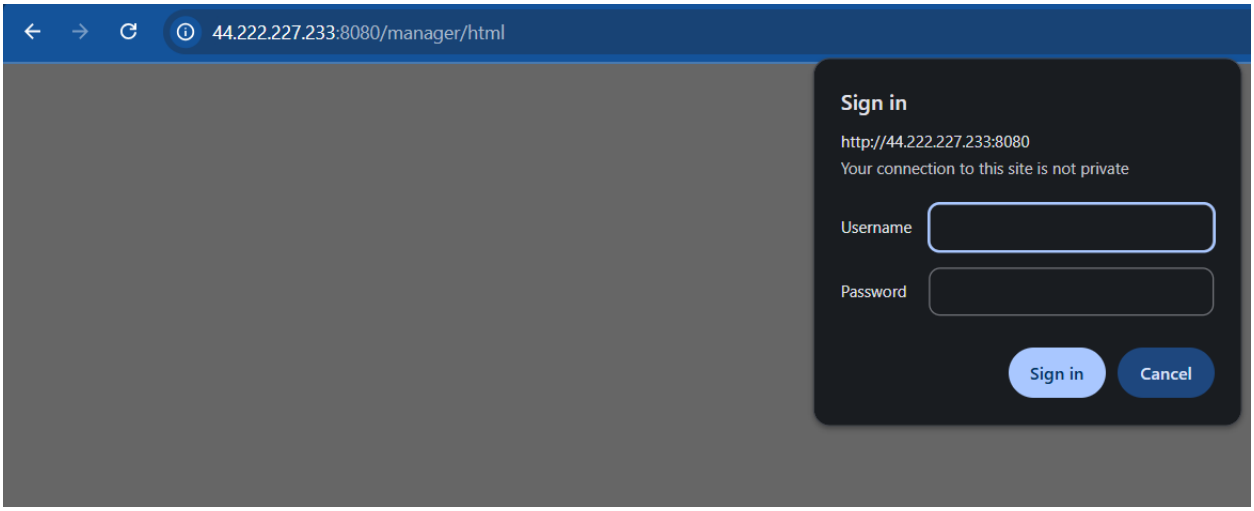
- This is the line that was commented so that when tomcat reads the file it will ignore this line.

```
<!--<Valve className="org.apache.catalina.valves.RemoteAddrValve"
allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->
```

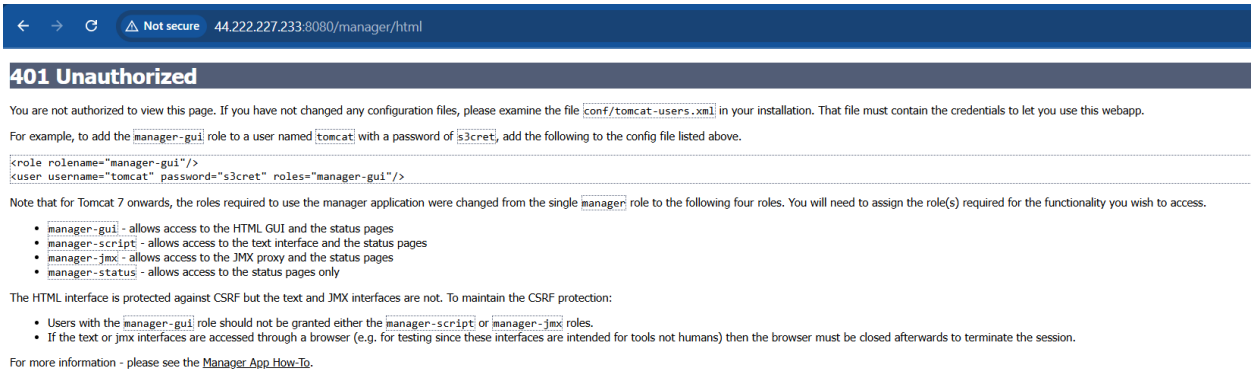
- Another option is to delete the line so that we have no need to comment on the line.

```
<Context antiResourceLocking="false" privileged="true" >
  <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
                   sameSiteCookies="strict" />
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|String)|org\.apache\.catalin
a\.filters\.CsrfPreventionFilter\$LruCache(?:\$1)?|java\.util\.(?:Linked)?HashMap"/>
</Context>
```

c. Now go to Tomcat Website again it'll ask for username and password which we didn't configure:



i. Select Cancel which takes us to this page:



d. To avoid this we need to configure a username and password in the conf directory in it we need to edit the tomcat-users.xml file

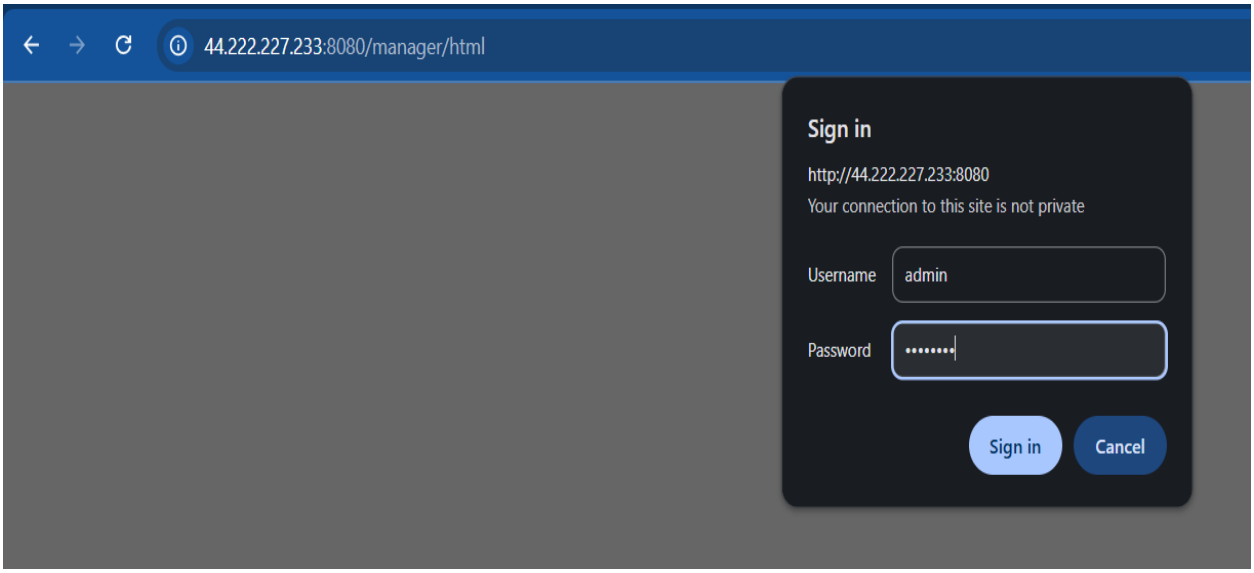
```
ubuntu@ip-172-31-19-39:~/apache-tomcat-9.0.110$ cd conf/
ubuntu@ip-172-31-19-39:~/apache-tomcat-9.0.110/conf$ ls
Catalina          catalina.properties  jaspic-providers.xml  logging.properties  tomcat-users.xml  web.xml
catalina.policy   context.xml          jaspic-providers.xsd  server.xml          tomcat-users.xsd
```

```
ubuntu@ip-172-31-19-39:~/apache-tomcat-9.0.110/conf$ vi tomcat-users.xml
```

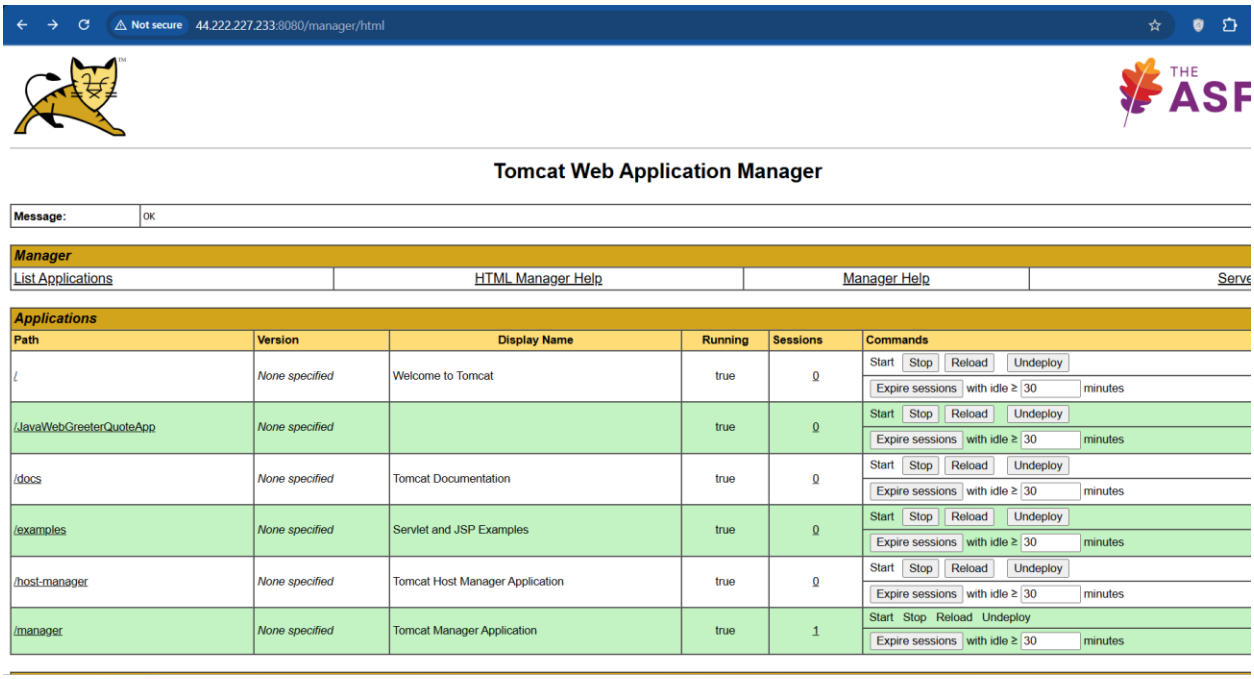
e. Here we need to add the user and password to this file.

```
-->
<!--
  <user username="admin" password="<must-be-changed>" roles="manager-gui"/>
  <user username="robot" password="<must-be-changed>" roles="manager-script"/>
-->
<!--
  The sample user and role entries below are intended for use with the
  examples web application. They are wrapped in a comment and thus are ignored
  when reading this file. If you wish to configure these users for use with the
  examples web application, do not forget to remove the <!.. ..> that surrounds
  them. You will also need to set the passwords to something appropriate.
-->
<!--
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
  <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
  <user username="role1" password="<must-be-changed>" roles="role1"/>
-->
<role rolename="manager-gui"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
</tomcat-users>
-- INSERT (paste) --
```

f. Now, reload the Webpage and Enter credentials we have given in tomcat webpage.



g. We will see this page where all the webapps in the webapp directory contained in the tomcat file.



5. Transfer WAR File to Deploy Server:

A. Copying pem file:

a. We need to make sure the Deploy server PEM file is on the Build server.

i. From your **local machine**, copy it to the build server and check in the Build server using:

```
scp -i "C:\Users\chara\Downloads\Build-Key.pem"
"C:\Users\chara\Downloads\Deploy-Key.pem"
ubuntu@<PUBLIC_IP_OF_BUILD_SERVER>:~/
```

```
PS C:\Users\chara> scp -i "C:\Users\chara\Downloads\Build-Key.pem" "C:\Users\chara\Downloads\Deploy-Key.pem" ubuntu@54.9
1.192.249:~/
The authenticity of host '54.91.192.249 (54.91.192.249)' can't be established.
ED25519 key fingerprint is SHA256:89hq+a1L7BrY5B/Y7LzQ0tybIPQRguMeBxKx4/sLlww.
This host key is known by the following other names/addresses:
    C:\Users\chara/.ssh/known_hosts:134: ec2-54-91-192-249.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added '54.91.192.249' (ED25519) to the list of known hosts.
Deploy-Key.pem                                                    100% 1678     6.1KB/s   00:00
```

```
ubuntu@ip-172-31-24-227:~$ ls
Deploy-Key.pem  JavaWebGreeterQuoteApp
```

B. Give proper Permissions:

a. After copying the pem file we need to give proper(read) permissions to the pem file.

```
ubuntu@ip-172-31-24-227:~$ chmod 400 ~/Deploy-Key.pem
ubuntu@ip-172-31-24-227:~$ ls -l
total 8
-r-------- 1 ubuntu ubuntu 1678 Oct  8 05:13 Deploy-Key.pem
drwxrwxr-x 5 ubuntu ubuntu 4096 Oct  8 05:01 JavaWebGreeterQuoteApp
```

C. Copy WAR file to Deploy server:

a. Inside the Build server we need to SCP(Secure Copy) the WAR file using:
scp -i ~/Deploy-Key.pem ~/JavaWebGreeterQuoteApp/target/*.war ubuntu@172.31.20.230:~/apache-tomcat-9.0.110/webapps/

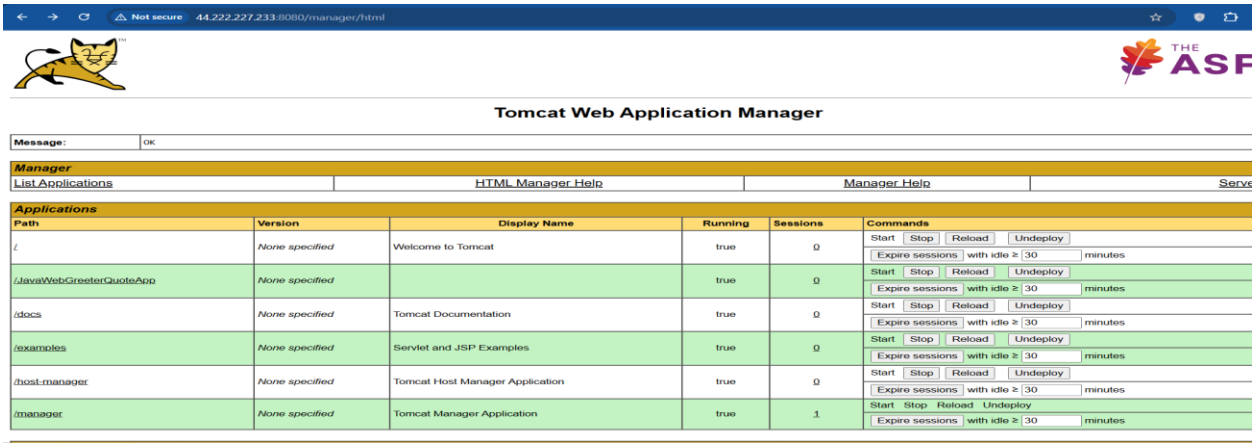
b. It will ask confirmation say “yes” it will send the copy to the Deploy server:

```
ubuntu@ip-172-31-24-227:~$ scp -i ~/Deploy-Key.pem /home/ubuntu/JavaWebGreeterQuoteApp/target/JavaWebGreeterQuoteApp.war
 ubuntu@172.31.19.39:~/apache-tomcat-9.0.110/webapps/
The authenticity of host '172.31.19.39 (172.31.19.39)' can't be established.
ED25519 key fingerprint is SHA256:MKtQSmaHTH3S4ZdJ37HnoNDV9qc5dLr7YQzKRZT+WNA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.19.39' (ED25519) to the list of known hosts.
JavaWebGreeterQuoteApp.war                                    100% 4337     6.1MB/s   00:00
```

c. Check in the Deploy server to find the WAR file in webapps directory.

```
ubuntu@ip-172-31-19-39:~/apache-tomcat-9.0.110$ cd webapps/
ubuntu@ip-172-31-19-39:~/apache-tomcat-9.0.110/webapps$ ls
JavaWebGreeterQuoteApp  JavaWebGreeterQuoteApp.war  ROOT  docs  examples  host-manager  manager
```

d. Now, check the webpage whether it is reflected in the Applications Table:



D. Restart the Tomcat or Deploy server:

a. Go to bin/ directory using:

i. cd ~/apache-tomcat-9.0.110/bin/

```
ubuntu@ip-172-31-19-39:~/apache-tomcat-9.0.110/webapps$ cd ..
ubuntu@ip-172-31-19-39:~/apache-tomcat-9.0.110$ cd bin/
ubuntu@ip-172-31-19-39:~/apache-tomcat-9.0.110/bin$ ls
bootstrap.jar       ciphers.sh                  daemon.sh      setclasspath.bat  startup.sh            version.bat
catalina-tasks.xml  commons-daemon-native.tar.gz  digest.bat     setclasspath.sh   tomcat-juli.jar       version.sh
catalina.bat        commons-daemon.jar          digest.sh      shutdown.bat      tomcat-native.tar.gz
catalina.sh         configtest.bat              makebase.bat   shutdown.sh       tool-wrapper.bat
ciphers.bat         configtest.sh               makebase.sh    startup.bat       tool-wrapper.sh
```

b. Run the Shutdown Script in the bin directory:

i. ./shutdown.sh

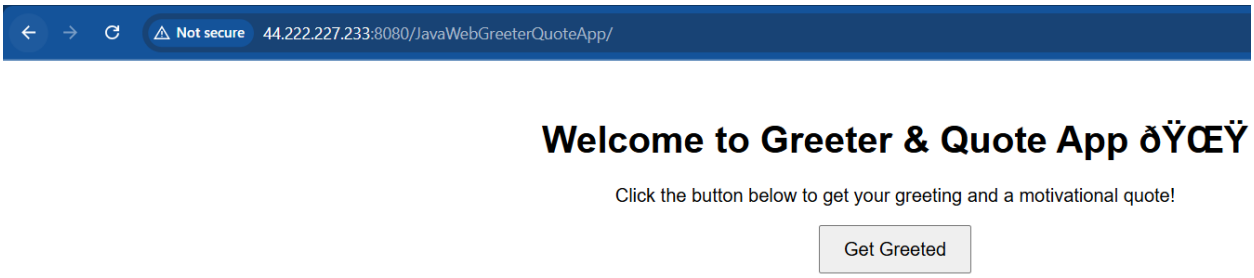
```
ubuntu@ip-172-31-19-39:~/apache-tomcat-9.0.110/bin$ ./shutdown.sh
Using CATALINA_BASE:   /home/ubuntu/apache-tomcat-9.0.110
Using CATALINA_HOME:   /home/ubuntu/apache-tomcat-9.0.110
Using CATALINA_TMPDIR: /home/ubuntu/apache-tomcat-9.0.110/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/ubuntu/apache-tomcat-9.0.110/bin/bootstrap.jar:/home/ubuntu/apache-tomcat-9.0.110/bin/tomca
t-juli.jar
Using CATALINA_OPTS:
NOTE: Picked up JDK_JAVA_OPTIONS:  --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.lang.invoke=AL
L-UNNAMED --add-opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java
.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transpor
t=ALL-UNNAMED
```

c. Run the Startup Script in the bin directory:

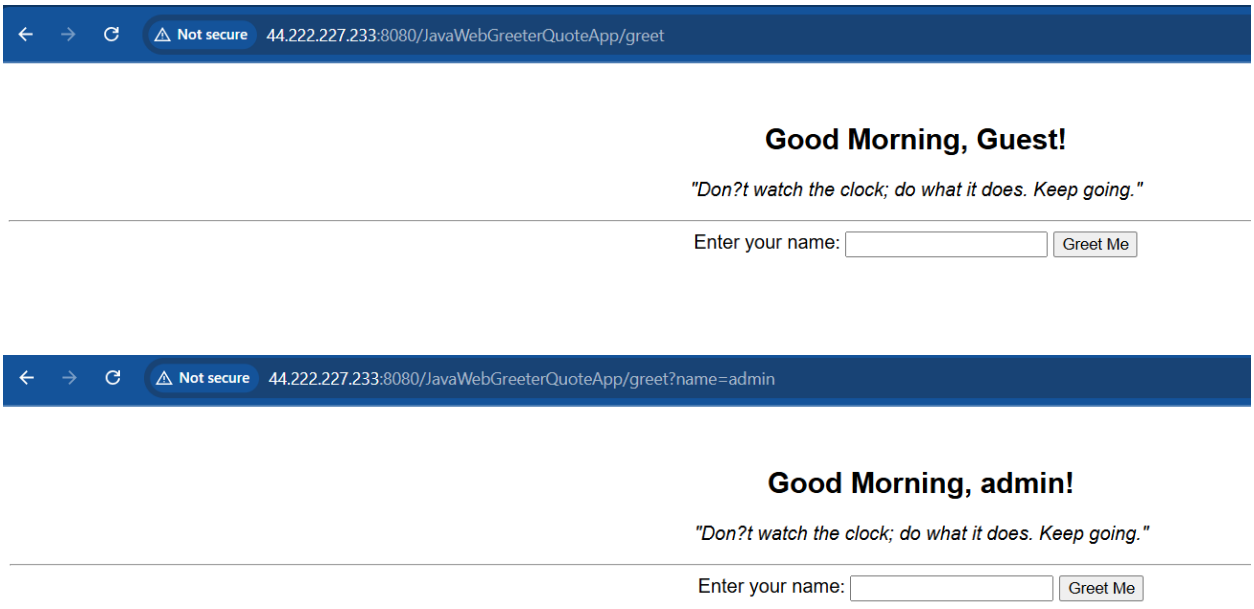
i. ./startup.sh

```
ubuntu@ip-172-31-19-39:~/apache-tomcat-9.0.110/bin$ ./startup.sh
Using CATALINA_BASE:   /home/ubuntu/apache-tomcat-9.0.110
Using CATALINA_HOME:   /home/ubuntu/apache-tomcat-9.0.110
Using CATALINA_TMPDIR: /home/ubuntu/apache-tomcat-9.0.110/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/ubuntu/apache-tomcat-9.0.110/bin/bootstrap.jar:/home/ubuntu/apache-tomcat-9.0.110/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

E. Open Webapp in Tomcat Application table:



F. Test the Application:



6. Conclusion:

This project demonstrates the complete process of building and deploying a simple **Java web greeter and Quote app** using **Maven** and **Tomcat** on **two separate servers** — one for building and another for deployment.

It highlights basic **DevOps workflow concepts**, including:

- Multi-server environment setup
- Maven build management
- WAR file transfer and deployment