

Application Deployment by Database

Step:1

*Launch 2 instances and name it as Build and Deploy.

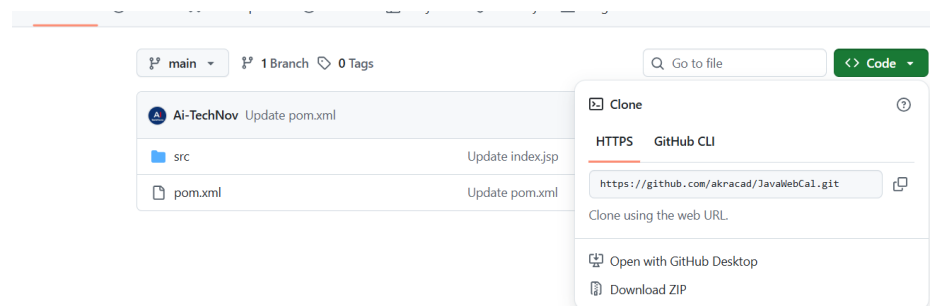
*Install java

```
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No VM guests are running outdated hypervisor (qemu) binaries on this host.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No VM guests are running outdated hypervisor (qemu) binaries on this host.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
```

*mvn package

```
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
```

*Here the build will be Success.

```
[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

Take another server named as (Deploy).

*Here we need to install Java and Tomcat.

*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 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
```

→untar the tomcat.

```
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

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	
sgr-0c5bb9e3689ec7264	SSH	TCP	22	Custom	0.0.0.0/0	Delete
-	Custom TCP	TCP	8080	Anywh...	0.0.0.0/0	Delete

[Add rule](#)

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

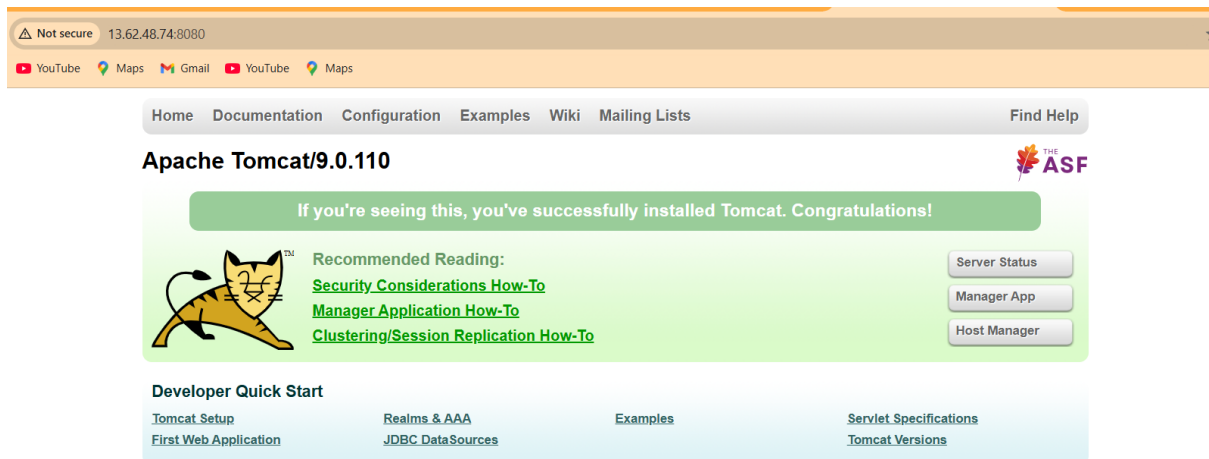
[Cancel](#) [Preview changes](#) [Save rules](#)

→ Start tomcat

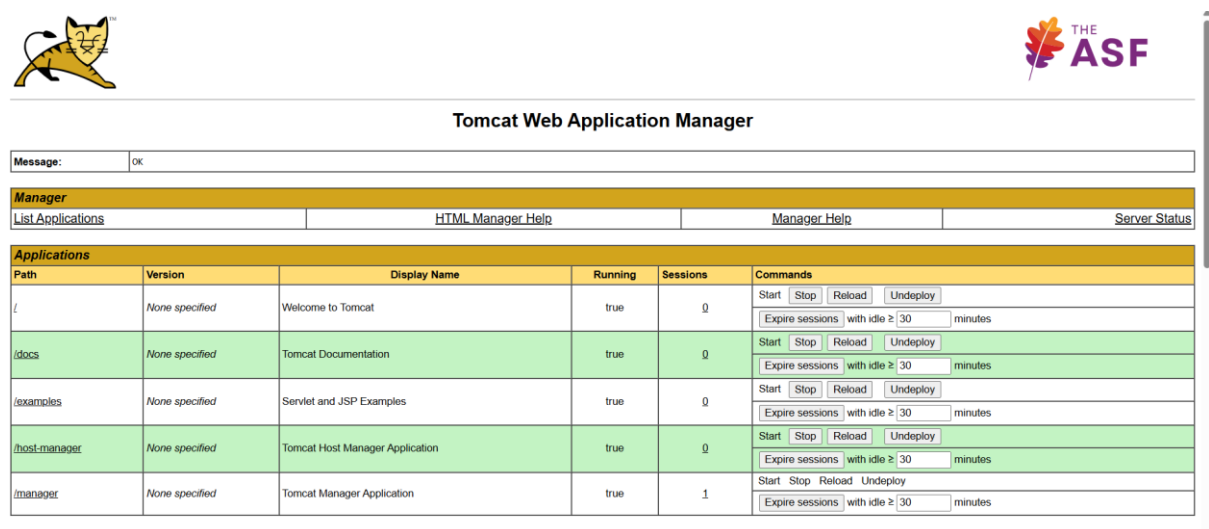
./start.sh

```
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).

```
ubuntu@ip-172-31-27-0:~$ cd .ssh/
ubuntu@ip-172-31-27-0:~/.ssh$ ls
authorized_keys  id_ed25519  id_ed25519.pub
ubuntu@ip-172-31-27-0:~/.ssh$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIEQtEz++8uG8v4KDNb54KbgC9vryMChQi9TyvQKzd4qJ ubuntu@ip-172-31-27-0
ubuntu@ip-172-31-27-0:~/.ssh$
```

Step:4(Deploy server)

*Go to .ssh

*vi authorized keys paste the keys generated in Build 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-172-31-26-7:~/tomcat/webapps$
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
ubuntu@ip-172-31-27-0:~$
```

Tomcat Web Application Manager

Message:

OK

Manager

List Applications

HTML Manager Help

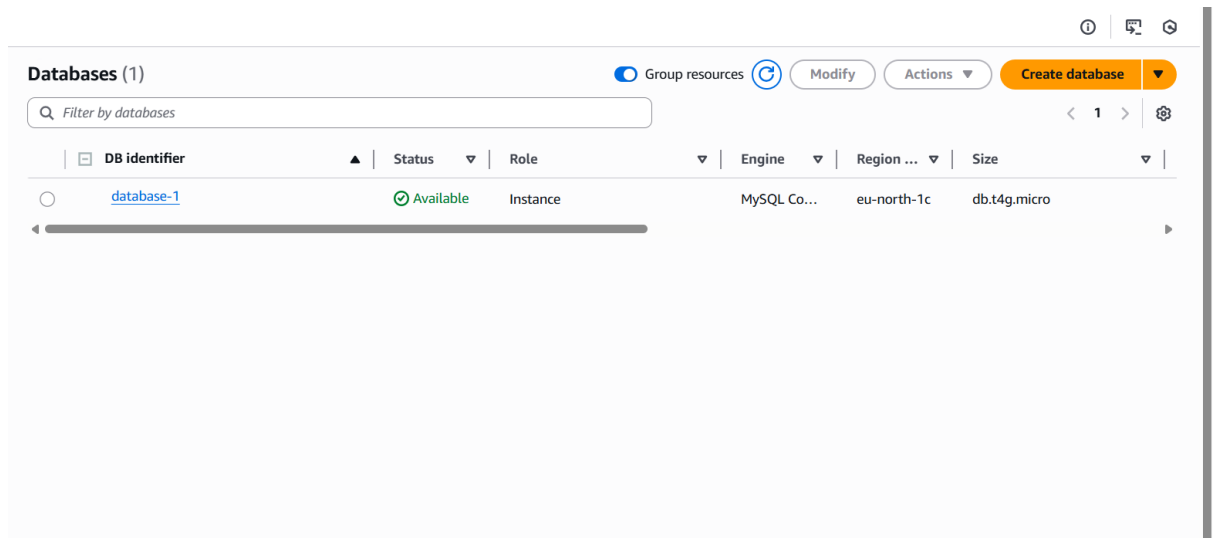
Manager Help

Server Status

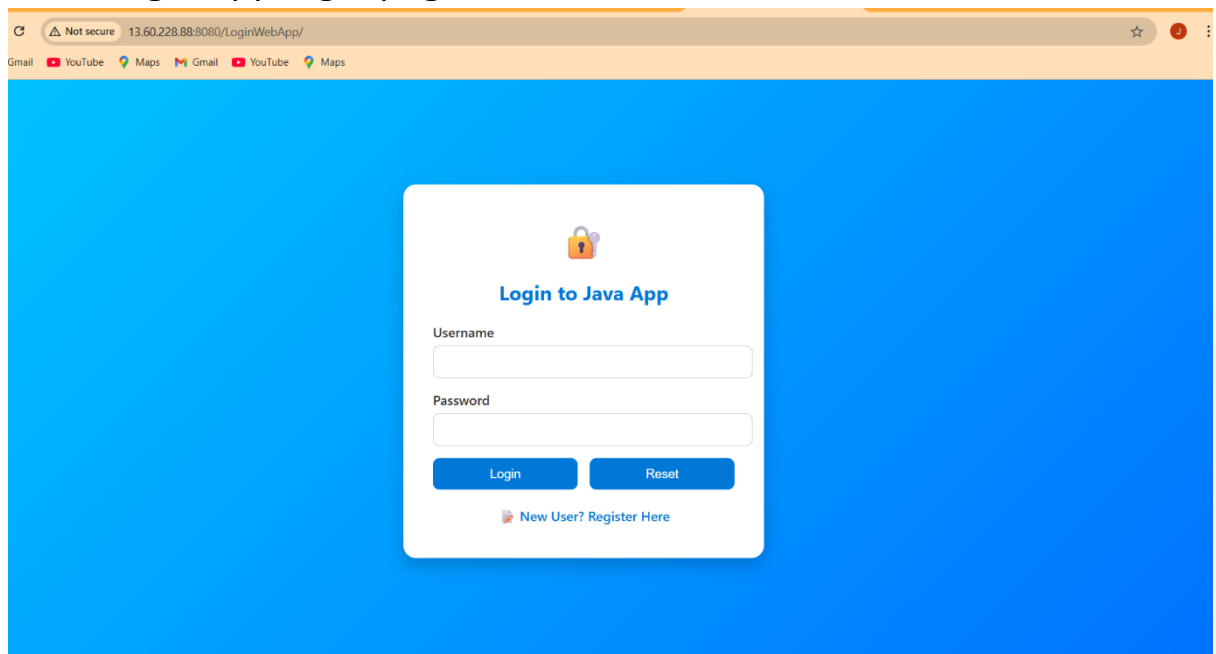
Applications

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

Create data base:



We will get app login page



Final output

