

WEB APPLICATION DEPLOYMENT IN TOMCAT SERVER THROUGH JENKINS DASHBOARD (UBUNTU 22.04).

To deploy web application in tomcat server through Jenkins dashboard, follow the below mentioned procedure.

Prerequisites are

- 1) Jenkins Server. (t2.Micro – Instance1)**
- 2) Tomcat Server. (t2.Micro – Instance2)**

Step by Step Procedure For Jenkins Server Setup.

First run the following command in terminal.

sudo apt-get update

After executing the above command, the following screen will display.



```
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f
Get:23 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en
Get:25 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f
Get:26 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-r
Get:27 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages
Get:28 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en
Get:29 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f
Get:30 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-r
Get:31 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [658 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [153 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [11.2 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [656 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [104 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [53 kB]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [769 kB]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [140 kB]
Get:39 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.4 kB]
Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [36.5 kB]
Get:41 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7060 B]
Get:42 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [26 kB]
Fetched 26.5 MB in 5s (5538 kB/s)
Reading package lists... Done
```

Now, we can proceed further in Jenkins installation, by executing the following commands.

This is the Debian package repository of Jenkins to automate installation and upgrade. To use this repository, first add the key to your system.

```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \
/usr/share/keyrings/jenkins-keyring.asc > /dev/null
```

Then add a Jenkins apt repository entry, by executing the following command.

```
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
```

As we have added Jenkins apt repository, we have to update the apt package list. By executing the following commands.

```
sudo apt-get update
```

After executing the above commands, the following screen will appear.

```
ubuntu@JenkinsServer:~$ sudo apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:5 https://pkg.jenkins.io/debian-stable binary/ Release ────────── Jenkins added to apt repository
Get:6 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [873 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [212 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [15.6 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [678 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [108 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [965 kB]
```

To install Jenkins, prerequisite is JAVA. To install it including supported packages, execute the following command.

```
sudo apt-get install fontconfig openjdk-11-jre -y
```

After executing the above command, the following screen will be displayed and java will be installed successfully.

```
auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/rmiregistry to provide /usr/bin/rmiregistry (rmiregistry) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/pack200 to provide /usr/bin/pack200 (pack200) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/unpack200 to provide /usr/bin/unpack200 (unpack200) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/lib/jexec to provide /usr/bin/jexec (jexec) in auto mode
Scanning processes ...
Scanning linux images ...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

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

To validate the JAVA installation, execute the following commands in terminal.

```
java --version
```

After executing that command, the following screen will appear.

```
ubuntu@JenkinsServer:~$ java --version  
openjdk 11.0.20 2023-07-18  
OpenJDK Runtime Environment (build 11.0.20+8-post-Ubuntu-1ubuntu122.04)  
OpenJDK 64-Bit Server VM (build 11.0.20+8-post-Ubuntu-1ubuntu122.04, mixed mode, sharing)
```

Displays java version

Finally, execute the following commands for Jenkins installation.

```
sudo apt-get install jenkins -y
```

After executing the above commands, jenkins will be installed successfully and to validate its installation and to see its status execute the following commands in terminal.

```
sudo systemctl status jenkins
```

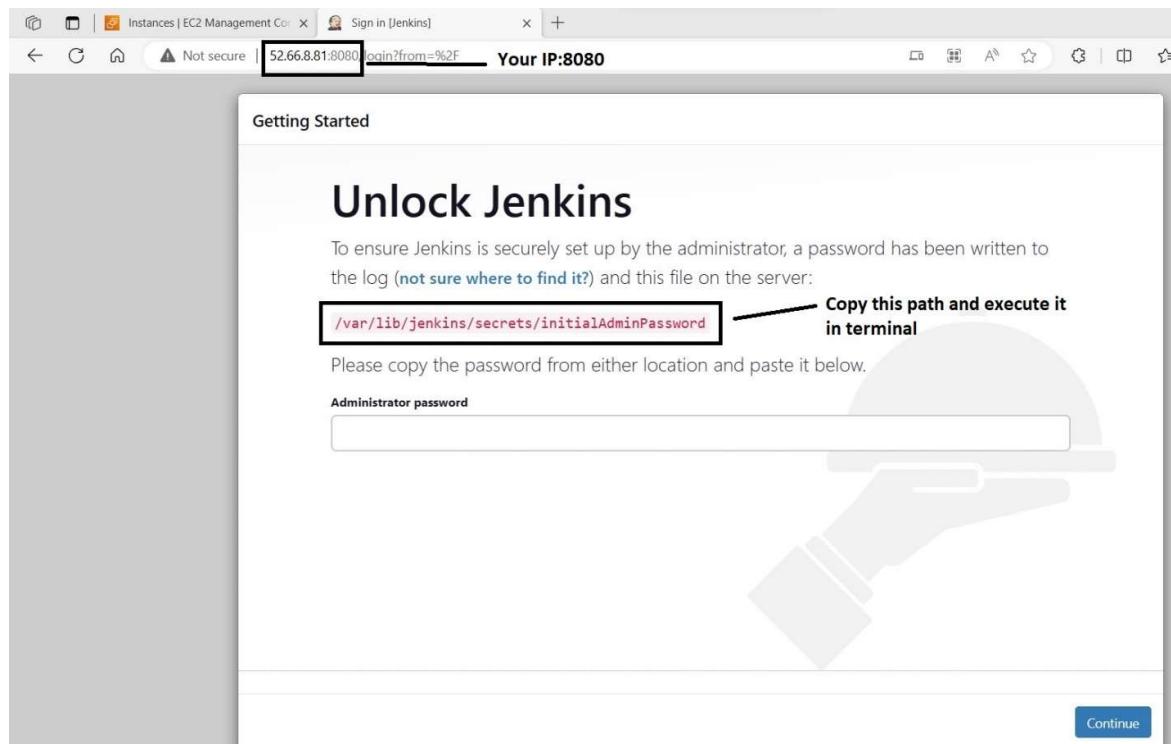
After executing the above commands, the following screen will appear.

```
ubuntu@JenkinsServer:~$ sudo systemctl status jenkins  
● jenkins.service - Jenkins Continuous Integration Server  
  Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)  
  Active: active (running) since Fri 2023-08-11 00:49:52 UTC; 2min 20s ago  
    Main PID: 23380 (java)  
      Tasks: 36 (limit: 1141)  
     Memory: 236.0M  
        CPU: 21.611s  
       CGroup: /system.slice/jenkins.service  
             └─23380 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/v  
  
Aug 11 00:49:37 JenkinsServer jenkins[23380]: *****  
Aug 11 00:49:37 JenkinsServer jenkins[23380]: *****  
Aug 11 00:49:37 JenkinsServer jenkins[23380]: WARNING: An illegal reflective access operation has occurred.  
Aug 11 00:49:37 JenkinsServer jenkins[23380]: WARNING: Illegal reflective access by org.codehaus.groovy.v  
Aug 11 00:49:37 JenkinsServer jenkins[23380]: WARNING: Please consider reporting this to the maintainers.  
Aug 11 00:49:37 JenkinsServer jenkins[23380]: WARNING: Use --illegal-access=warn to enable warnings of fu  
Aug 11 00:49:37 JenkinsServer jenkins[23380]: WARNING: All illegal access operations will be denied in a  
Aug 11 00:49:52 JenkinsServer jenkins[23380]: 2023-08-11 00:49:52.414+0000 [id=29]           INFO      jen  
Aug 11 00:49:52 JenkinsServer jenkins[23380]: 2023-08-11 00:49:52.436+0000 [id=22]           INFO      hud  
Aug 11 00:49:52 JenkinsServer systemd[1]: Started Jenkins Continuous Integration Server.  
lines 1-20/20 (END)
```

After successful installation of jenkins, go to your browser and enter the following URL for further setup.

http://YOUR IP :8080 (In AWS security groups, we have to allow 8080 port number)

After entering the above URL, the following screen will appear.



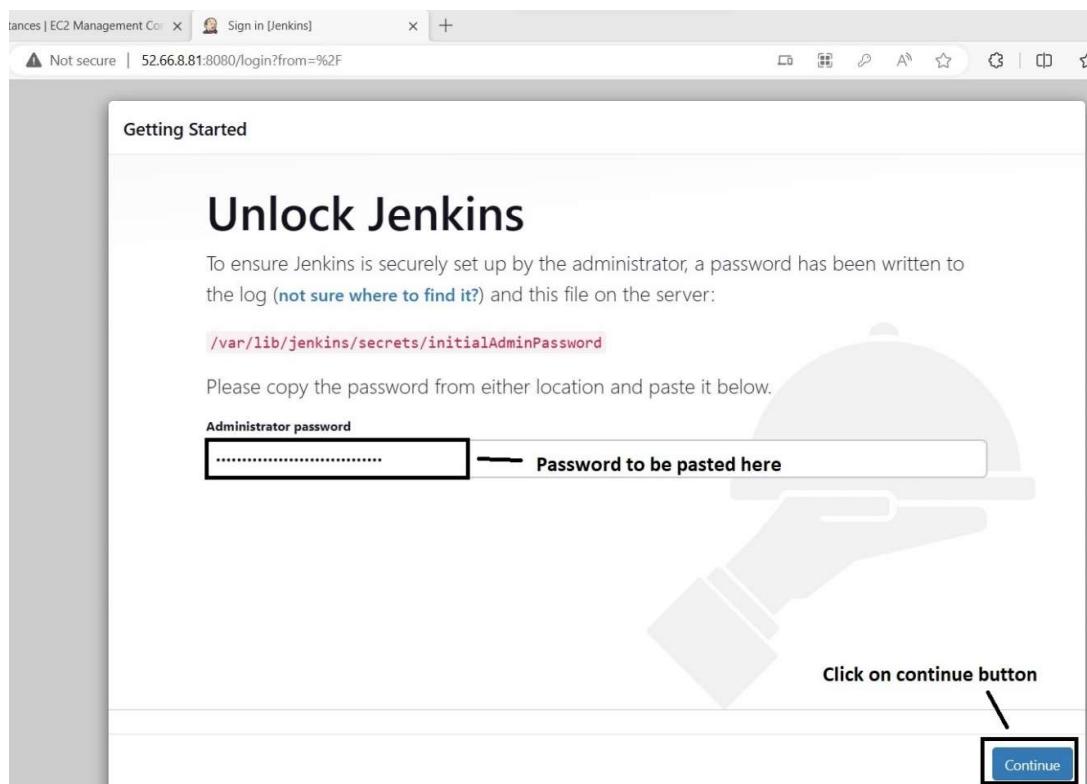
Then you have to copy that given path and execute the following commands in the terminal for initial admin password.

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

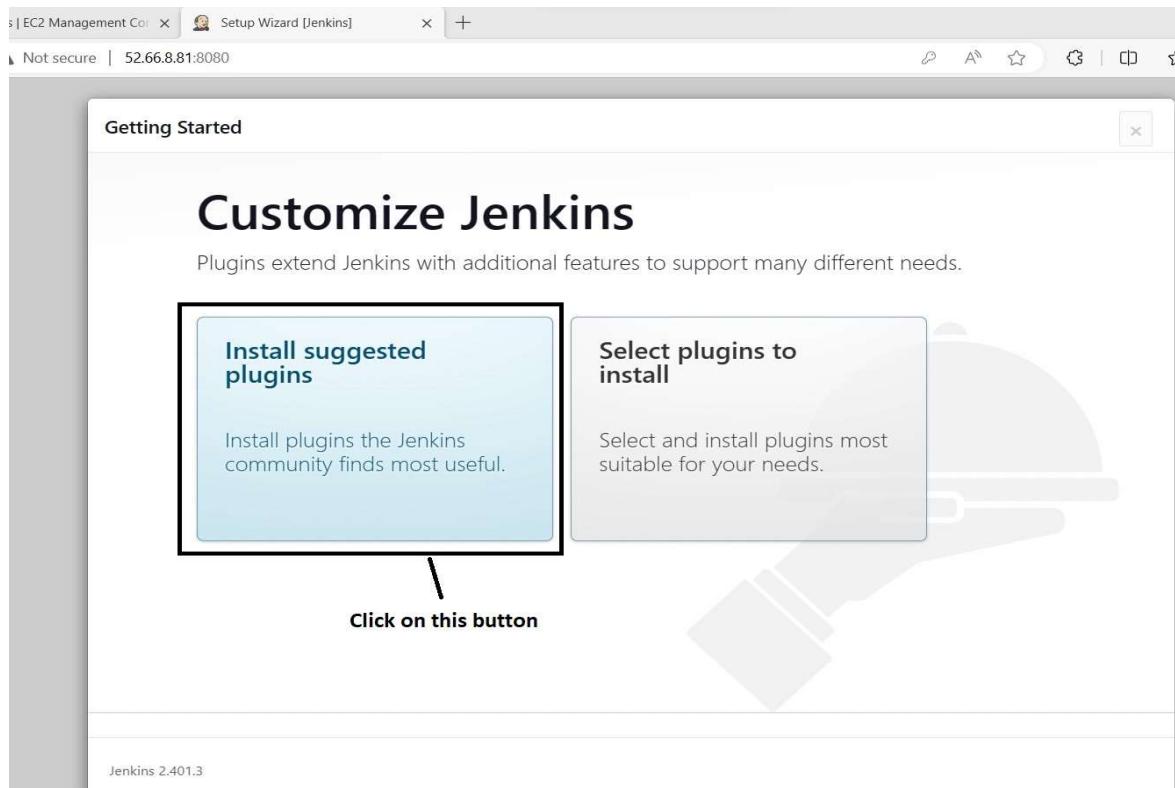
The following output message will be displayed as shown below.

The terminal window shows two tabs: '6. 13.232.106.159 (Tomcat_Server)' and '8. 52.66.8.81 (Jenkins_Server)'. The Jenkins tab is active. It displays the command 'ubuntu@JenkinsServer:~\$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword' followed by the output '005728519dce4300bff9905ce11d2a4e'. A red box highlights this output, and an arrow points to it with the text 'Password to be copied and paste it in browser'.

Then you have to copy that password and pasted in browser (administrator password section) and finally click on continue button. As shown in below screenshot.



After clicking on continue button, the following page will appear.



After clicking on ‘Install suggested plugin’ button, the following page will appear.

The screenshot shows the Jenkins 'Getting Started' page. At the top, there's a header with tabs like 'Management Center' and 'Setup Wizard [Jenkins]'. Below the header, it says 'secure | 52.66.8.81:8080'. The main area has a title 'Getting Started' and a sub-section 'Plugins installation under progress'. A progress bar is shown below the sub-section. To the right of the progress bar, there's a sidebar with sections for 'Ionicons API', 'Folders', and 'OWASP Markup Formatter'. The main content area contains a grid of plugin icons and names. The grid is organized into four columns: 'Folders' (with 'Folders' checked), 'OWASP Markup Formatter' (with a checkmark), 'Build Timeout' (unchecked), 'Credentials Binding' (unchecked). The second column contains 'Timestamper' (unchecked), 'Workspace Cleanup' (unchecked), 'Ant' (unchecked), 'Gradle' (unchecked). The third column contains 'Pipeline' (unchecked), 'GitHub Branch Source' (unchecked), 'Pipeline: GitHub Groovy Libraries' (unchecked), 'Pipeline: Stage View' (unchecked). The fourth column contains 'Git' (unchecked), 'SSH Build Agents' (unchecked), 'Matrix Authorization Strategy' (unchecked), 'PAM Authentication' (unchecked). The bottom right corner of the grid has a note '** - required dependency'. A black box highlights the 'Folders' row in the grid.

Plugins installation under progress, shown in below screenshot.

The screenshot shows the Jenkins 'Getting Started' page. At the top, there's a header with tabs like 'Management Center' and 'Setup Wizard [Jenkins]'. Below the header, it says 'secure | 52.66.8.81:8080'. The main area has a title 'Getting Started' and a sub-section 'Installed Plugin'. A blue arrow points from the 'Installed Plugin' sub-section to the 'Credentials Binding' row in the grid. Another blue arrow points from the 'To be installed plugin' sub-section to the 'Email Extension' row in the grid. The grid is organized into four columns: 'Folders' (with 'Folders' checked), 'OWASP Markup Formatter' (with a checkmark), 'Build Timeout' (checked), 'Credentials Binding' (checked). The second column contains 'Timestamper' (checked), 'Workspace Cleanup' (checked), 'Ant' (checked), 'Gradle' (checked). The third column contains 'Pipeline' (checked), 'GitHub Branch Source' (checked), 'Pipeline: GitHub Groovy Libraries' (checked), 'Pipeline: Stage View' (checked). The fourth column contains 'Git' (checked), 'SSH Build Agents' (checked), 'Matrix Authorization Strategy' (checked), 'PAM Authentication' (checked). The bottom right corner of the grid has a note '** - required dependency'. A black box highlights the 'Email Extension' row in the grid. A grey box highlights the 'Credentials Binding' row in the grid.

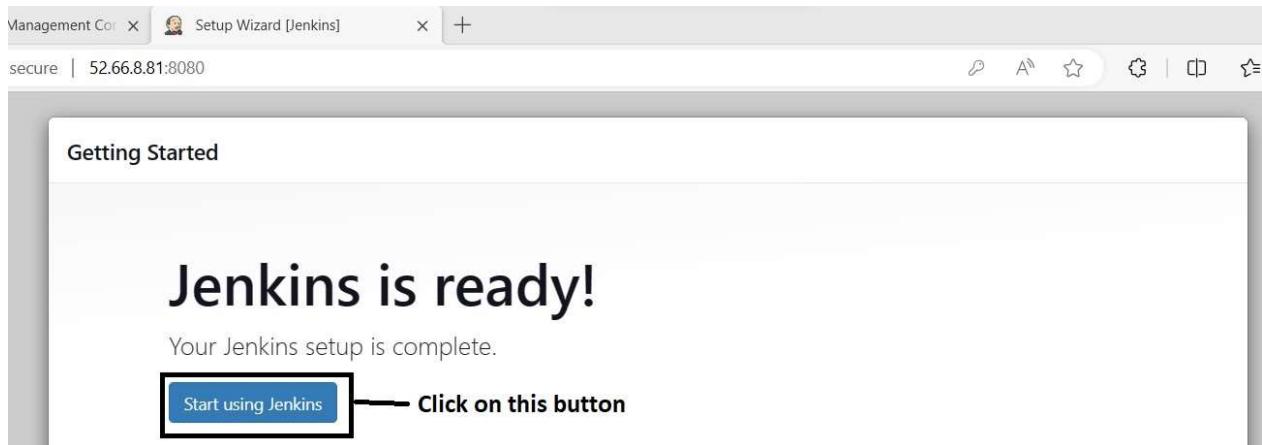
After successful installation of plugins, the following page will appear.

The screenshot shows the Jenkins Setup Wizard at step 1, titled 'Create First Admin User'. It contains five input fields: 'Username', 'Password', 'Confirm password', 'Full name', and 'E-mail address'. Below the fields is a note: 'Jenkins 2.401.3'. At the bottom right are two buttons: 'Skip and continue as admin' and a blue 'Save and Continue' button. A callout points to the 'Save and Continue' button with the text 'Click on this button'. Another callout points to all the input fields with the text 'All details to be filled'.

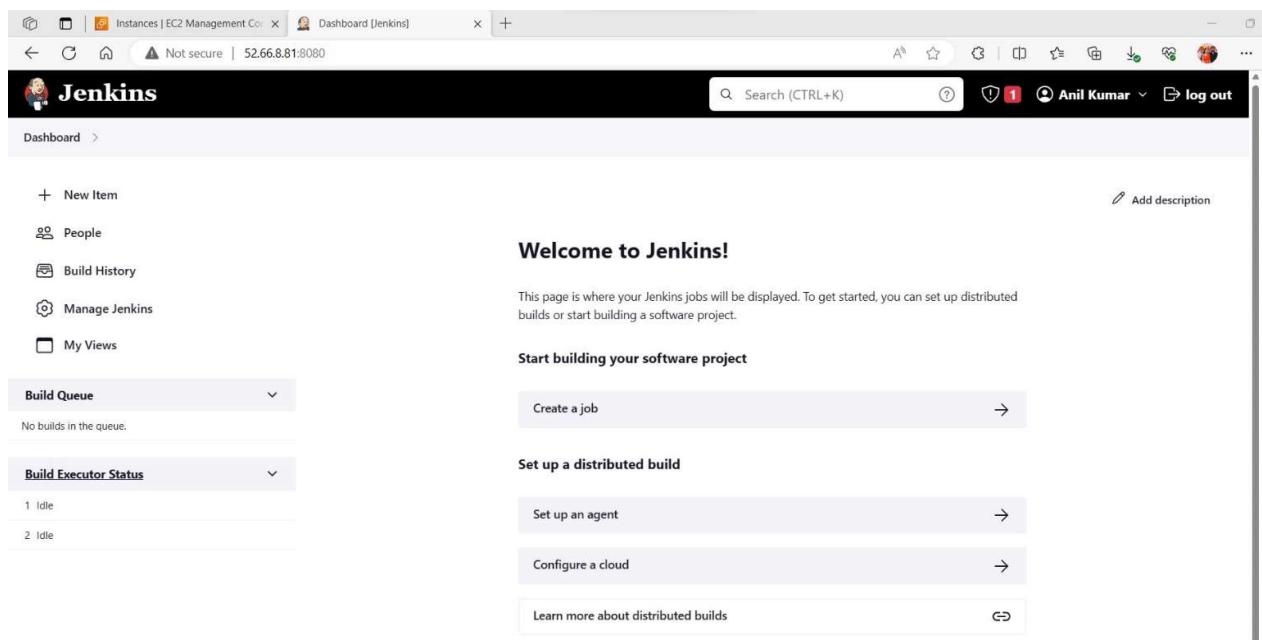
After, clicking on ‘Save and Continue’ button, the following page will appear.

The screenshot shows the Jenkins Setup Wizard at step 2, titled 'Instance Configuration'. It has a single input field for 'Jenkins URL' containing 'http://52.66.8.81:8080/'. Below the field is a detailed description of what the Jenkins URL is used for. At the bottom right are two buttons: 'Not now' and a blue 'Save and Finish' button. A callout points to the 'Save and Finish' button with the text 'Click on this button'.

After clicking on ‘Save and Finish’ button, the following page will appear.



After clicking on ‘Save and Finish’ button the following page will appear.



Finally, jenkins setup was completed successfully.

Now, it's time to setup the **Tomcat server**.

Step by Step Procedure For Tomcat Server Setup.

First execute the following command in terminal.

sudo apt-get update

After, executing the above command the following page will appear.

```
ubuntu@TomcatServer:~$ sudo apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [873 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [15.6 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [965 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [21.5 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [41.6 kB]
Fetched 2254 kB in 1s (2028 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

After, updating apt repository execute the following commands to install the Tomcat server.

sudo apt-get install tomcat9 tomcat9-admin -y

After, executing the above command tomcat9 will be installed.

To view the tomcat server status, execute the following commands in terminal.

sudo systemctl status tomcat9

After executing the above commands, the following screen will appear.

```
ubuntu@TomcatServer:~$ sudo systemctl status tomcat9
● tomcat9.service - Apache Tomcat 9 Web Application Server
   Loaded: loaded (/lib/systemd/system/tomcat9.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-08-11 03:27:27 UTC; 4min 43s ago
     Docs: https://tomcat.apache.org/tomcat-9.0-doc/index.html
 Process: 7220 ExecStartPre=/usr/libexec/tomcat9/tomcat-update-policy.sh (code=exited, status=0/SUCCESS)
 Main PID: 7231 (java)
    Tasks: 28 (limit: 1141)
   Memory: 120.5M
      CPU: 7.326s
     CGroup: /system.slice/tomcat9.service
             └─7231 /usr/lib/jvm/java-11-openjdk-amd64/bin/java -Djava.util.logging.config.file=/var/lib/tomcat9/conf/log

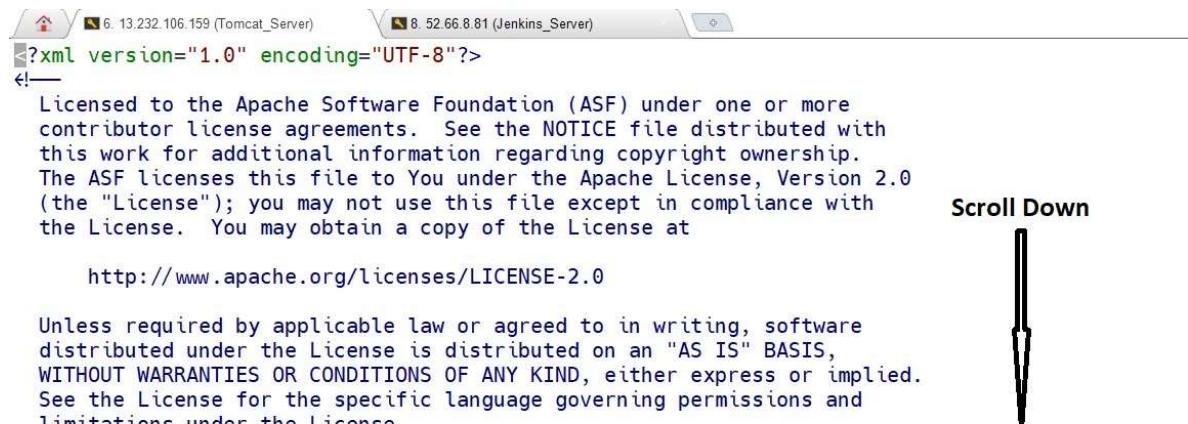
Aug 11 03:27:33 TomcatServer tomcat9[7231]: Deployment of deployment descriptor [/etc/tomcat9/Catalina/localhost/host-man
Aug 11 03:27:33 TomcatServer tomcat9[7231]: Deploying deployment descriptor [/etc/tomcat9/Catalina/localhost/manager.xml]
Aug 11 03:27:33 TomcatServer tomcat9[7231]: The path attribute with value [/manager] in deployment descriptor [/etc/tomca
Aug 11 03:27:34 TomcatServer tomcat9[7231]: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug log
Aug 11 03:27:34 TomcatServer tomcat9[7231]: Deployment of deployment descriptor [/etc/tomcat9/Catalina/localhost/manager.
Aug 11 03:27:34 TomcatServer tomcat9[7231]: Deploying web application directory [/var/lib/tomcat9/webapps/ROOT]
Aug 11 03:27:35 TomcatServer tomcat9[7231]: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug log
Aug 11 03:27:36 TomcatServer tomcat9[7231]: Deployment of web application directory [/var/lib/tomcat9/webapps/ROOT] has f
Aug 11 03:27:36 TomcatServer tomcat9[7231]: Starting ProtocolHandler ["http-nio-8080"]
Aug 11 03:27:36 TomcatServer tomcat9[7231]: Server startup in [4796] milliseconds
lines 1-22/22 (END)
```

Tomcat server installation completed successfully.

Now, we have to execute the following commands to configure the tomcat users.xml file.

sudo vi /var/lib/tomcat9/conf/tomcat-users.xml

After executing the above command, the following screen will appear.



```
?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<tomcat-users xmlns="http://tomcat.apache.org/xml"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
               version="1.0">
<!--
By default, no user is included in the "manager-gui" role required
to operate the "/manager/html" web application. If you wish to use this app,
you must define such a user - the username and password are arbitrary.
-->
```

After scrolling down, the following mentioned content will appear.

```
<!--
<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"/>
-->
We have to paste the content here, before </tomcat-users> tag
</tomcat-users>
```

Paste the following content in tomcat-users.xml file, above the </tomcat-users> tag image shown in next page.

```
<role rolename="admin-gui"/>
<role rolename="admin-script"/>
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<role rolename="manager-status"/>
<user username="tomcat" password="tomcat" roles="manager-script,admin-
gui,admin-script,manager-gui,manager-status"/>
```

After, pasting the content. Save the file. Based on your editor commands will change.
Ex: For VI it is :wq to save the file.

After pasting the content, the page looks like as mentioned below.

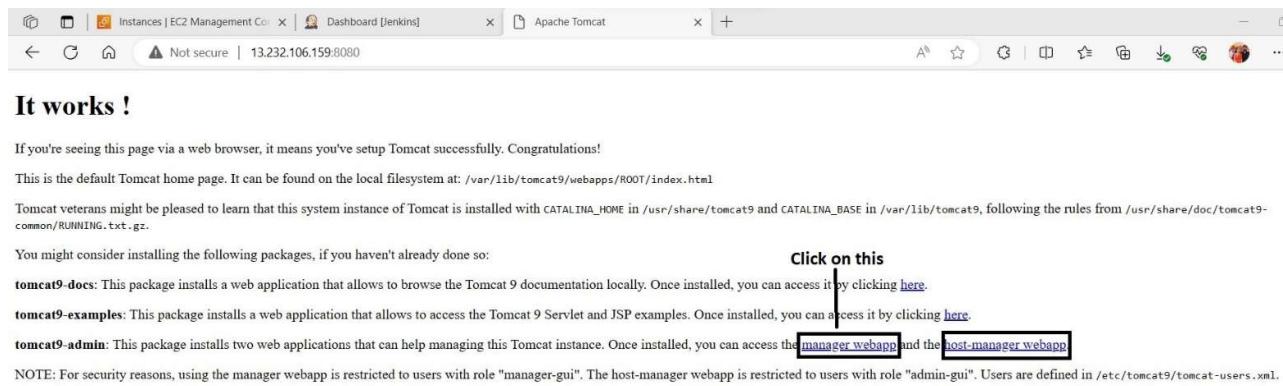
```
<user username="tomcat" password="<must-be-changed>" roles="tomcat,role1"/>
<user username="role1" password="<must-be-changed>" roles="role1"/>
→
<role rolename="admin-gui"/>
<role rolename="admin-script"/>
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<role rolename="manager-status"/>
<user username="tomcat" password="tomcat" roles="manager-script,admin-gui,admin-script,manager-gui,manager-status"/>
</tomcat-users>
```

Content pasted

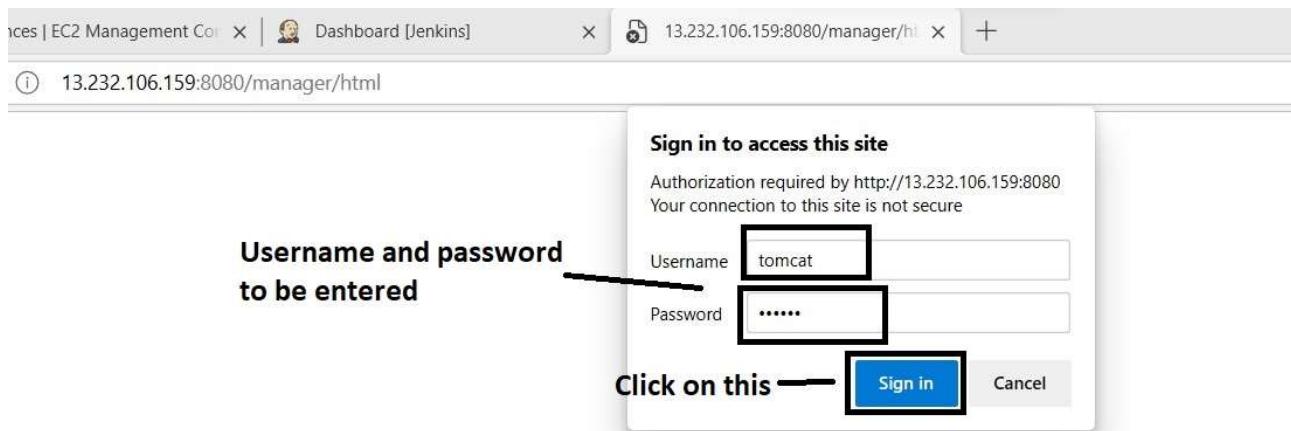
Now, enter the following URL in your browser.

http://YOUR IP:8080/

After entering the URL in browser, the following page will appear.



After clicking on 'manager webapp' link, the following page will appear.



After entering username and password, clicking on ‘sign in’ button the following page will appear.

The screenshot shows the Tomcat Web Application Manager interface. At the top, there's a header with tabs for 'Instances | EC2 Management' and 'Dashboard [Jenkins]'. Below the header, the URL is 13.232.106.159:8080/manager/html. On the left, there's a cartoon cat icon. To the right is the Apache logo. The main content area has a title 'Tomcat Web Application Manager'. It includes a 'Message:' field with 'OK' and a 'Manager' section with tabs for 'List Applications', 'HTML Manager Help', 'Manager Help', and 'Server Status'. A large table titled 'Applications' lists three entries:

Path	Version	Display Name	Running	Sessions	Commands
/	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	1	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

Below the table is a 'Deploy' section with fields for 'Context Path', 'Version (for parallel deployment)', 'XML Configuration file path', 'WAR or Directory path', and a 'Deploy' button.

In this page, we can see our deployed applications.
Tomcat is ready to deploy our web applications.

Now, we have to integrate the jenkins and tomcat for deploying our web application. Which is built by maven.

Step by step process for Integrating jenkins and Tomcat server

As already we had installed jenkins, open the jenkins dashboard by using following URL
http://YOUR IP :8080 After entering that URL, the following page appears.

The screenshot shows the Jenkins dashboard. The URL in the browser is 52.66.8.81:8080. The page features a 'Jenkins' logo and a search bar. On the left, there's a sidebar with links: '+ New Item', 'People', 'Build History', 'Manage Jenkins' (which is highlighted with a red box and a callout 'Click on'), and 'My Views'. In the center, it says 'Welcome to Jenkins!' and 'Start building your software project'. Below that, it says 'This page is where your Jenkins jobs will be displayed or start building a software project.' At the bottom, there's a 'Build Queue' dropdown and the text 'Page 12 of 30'.

After clicking on ‘Manage Jenkins’ button, the following page will appear.

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', 'Manage Jenkins' (which is highlighted), and 'My Views'. Below this are sections for 'Build Queue' (empty) and 'Build Executor Status' (2 Idle). The main area is titled 'Manage Jenkins' with a sub-section 'System Configuration'. It includes links for 'System' (global settings), 'Tools' (configure tools), and 'Nodes and Clouds' (control various nodes). A prominent 'Plugins' link is highlighted with a box and the instruction 'Click on'. At the top right, there are buttons for 'Set up agent', 'Set up cloud', and 'Dismiss'.

After clicking on ‘Plugins’ button, the following page will appear.

The screenshot shows the Jenkins Plugins page. The left sidebar has tabs for 'Updates' (selected), 'Available plugins' (highlighted with a box and 'Click on'), 'Installed plugins', 'Advanced settings', and 'Download progress'. The main content area is titled 'Plugins' with a search bar. A table lists plugin updates, with columns for 'Name' (sorted by release date) and 'Released'. At the bottom, a message says 'Update information obtained: 4 hr 19 min ago' with a 'Check now' button.

After clicking on ‘Available Plugins’ button the following page will appear, then we have to search for deploy to container plugin and it have to be installed.

The screenshot shows the Jenkins 'Available plugins' page. On the left sidebar, 'Available plugins' is selected. In the main area, a search bar contains the text 'deploy to container'. Below the search bar, the 'Deploy to container 1.16' plugin is listed under 'Artifact Uploaders'. A callout box points to the search bar with the text 'Search for deploy to container'. Another callout box points to the checkbox in the plugin listing with the text 'Select the check Box'. A third callout box points to the 'Install without restart' button with the text 'Click on'. The 'Install without restart' button is highlighted with a black border.

Note:- Deploy to container plugin is required to integrate jenkins and tomcat.

After clicking on ‘Install without restart’ button, the following page will appear.

The screenshot shows the Jenkins 'Download progress' page. The 'Dashboard' link in the top navigation bar is highlighted with a black box and a callout arrow pointing to it with the text 'Click on'. The 'Available plugins' link in the sidebar is also highlighted with a black box and a callout arrow pointing to it with the text 'Click on'. In the main area, the 'Download progress' section is titled 'Download progress'. It shows a list of plugins under 'Preparation' and 'Ionicons API', 'Folders', 'OWASP Markup Formatter', 'bouncycastle API', 'Instance Identity', 'JavaBeans Activation Framework (JAF) API', and 'javaMail API'. Each item has a green checkmark icon next to it, indicating success. To the right of the list, a vertical scroll bar is shown with a downward-pointing arrow, and a callout box points to it with the text 'Scroll down'. Another callout box points to the status column with the text 'We can see the status of plugin installation'.

At the bottom, we can see plugin installation status.

After clicking on ‘**Dashboard**’ option, the following page will appear. To build any project initially we require maven tool, to configure it click on ‘**Manage Jenkins**’ button.

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links: '+ New Item', 'People', 'Build History', 'Manage Jenkins' (which is highlighted with a black box and has a 'Click on' arrow pointing to it), and 'My Views'. On the right, there's a main area with the heading 'Welcome to Jenkins'. It says 'This page is where you build or start building' and 'Start building your first job'. At the bottom right is a 'Create a job' button.

After clicking on ‘**Manage Jenkins**’ button, the following page will appear.

The screenshot shows the 'Manage Jenkins' page. On the left, there's a sidebar with links: '+ New Item', 'People', 'Build History', 'Manage Jenkins' (which is highlighted with a black box and has a 'Click on' arrow pointing to it), and 'My Views'. Below that is a 'Build Queue' section showing 'No builds in the queue.' On the right, there's a 'System Configuration' section with three items: 'System' (Configure global settings and paths.), 'Tools' (Configure tools, their locations and automatic installers. - this is highlighted with a black box and has a 'Click on' arrow pointing to it.), and 'Nodes and Clouds' (Add, remove, control and monitor the various nodes that Jenkins runs jobs on.).

After clicking on ‘Tools’ button, the following page will appear.

A screenshot of a web browser showing the Jenkins 'Tools' configuration page. The URL is 52.66.8.81:8080/manage/configureTools/. The page has a header with tabs for Instances | EC2 Management Consoles and Tools [Jenkins]. Below the header, there's a navigation bar with Dashboard, Manage Jenkins, and Tools. The main content area is titled 'Tools' and contains sections for Maven Configuration, Default settings provider, and Default global settings provider. A vertical arrow points downwards from the top of the page towards the bottom, with the text 'Scroll down to find Maven Installations option' positioned next to it.

After scrolling down, you can find ‘Maven Installation’ option as shown in below screenshot.

A screenshot of the 'Maven' configuration screen. The title is 'Maven' and the section is 'Maven installations'. It shows a list of Maven installations on the system. A button labeled 'Add Maven' is highlighted with a black box and a callout arrow pointing to it, with the text 'Click on' next to it. At the bottom are 'Save' and 'Apply' buttons.

After clicking on ‘Add Maven’ button, the following screen will appear.

A screenshot of the 'Maven' configuration screen showing the 'Add Maven' dialog. The 'Name' field is filled with 'maven 3.9.4' and is highlighted with a black box and a callout arrow pointing to it, with the text 'You can give, your own name to identify the maven' next to it. The 'Install automatically' checkbox is checked. Below it, the 'Install from Apache' section is expanded, showing the 'Version' field with '3.9.4' selected and highlighted with a black box and a callout arrow pointing to it, with the text 'You can select, supported version.' next to it. At the bottom are 'Add Maven', 'Save', and 'Apply' buttons.

After clicking on ‘Save’ button, the following page will appear.

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links: '+ New Item', 'Click on People', 'Build History', 'Manage Jenkins' (which is highlighted), and 'My Views'. Below these are sections for 'Build Queue' (empty) and 'Build Executor Status' (1 Idle, 2 Idle). The main area is titled 'Manage Jenkins' and contains a message about building on a built-in node. It has two main sections: 'System Configuration' (with 'System' and 'Tools' sub-links) and 'Nodes and Clouds' (with a cloud icon and a link to add nodes). A search bar at the top right says 'Search (CTRL+K)'.

After clicking on ‘New Item’ button, the following page will appear.

The screenshot shows the Jenkins 'New Item' creation page. The title bar says 'New Item [Jenkins]'. The main form has a field 'Enter an item name' with 'Tomcat Deployment' entered. Below it is a 'Required field' note. A 'Select' arrow points to the 'Freestyle project' option, which is highlighted with a box. Other options shown are 'Pipeline' (with a description of orchestrating long-running activities), 'Multi-configuration project' (for testing on multiple environments), and 'Folder' (for grouping items). At the bottom, an 'OK' button is highlighted with a box, and a 'Multibranch Pipeline' link is visible.

After selecting ‘freestyle project’ and clicking on ‘ok’, the following page will appear.

The screenshot shows the Jenkins configuration interface for a 'Tomcat_Deployment' job. The 'General' tab is selected. A callout box points to the 'Source Code Management' section, which is highlighted with a black border. The 'Enabled' toggle switch is turned on. The 'Description' field is empty. Below it, there are several optional checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterized', 'Throttle builds', and 'Execute concurrent builds if necessary'. An 'Advanced' dropdown menu is visible at the bottom.

After clicking on ‘Source code Management’ button, the following page will appear.

The screenshot shows the 'Source Code Management' configuration page. The 'General' tab is selected. A callout box points to the 'Git' option, which is highlighted with a black border. The 'Build Triggers' and 'Build Environment' tabs are also visible.

After clicking on ‘Git’ button, the following page will appear.

The screenshot shows the Jenkins configuration interface for a Git repository. On the left, there's a sidebar with several tabs: General, Source Code Management (which is selected and highlighted in grey), Build Triggers, Build Environment (with a red box around it and the text 'Click on'), Build Steps, and Post-build Actions. The main panel is titled 'Configure' and has a 'Git' tab selected. It contains sections for 'Repositories', 'Branches to build', and an 'Advanced' dropdown. In the 'Repositories' section, there's a 'Repository URL' field containing 'https://github.com/mannem302/AnilKumar.git' with a red box around it and the text 'Paste your repository URL'. Below it is a 'Credentials' section with a dropdown set to '- none -' and a 'Add' button. In the 'Branches to build' section, there's a 'Branch Specifier (blank for 'any')' field containing '/main' with a red box around it and the text 'Based on your repository branch'. There's also an 'Add Repository' button.

You can see the branch details, in my repository.

The screenshot shows the GitHub repository details page for 'mannem302 / AnilKumar'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area shows the repository name 'AnilKumar' (Public). Below it, there's a summary showing 'main', '1 branch', and '0 tags' with a red box around the 'main' dropdown and the text 'Main Branch'. A list of files and their commit history is shown: 'mannem302 Update README.md' (cb677bb, 18 hours ago), 'src/main/webapp changes done', '.gitignore tomcat sample webapp created', 'README.md Update README.md', and 'pom.xml changes done'.

After clicking on ‘Build Environment’ option, the following page will appear.

The screenshot shows the Jenkins configuration interface for a job named 'Tomcat_Deployment'. The top navigation bar includes tabs for 'Instances | EC2 Management Config', 'Tomcat_Deployment Config [Jen]', and '/manager'. Below the navigation is a breadcrumb trail: 'Dashboard > Tomcat_Deployment > Configuration'. On the left, a sidebar lists 'General', 'Source Code Management', 'Build Triggers', 'Build Environment' (which is selected and highlighted in grey), 'Build Steps', and 'Post-build Actions'. The main content area is titled 'Configure' and contains several checkboxes for build options: 'Delete workspace before build starts', 'Use secret text(s) or file(s)', 'Add timestamps to the Console Output', 'Inspect build log for published build scans', 'Terminate a build if it's stuck', and 'With Ant'. Below these is a section titled 'Build Steps' with a button labeled 'Add build step ▾'. A callout arrow points to this button with the text 'Click on dropdown box'. Further down is a 'Post-build Actions' section with a similar 'Add post-build action ▾' button. At the bottom are 'Save' and 'Apply' buttons.

After clicking on ‘Add build step’ dropdown box the following page will appear.

This screenshot shows the expanded 'Add build step' dropdown menu from the previous screen. The menu is titled 'Build Steps' and includes a 'Filter' input field. Below the filter are several options: 'Execute Windows batch command', 'Execute shell', 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets' (which is highlighted with a black box and has a callout arrow pointing to it with the text 'Select this option'), 'Run with timeout', and 'Set build status to "pending" on GitHub commit'. The 'Invoke top-level Maven targets' option is the one intended to be selected for this task.

After selecting ‘**Invoke top-level Maven targets**’ options, the following page will appear.

Build Steps

The screenshot shows the 'Invoke top-level Maven targets' configuration page. It includes fields for 'Maven Version' (set to '(Default)'), 'Goals' (an empty input field), and an 'Advanced' dropdown menu. A callout points to the dropdown button with the instruction 'Click on dropdown button'. At the bottom is a 'Add build step ▾' button.

After clicking on dropdown button, the following page will appear.

Build Steps

The screenshot shows the same configuration page after clicking the dropdown button. The 'Maven Version' dropdown now has a selected item: 'maven 3.9.4'. A callout points to this option with the instruction 'Select this option'. The rest of the interface remains the same, including the 'Goals' field, 'Advanced' dropdown, and 'Add build step ▾' button.

After selecting ‘maven 3.9.4’ option, the following page will appear.

Configure

Maven Version
maven 3.9.4

Goals
clean package

Post-build Actions

Add post-build action ▾

Save Apply

maven goals ‘clean package’.

After clicking on ‘Post-build Actions’ button, the following page will appear.

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Select this option

Deploy war/ear to a container

Aggregate downstream test results

Archive the artifacts

Build other projects

Publish JUnit test result report

Record fingerprints of files to track usage

Git Publisher

E-mail Notification

Editable Email Notification

Set GitHub commit status (universal)

Set build status on GitHub commit [deprecated]

Delete workspace when build is done

Add post-build action ▾

Save Apply

After selecting ‘Deploy war/ear to a container’ option, the following page will appear.

Configure

Post-build Actions

WAR/EAR files ? ****/*.war**

Context path ?

If context.xml file is not available in github, you have to mention here. Otherwise not required.

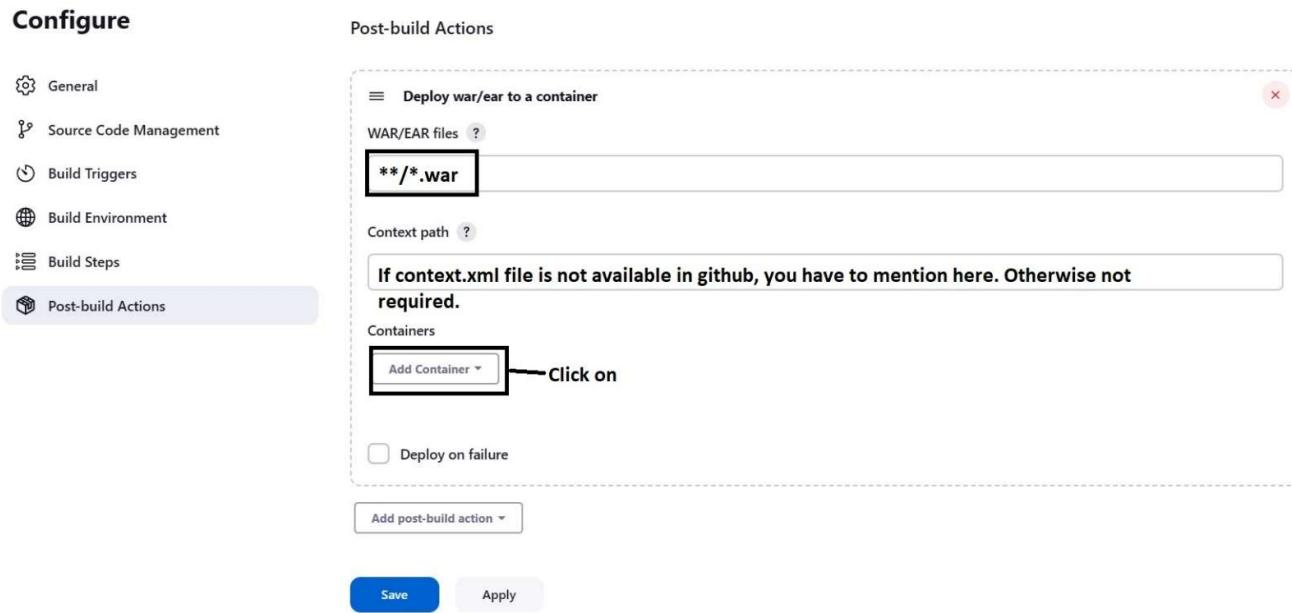
Containers

Add Container ▾ **Click on**

Deploy on failure

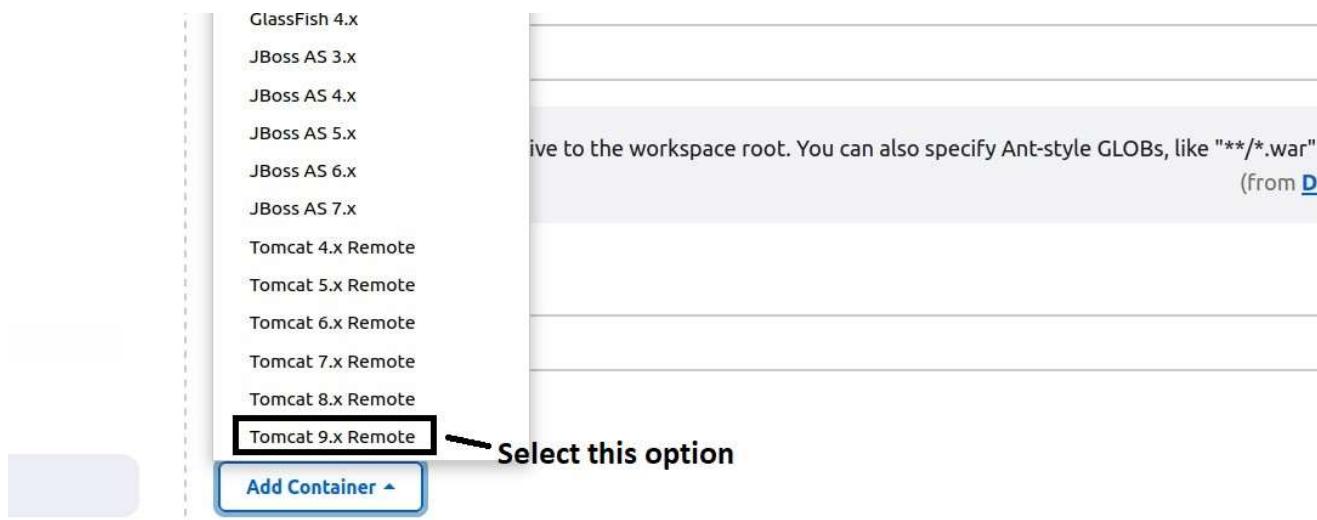
Add post-build action ▾

Save Apply



After clicking on ‘Add Container’ button, the following page will appear. Based on your tomcat version, you have to select the option.

GlassFish 4.x
JBoss AS 3.x
JBoss AS 4.x
JBoss AS 5.x
JBoss AS 6.x
JBoss AS 7.x
Tomcat 4.x Remote
Tomcat 5.x Remote
Tomcat 6.x Remote
Tomcat 7.x Remote
Tomcat 8.x Remote
Tomcat 9.x Remote **Select this option**
Add Container ▾



After selecting that ‘Tomcat 9.x Remote’ option, the following page will appear.

Containers

- Tomcat 9.x Remote

Credentials

- none -

Save Apply

↓ Scroll down

After scrolling down, the following page will appear.

Containers

- Tomcat 9.x Remote

Credentials

- none -

Add ▾ Click on

Tomcat URL ?

paste the tomcat server URL http://Your_IP:8080

Advanced ▾

Add Container ▾

Deploy on failure

Add post-build action ▾

Save Apply

After clicking on ‘Add’ dropdown button, the following page will appear.

Containers

- Tomcat 9.x Remote

Credentials

- none -

Add ▾ Click on

Jenkins

Tomcat URL ?

After clicking on ‘Jenkins’ option, the following page will appear.

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain: Global credentials (unrestricted)

Kind: Username with password

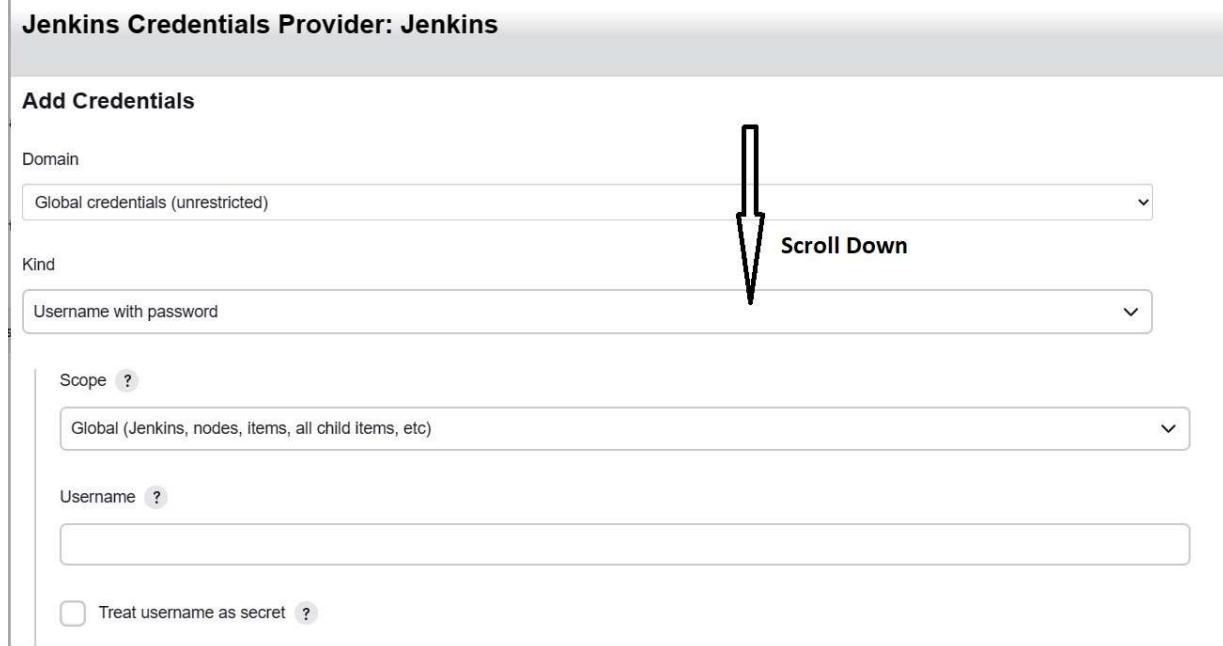
Scope: Global (Jenkins, nodes, items, all child items, etc)

Username:

Treat username as secret

▼

Scroll Down



After scrolling down, we can enter ‘username’ and ‘password’ as shown in below screenshot.

Global (Jenkins, nodes, items, all child items, etc)

Username:

Treat username as secret

Password:

ID:

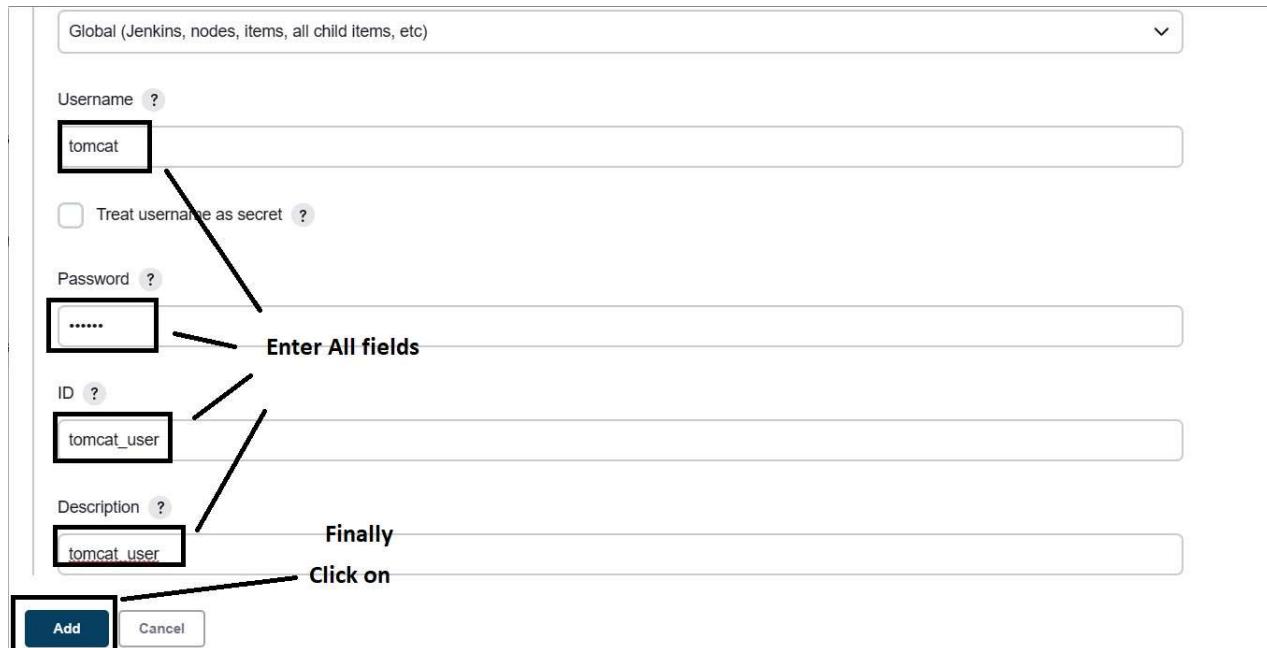
Description:

Enter All fields

Finally

Click on

Add **Cancel**



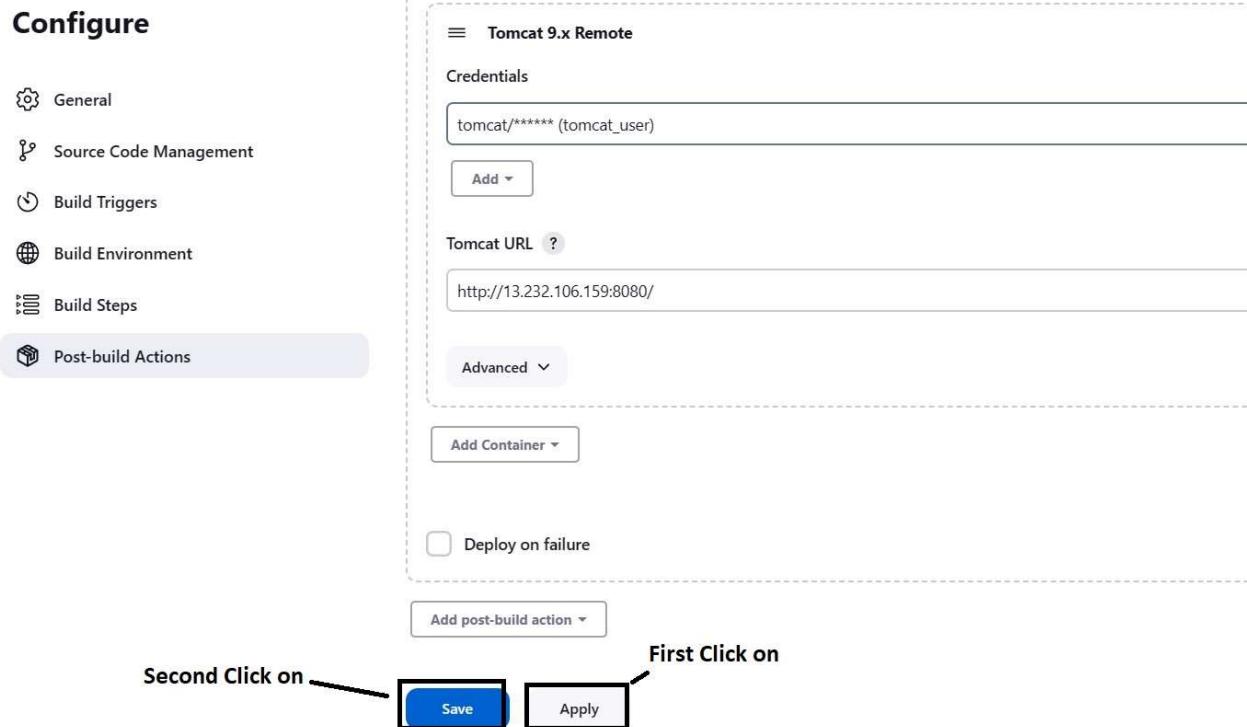
After clicking on ‘Add’ button, the following page will appear.

The screenshot shows a user interface for adding a container. At the top left is a 'Containers' section. Below it is a dashed box containing a 'Tomcat 9.x Remote' configuration. Inside this box, under the 'Credentials' heading, there is a dropdown menu with the option '- none -' selected. To the right of the dropdown is a button labeled 'Add ▾'. A handwritten-style arrow points from the text 'Click on dropdown option' to the 'Add ▾' button.

After clicking on dropdown option, the following page will appear.

This screenshot shows the same configuration interface after the 'Add ▾' button was clicked. The dropdown menu is now open, displaying three options: '- none -', 'tomcat/***** (tomcat_user)', and 'tomcat/***** (tomcat_root)'. The option 'tomcat/***** (tomcat_user)' is highlighted with a black rectangular box. A handwritten-style arrow points from the text 'Select this user' to this highlighted option.

After selecting that user, the following option will appear.



After clicking on ‘Apply’ and ‘Save’ buttons the following page will appear.

The screenshot shows the Jenkins Project Tomcat_Deployment dashboard. At the top, there's a navigation bar with tabs for Instances | EC2 Management, Tomcat_Deployment [Jenk..., /manager, and AnilKumar/src/main/web. Below the navigation bar is the Jenkins logo and a search icon. The main content area has a dark header with the project name 'Project Tomcat_Deployment'. On the left, there's a sidebar with the following items:

- Status (highlighted with a grey background)
- </> Changes
- Workspace
- Build Now (highlighted with a black border and an arrow pointing to it with the text 'Click on')
- Configure
- Delete Project
- Rename

On the right side, there's a 'Permalinks' section and a 'Build History' section. The 'Build History' section includes a 'trend' dropdown, a 'Filter builds...' input field, and a message 'No builds'. At the bottom, there are links for 'Atom feed for all' and 'Atom feed for failures'.

After clicking on ‘Build Now’ button, the project deployment process starts as shown in below screenshots.

The screenshot shows the Jenkins interface for the 'Project Tomcat_Deployment' page. On the left, there is a sidebar with options: Status, Changes, Workspace, Build Now (which is highlighted with a yellow background), Configure, Delete Project, and Rename. Below the sidebar is a 'Build History' section. It includes a 'trend' dropdown, a 'Filter builds...' input field, and a search icon. A specific build entry for '#2 Aug 11, 2023, 6:16 AM' is highlighted with a black rectangle. A large arrow points from the text 'Click on' to this build entry. At the bottom of the build history section are links for 'Atom feed for all' and 'Atom feed for failures'.

After clicking on that build, the following page will appear.

The screenshot shows the Jenkins interface for the 'Build #2 (Aug 11, 2023, 6:16:50 AM)' page. The top navigation bar shows 'Dashboard > Tomcat_Deployment > #2'. The left sidebar has options: Status, Changes, Console Output (which is highlighted with a yellow background), Edit Build Information, Delete build '#2', and Git Build Data. The main content area displays a green checkmark icon followed by the text 'Build #2 (Aug 11, 2023, 6:16:50 AM)'. It shows that there were 'No changes.' and it was 'Started by user Anil Kumar'. Below that, it shows the 'Revision' and 'Repository' information: cb677bbeb4b8bd84a67ed409bd7f1ad5f6036d79 and <https://github.com/mannem302/AnilKumar.git>. A large arrow points from the text 'Click on' to the 'Console Output' link in the sidebar.

To view the deployment status, click on ‘Console Output’ button. After clicking on that button, the following page will appear.

The screenshot shows the Jenkins interface for a Tomcat Deployment job. The top navigation bar includes links for Status, Changes, and Console Output, with 'Console Output' being the active tab. A search bar and user information for 'Anil Kumar' are also present. The main content area is titled 'Console Output' with a green checkmark icon. It displays deployment logs from Maven repositories, including progress messages for downloading various dependencies like maven-doxia-tools and commons-parent. A note on the right says 'Scroll down / It automatically displays the log while project is building.' with a downward arrow. The logs show the assembly of a webapp named 'AnilKumar' and the deployment of its WAR file to a Tomcat 9.x container. The final message indicates a 'Web Application successfully deployed on tomcat Server'.

After scrolling down, you can view the project status. As shown in below screenshot.

```
[INFO] Assembling webapp [AnilKumar] in [/var/lib/jenkins/workspace/Tomcat_Deployment/target/AnilKumar-1.0]
[INFO] Processing war project
[INFO] Copying webapp resources [/var/lib/jenkins/workspace/Tomcat_Deployment/src/main/webapp]
[INFO] Webapp assembled in [31 msecs]
[INFO] Building war: /var/lib/jenkins/workspace/Tomcat_Deployment/target/AnilKumar-1.0.war
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 10.620 s
[INFO] Finished at: 2023-08-11T06:17:09Z
[INFO] -----
[DeployPublisher][INFO] Attempting to deploy 1 war file(s)
[DeployPublisher][INFO] Deploying /var/lib/jenkins/workspace/Tomcat_Deployment/target/AnilKumar-1.0.war to container Tomcat 9.x Remote with context AnilKumar
[/var/lib/jenkins/workspace/Tomcat_Deployment/target/AnilKumar-1.0.war] is not deployed. Doing a fresh deployment.
Deploying [/var/lib/jenkins/workspace/Tomcat_Deployment/target/AnilKumar-1.0.war]
Finished: SUCCESS
————— Web Application successfully deployed on tomcat Server
```

To view the deployed web application, open the tomcat server URL in browser.

YOUR_IP:8080, tomcat page opens as shown in following screenshot.

The screenshot shows a browser window with the URL `13.232.106.159:8080/manager/html`. The title bar says "Tomcat Web Application Manager". The page features a cartoon cat logo at the top left. A message box says "Message: OK". Below it is a yellow header bar with "Manager" and links for "List Applications" and "HTML Manager Help". The main content is a table titled "Applications" with columns: Path, Version, Display Name, Running, and Sessions. There are four rows:

Path	Version	Display Name	Running	Sessions
/	None specified	Your Web Application	true	0
/AnilKumar	None specified		true	0
/host-manager	None specified	Tomcat Host Manager Application	true	0
/manager	None specified	Tomcat Manager Application	true	1

A red box highlights the path `/AnilKumar` in the first row, with an arrow pointing to it labeled "Click on".

After clicking on that '**AnilKumar**' web application, the following page will appear.

The screenshot shows a browser window with the URL `13.232.106.159:8080/AnilKumar/`. The title bar says "Tomcat Deployment #2". The page displays the message "You have successfully deployed, web-application in Tomcat Server" and "Created By Anil Kumar Mannem".