Application Deployment with Database (PAAS model)

Step:1

→Launch 2 instances, one for Build and another for Deploy.

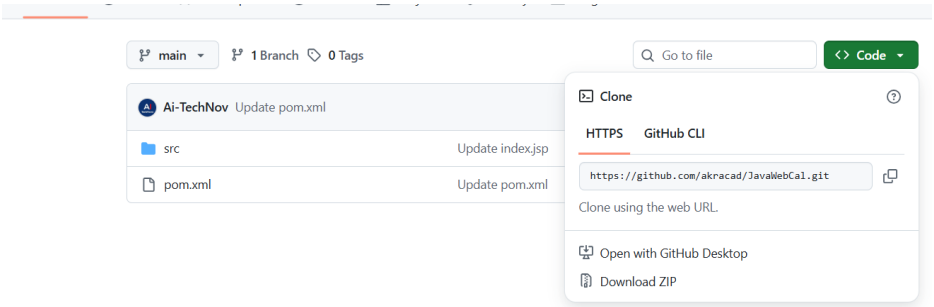
In Build server install java

```
ubuntu@ip-172-31-27-0:~$ 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-172-31-27-0:~$ sudo apt install openjdk-17-jre-headless
```

→Install maven

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-27-0:~$ sudo apt install maven
```

→Clone the code from github.




```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-27-0:~$ git clone https://github.com/akracad/JavaWebCal.git
```

→Build the code by using “mvn package” command

```
ubuntu@ip-172-31-27-0:~/JavaWebCal$ >pom.xml
ubuntu@ip-172-31-27-0:~/JavaWebCal$ vi pom.xml
ubuntu@ip-172-31-27-0:~/JavaWebCal$ mvn package
```

→The build will be Success and the output will be shown as below

```
[INFO] Packaging webapp
[INFO] Assembling webapp [webapp] in [/home/ubuntu/JavaWebCal/target/webapp]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/ubuntu/JavaWebCal/src/main/webapp]
[INFO] Building war: /home/ubuntu/JavaWebCal/target/webapp.war
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  20.638 s
[INFO] Finished at: 2025-10-07T06:35:36Z
[INFO] ------------------------------------------------------------------------
ubuntu@ip-172-31-27-0:~/JavaWebCal$
```

Step :2

Now in another server which is used to deploy the application,

→Install Java (same step as installed in Build server)

→Install Tomcat.

```
ubuntu@ip-172-31-26-7:~$ wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.110/bin/apache-tomcat-9.0.110.tar.gz
```

→We get a tar file, then untar the tar file.

```
ubuntu@ip-172-31-26-7:~$ ls
apache-tomcat-9.0.110.tar.gz
ubuntu@ip-172-31-26-7:~$ tar -xvf apache-tomcat-9.0.110.tar.gz
```

→Rename it.

```
ubuntu@ip-172-31-26-7:~$ ls
apache-tomcat-9.0.110  apache-tomcat-9.0.110.tar.gz
ubuntu@ip-172-31-26-7:~$ mv apache-tomcat-9.0.110 tomcat
```

→Comment in webapps/host manager and manager /META/context.xml

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

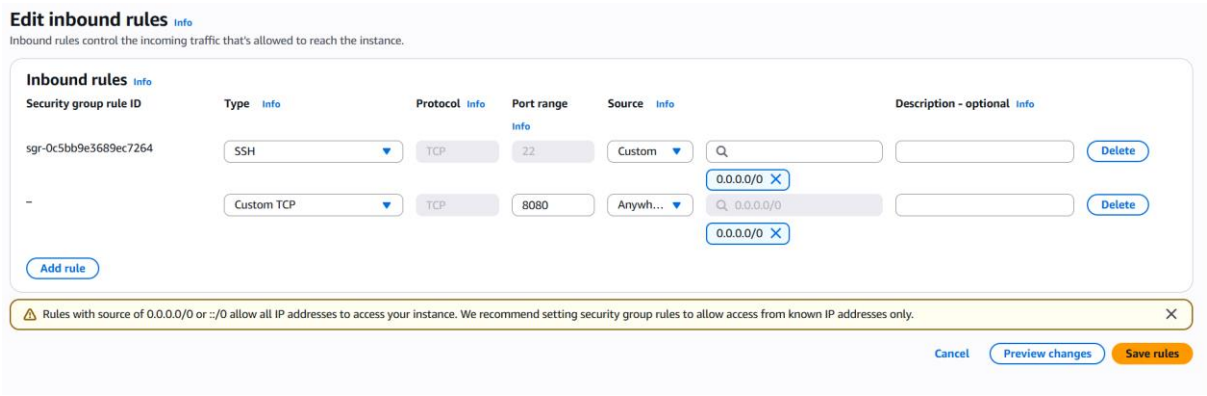

```
<!--  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->
```


➔Change the password in conf/tomcat-user.xml

```
<user username="admin" password="admin" roles="manager-gui"/>
<user username="robot" password="admin" roles="manager-script"/>

<!--
```

→Give port number in Security group as 8080

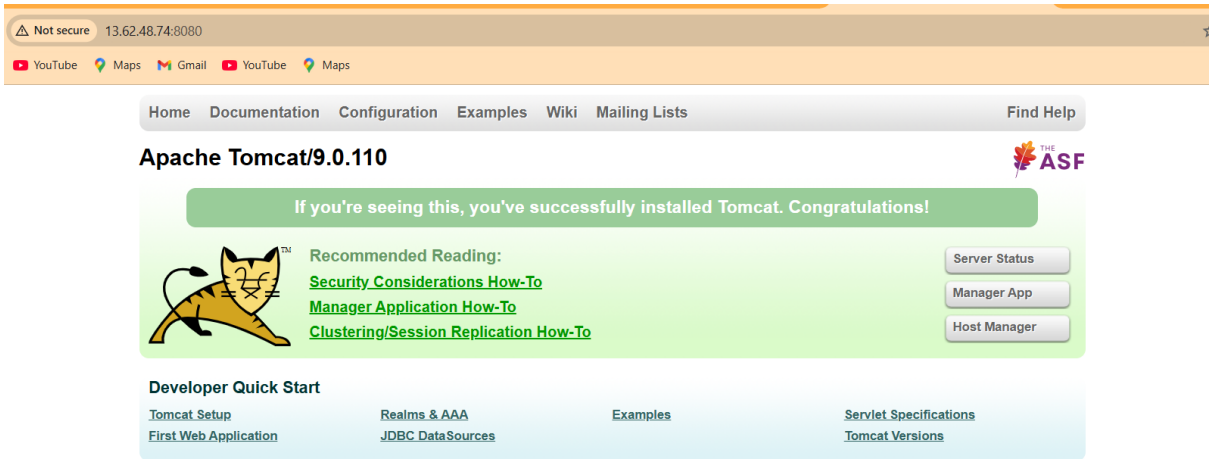


→Start tomcat

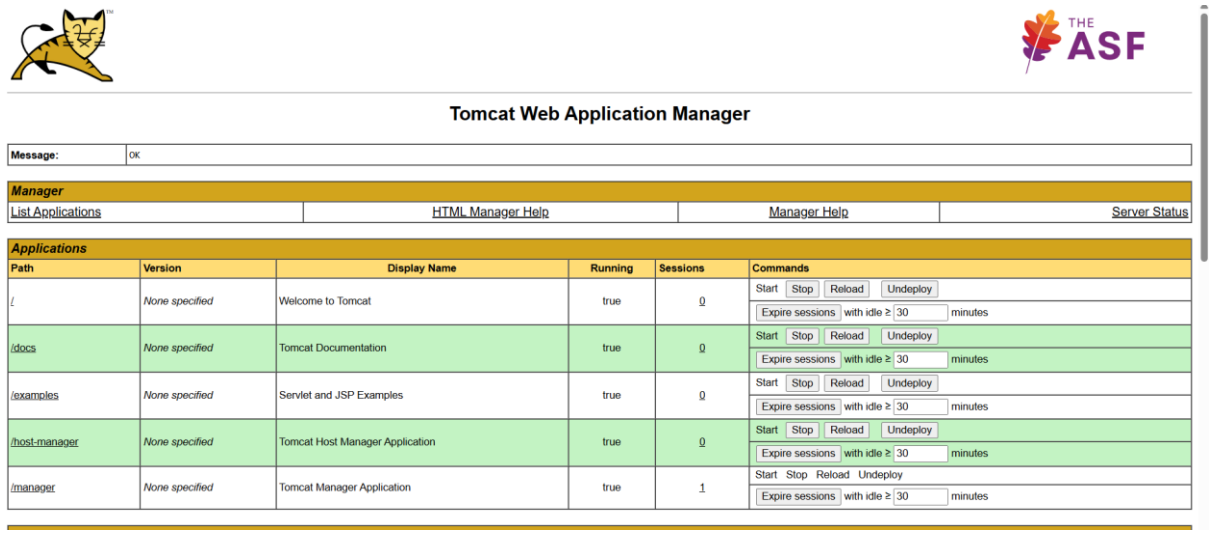
./start.sh

```
ubuntu@ip-172-31-26-7:~/tomcat/bin$ ./startup.sh
Using CATALINA_BASE:   /home/ubuntu/tomcat
Using CATALINA_HOME:   /home/ubuntu/tomcat
Using CATALINA_TMPDIR: /home/ubuntu/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/ubuntu/tomcat/bin/bootstrap.jar:/home/ubuntu/tomcat/bin/tomcat-
juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

→Access Tomcat in web browser with public ip and port number.



→Login to Manager with username and password(admin)



Step :3(Build server).

- Create ssh-keygen
- Cat public id generated
- Copy the public key and paste it in Deploy server (Authorized keys.

Step:4(Deploy server)

→ Go to .ssh

→ vi authorized keys paste the keys generated in Bulid server.

Step:5

→ Copying the War file from Build server to Deploy server.
(scp /home/ubuntu/Javawebcal/target/*.war ubuntu@deploy-public-ip:/home/ubuntu/tomcat/webapps).

```
ubuntu@ip-172-31-26-7:~/tomcat/webapps$ ls
ROOT  docs  examples  host-manager  manager  webapp  webapp.war
```

```
ubuntu@ip-172-31-27-0:~$ scp /home/ubuntu/JavaWebCal/target/*.war ubuntu@13.62.48.74:/home/u
buntu/tomcat/webapps
webapp.war                                    100% 3901     8.1MB/s   00:00
```

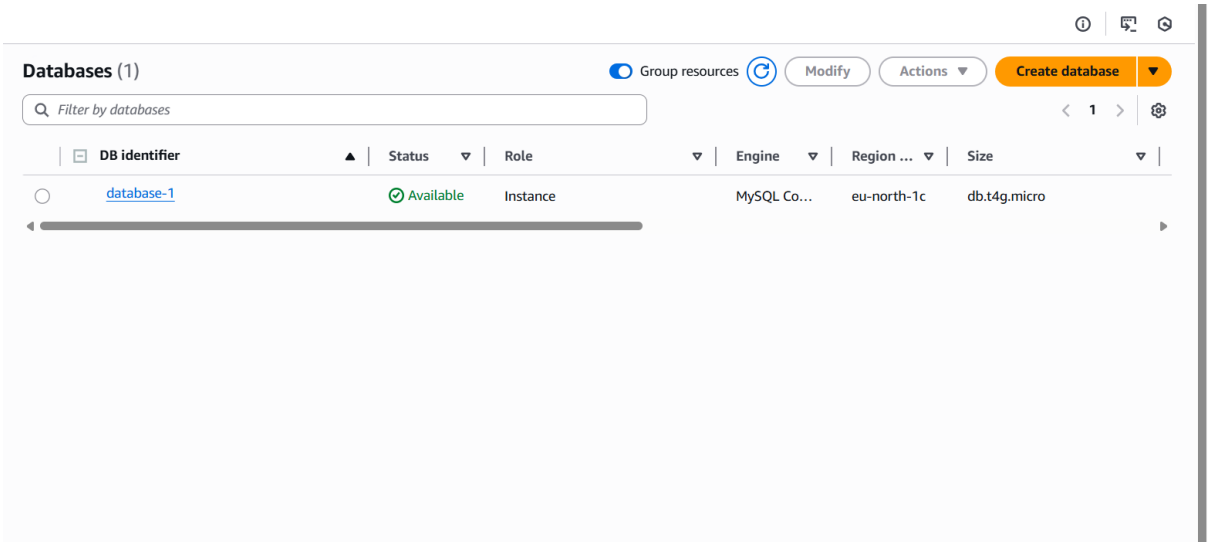
Tomcat Web Application Manager

Message:	OK

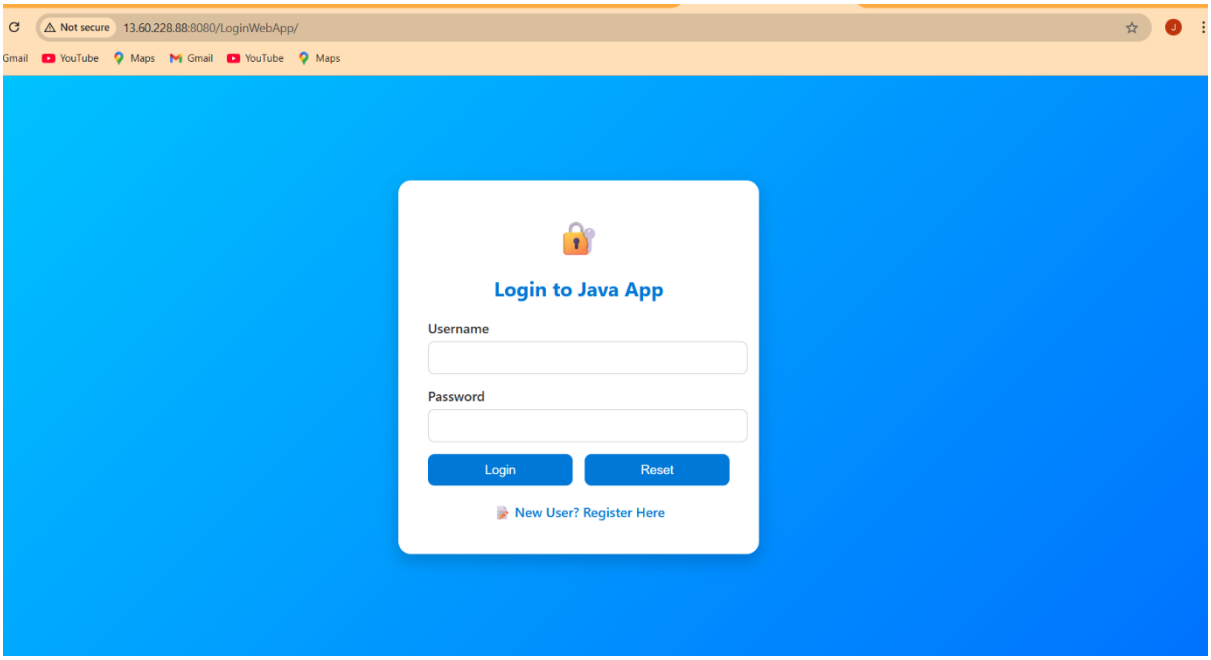
Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/webapp	None specified	Servlet	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

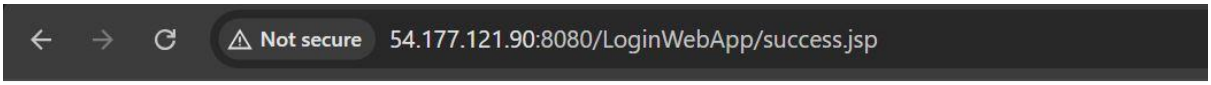
Create data base:



We will get app login page



Final output



Welcome subbu Log out