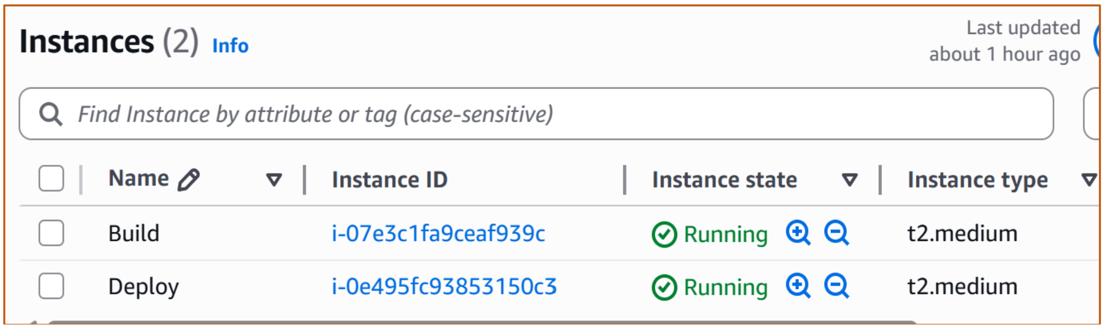
How to build a Java application by using Maven and deploy it on a Tomcat server.

Step 1: Create two virtual machines

- Name the servers, such as the build and deploy server.



Instances (2) Info

Last updated
about 1 hour ago

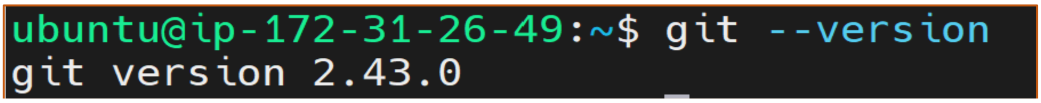
Find Instance by attribute or tag (case-sensitive)

□	Name	Instance ID	Instance state	Instance type
□	Build	i-07e3c1fa9ceaf939c	✓ Running	t2.medium
□	Deploy	i-0e495fc93853150c3	✓ Running	t2.medium

Step 2: The steps below are to be performed on the Build Server.

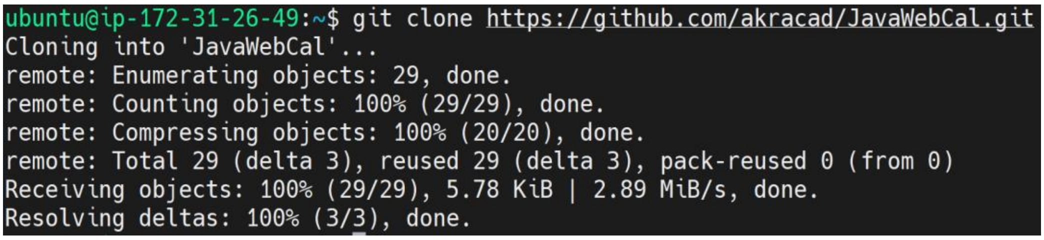
i. Clone the Git repo:

- Check if Git is installed or not by using “git –version”.
- If Git is not installed, install git.



```
ubuntu@ip-172-31-26-49:~$ git --version
git version 2.43.0
```

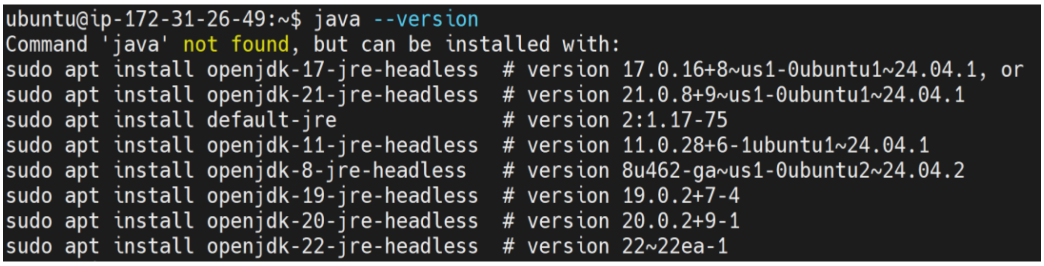
- git clone <your repo url>



```
ubuntu@ip-172-31-26-49:~$ git clone https://github.com/akracad/JavaWebCal.git
Cloning into 'JavaWebCal'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 29 (delta 3), reused 29 (delta 3), pack-reused 0 (from 0)
Receiving objects: 100% (29/29), 5.78 KiB | 2.89 MiB/s, done.
Resolving deltas: 100% (3/3), done.
```

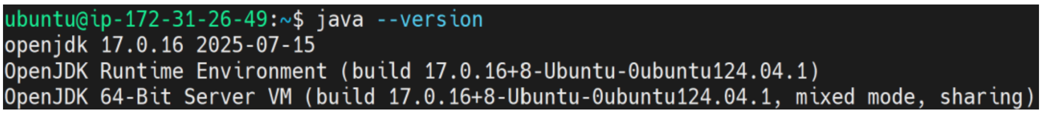
ii. Install the Java:

- First check the Java is installed or not by using “java –version”



```
ubuntu@ip-172-31-26-49:~$ 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
```

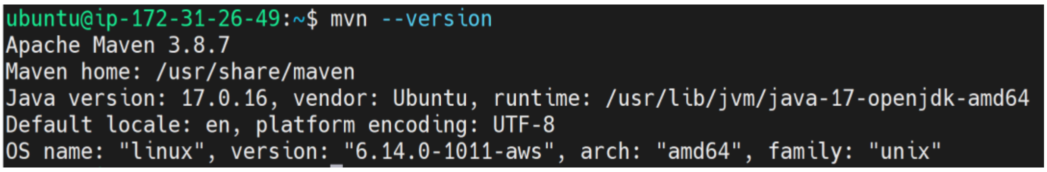
- If Java is not installed, install Java by using “sudo apt install openjdk-17-jre-headless”. You will get these commands when you check the Java version.
- Java version depends on the code written for which version.
- Again, check if Java is installed properly or not using “java –version”.



```
ubuntu@ip-172-31-26-49:~$ java --version
openjdk 17.0.16 2025-07-15
OpenJDK Runtime Environment (build 17.0.16+8-Ubuntu-0ubuntu124.04.1)
OpenJDK 64-Bit Server VM (build 17.0.16+8-Ubuntu-0ubuntu124.04.1, mixed mode, sharing)
```

iii. Install the Maven:

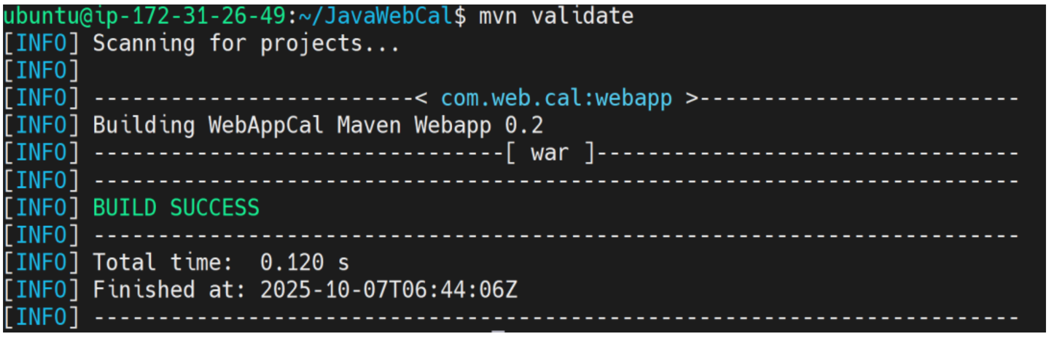
- First check the Maven is installed or not by using “mvn –version”
- If not, install Maven by using the command “sudo apt install mvn”.
- After installation, check whether Maven is installed properly or not using “mvn –version”.



```
ubuntu@ip-172-31-26-49:~$ mvn --version
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 17.0.16, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.14.0-1011-aws", arch: "amd64", family: "unix"
```

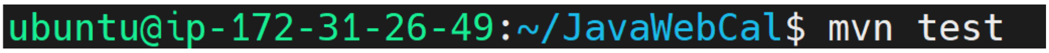
iv. Build the Application:

- Change directory to the project “cd JavaWebCal”.
- Run the “mvn validate”.



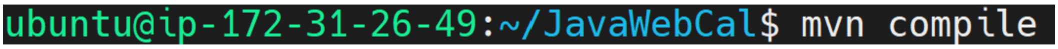
```
ubuntu@ip-172-31-26-49:~/JavaWebCal$ mvn validate
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< com.web.cal:webapp >------------------------
[INFO] Building WebAppCal Maven Webapp 0.2
[INFO] --------------------------------[ war ]--------------------------------
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.120 s
[INFO] Finished at: 2025-10-07T06:44:06Z
[INFO] ------------------------------------------------------------------------
```

- Run the “mvn test”.



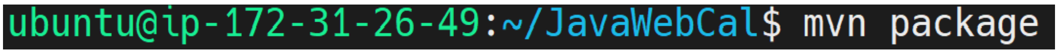
```
ubuntu@ip-172-31-26-49:~/JavaWebCal$ mvn test
```

- Run the “mvn compile”.



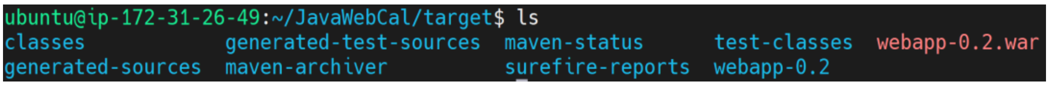
```
ubuntu@ip-172-31-26-49:~/JavaWebCal$ mvn compile
```

- Run the “mvn package”.



```
ubuntu@ip-172-31-26-49:~/JavaWebCal$ mvn package
```

- After “mvn package” war file is created in the target folder.

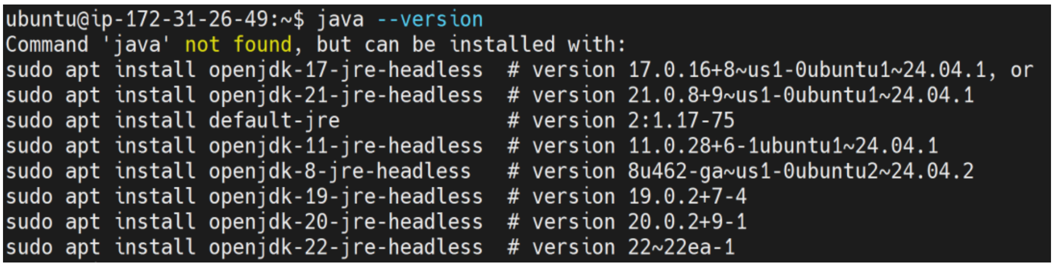


```
ubuntu@ip-172-31-26-49:~/JavaWebCal/target$ ls
classes            generated-test-sources  maven-status      test-classes  webapp-0.2.war
generated-sources  maven-archiver          surefire-reports  webapp-0.2
```

Step 3: The steps below are to be performed on the Deploy Server

i. **Install the Java:**

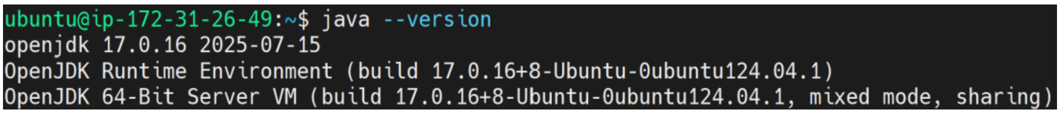
- First check the Java is installed or not by using “java –version”



```
ubuntu@ip-172-31-26-49:~$ 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
```

- If Java is not installed, install Java by using “sudo apt install openjdk-17-jre-headless”. You will get these commands when you check the Java version.
- Java version depends on the code written for which version.

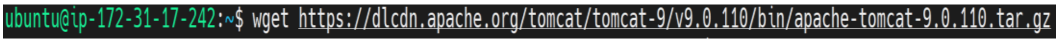
- Again, check if Java is installed properly or not using “java –version”.



```
ubuntu@ip-172-31-26-49:~$ java --version
openjdk 17.0.16 2025-07-15
OpenJDK Runtime Environment (build 17.0.16+8-Ubuntu-0ubuntu124.04.1)
OpenJDK 64-Bit Server VM (build 17.0.16+8-Ubuntu-0ubuntu124.04.1, mixed mode, sharing)
```

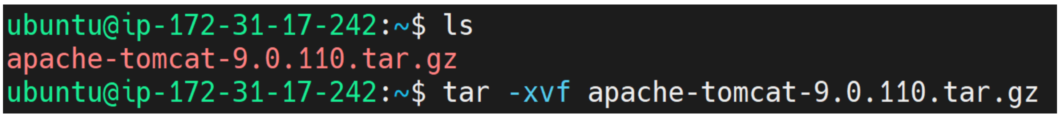
ii. **Install the Tomcat Server:**

- First, copy the Tomcat version 9 link from the official Tomcat website.
- Download by using wget.



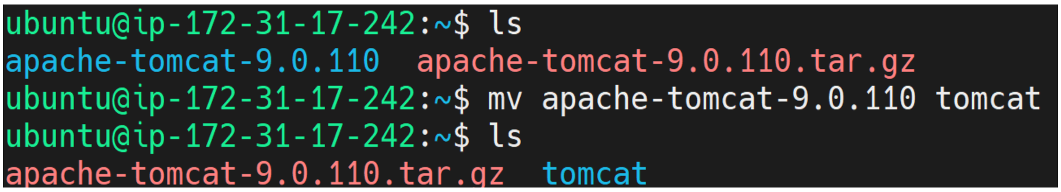
```
ubuntu@ip-172-31-17-242:~$ wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.110/bin/apache-tomcat-9.0.110.tar.gz
```

- Next, extract the tar file.



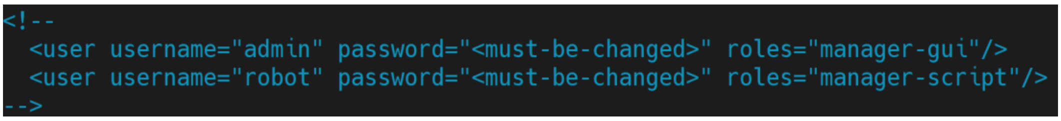
```
ubuntu@ip-172-31-17-242:~$ ls
apache-tomcat-9.0.110.tar.gz
ubuntu@ip-172-31-17-242:~$ tar -xvf apache-tomcat-9.0.110.tar.gz
```

- Next, rename the default Tomcat server name to a short name like "tomcat"



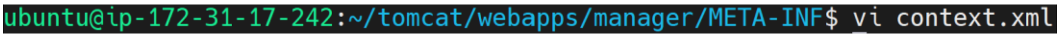
```
ubuntu@ip-172-31-17-242:~$ ls
apache-tomcat-9.0.110  apache-tomcat-9.0.110.tar.gz
ubuntu@ip-172-31-17-242:~$ mv apache-tomcat-9.0.110 tomcat
ubuntu@ip-172-31-17-242:~$ ls
apache-tomcat-9.0.110.tar.gz  tomcat
```

- Next, switch to the bin folder and start the server by using "./startup sh"
- Next, uncommit the lines below and give username and password in the "tomcat-users.xml" file, which is located in tomcat/conf.

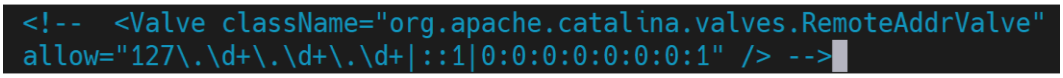


```
<!--
  <user username="admin" password="<must-be-changed>" roles="manager-gui"/>
  <user username="robot" password="<must-be-changed>" roles="manager-script"/>
-->
```

- Next, comment out the lines below in context.xml, and this file is located in "tomcat/webapps/manager/META-INF".

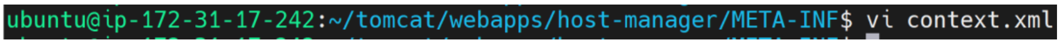


```
ubuntu@ip-172-31-17-242:~/tomcat/webapps/manager/META-INF$ vi context.xml
```

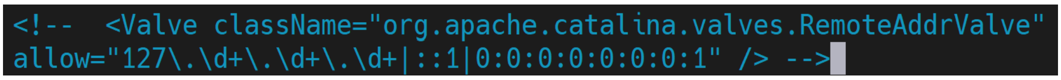


```
 <!--  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
 allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->
```

- Next, comment out the lines below in context.xml, and this file is located in "tomcat/webapps/host-manager/META-INF".

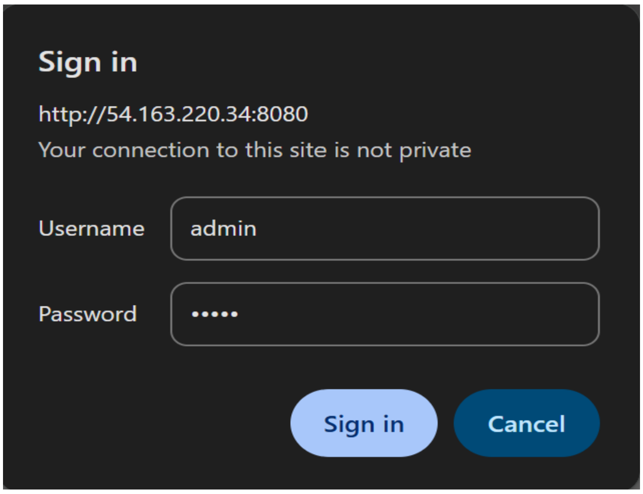


```
ubuntu@ip-172-31-17-242:~/tomcat/webapps/host-manager/META-INF$ vi context.xml
```



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

- Next, switch to the bin folder and shut down and start the server because if we modify any configurations first, we need to restart the server; otherwise changes will not reflect.
- Log in to the Tomcat Manager by using the username and password that we gave in the tomcat-users.xml file.



Sign in

http://54.163.220.34:8080

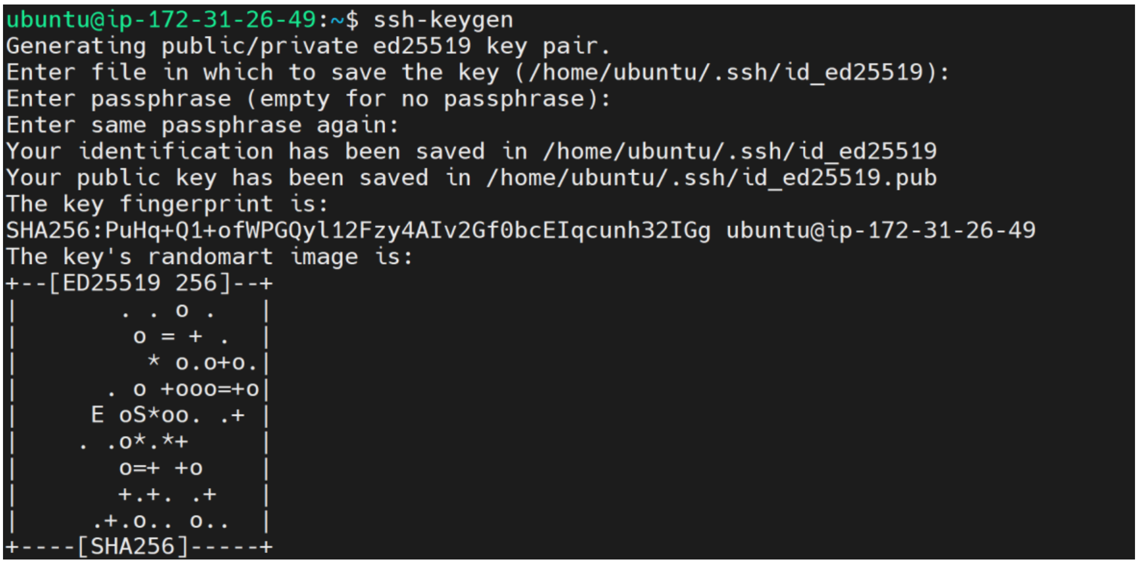
Your connection to this site is not private

Username admin

Password •••••

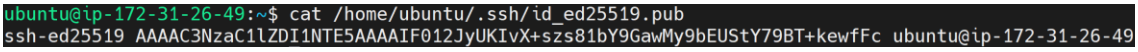
Step 4: Move the artefact from the Build server to the Deploy server:

i. First, generate the SSH key in the build server by using the "ssh-keygen" command.



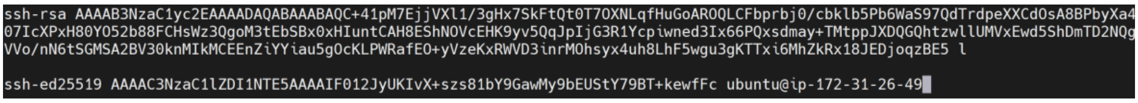
```
ubuntu@ip-172-31-26-49:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_ed25519
Your public key has been saved in /home/ubuntu/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:PuHq+Q1+ofWPGQyl12Fzy4AIv2Gf0bcEIqcunh32IGg ubuntu@ip-172-31-26-49
The key's randomart image is:
+--[ED25519 256]--+
|       . . o .   |
|        o = + .  |
|         * o.o+o.|
|      . o +ooo=+o|
|     E oS*oo. .+ |
|    . .o*.*+     |
|       o=+ +o    |
|       +.+. .+   |
|     .+.o.. o..  |
+----[SHA256]-----+
```

ii. Copy the public key and path is "/home/ubuntu/.ssh/id_ed25519.pub"



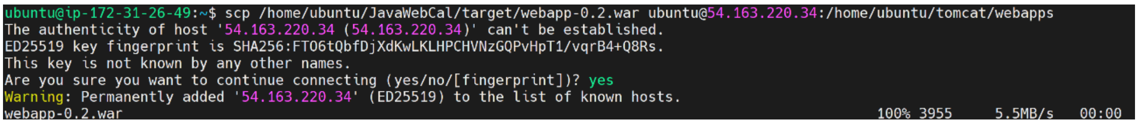
```
ubuntu@ip-172-31-26-49:~$ cat /home/ubuntu/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIF012JyUKIvX+szs81bY9GawMy9bEUStY79BT+kewfFc ubuntu@ip-172-31-26-49
```

iii. Paste the public key in the ".ssh/authorized_keys" file in the Deploy server.



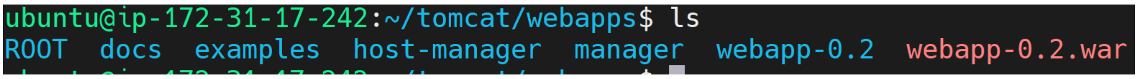
```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC+41pM7EjjVXl1/3gHx7SkFtQt0T7OXNLqfHuGoAROQLCFbprbj0/cbklb5Pb6WaS97QdTrdpeXXCdOsA8BPbyXa4
07IcXPxH80YO52b88FCHsWz3QgoM3tEbSBx0xHIuntCAH8EShNOVcEHK9yv5QqJpIjG3R1Ycpiwned3Ix66PQxsdmay+TMtppJXDQGQhtzwllUMVxEwd5ShDmTD2NQg
VVo/nN6tSGMSA2BV30knMIkMCEEnZiYYiau5gOcKLPWRafEO+yVzeKxRWVD3inrMOhsyx4uh8LhF5wgu3gKTTxi6MhZkRx18JEDjoqzBE5 l

ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIF012JyUKIvX+szs81bY9GawMy9bEUStY79BT+kewfFc ubuntu@ip-172-31-26-49
```

iv. Next, run the secure copy command on the build server to move the artifact copy from the build server to the deploy server.



```
ubuntu@ip-172-31-26-49:~$ scp /home/ubuntu/JavaWebCal/target/webapp-0.2.war ubuntu@54.163.220.34:/home/ubuntu/tomcat/webapps
The authenticity of host '54.163.220.34 (54.163.220.34)' can't be established.
ED25519 key fingerprint is SHA256:FT06tQbfDjXdKwLKLHPCHVNzGQPvHpT1/vqrB4+Q8Rs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.163.220.34' (ED25519) to the list of known hosts.
webapp-0.2.war                                                                        100% 3955     5.5MB/s   00:00
```

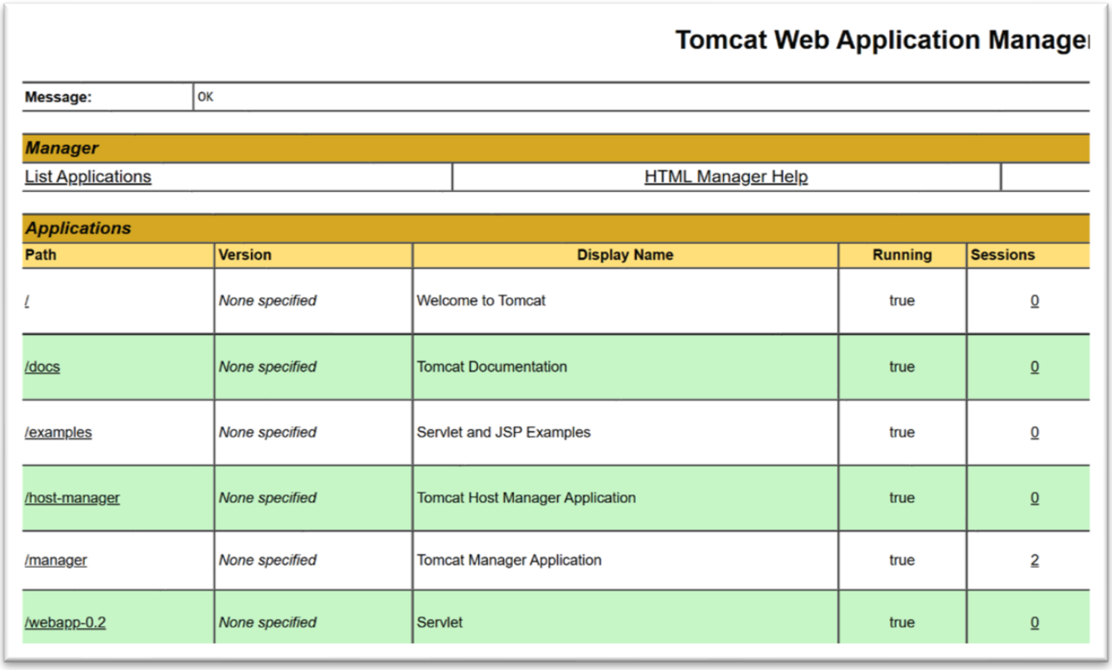
v. Next, check the artifact in webapps on the deploy server.



```
ubuntu@ip-172-31-17-242:~/tomcat/webapps$ ls
ROOT  docs  examples  host-manager  manager  webapp-0.2  webapp-0.2.war
```

Step 5: Access the application:

i. First, check the application path in Tomcat Manager.



Tomcat Web Application Manage

Message:	OK

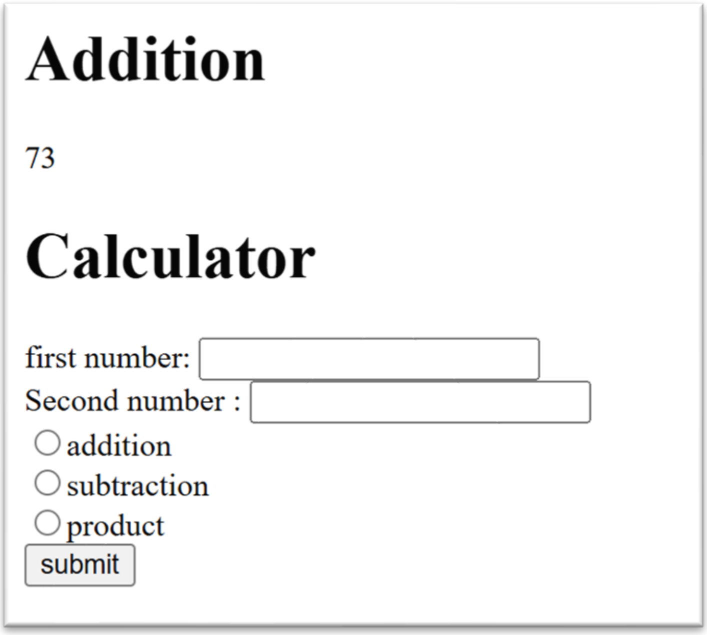
Manager

List Applications | HTML Manager Help

Applications

Path	Version	Display Name	Running	Sessions
/	*None specified*	Welcome to Tomcat	true	0
/docs	*None specified*	Tomcat Documentation	true	0
/examples	*None specified*	Servlet and JSP Examples	true	0
/host-manager	*None specified*	Tomcat Host Manager Application	true	0
/manager	*None specified*	Tomcat Manager Application	true	2
/webapp-0.2	*None specified*	Servlet	true	0

ii. Next, click on the application path on Tomcat Manager, and it will redirect to the application.



Addition

73

Calculator

first number:

Second number :

○ addition
○ subtraction
○ product