

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)**
- 3) SonarQube Server (t2.Medium – Instance3)**

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.



```
6. 13.232.106.159 (Tomcat_Server) 8. 52.66.8.81 (Jenkins_Server)
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
```

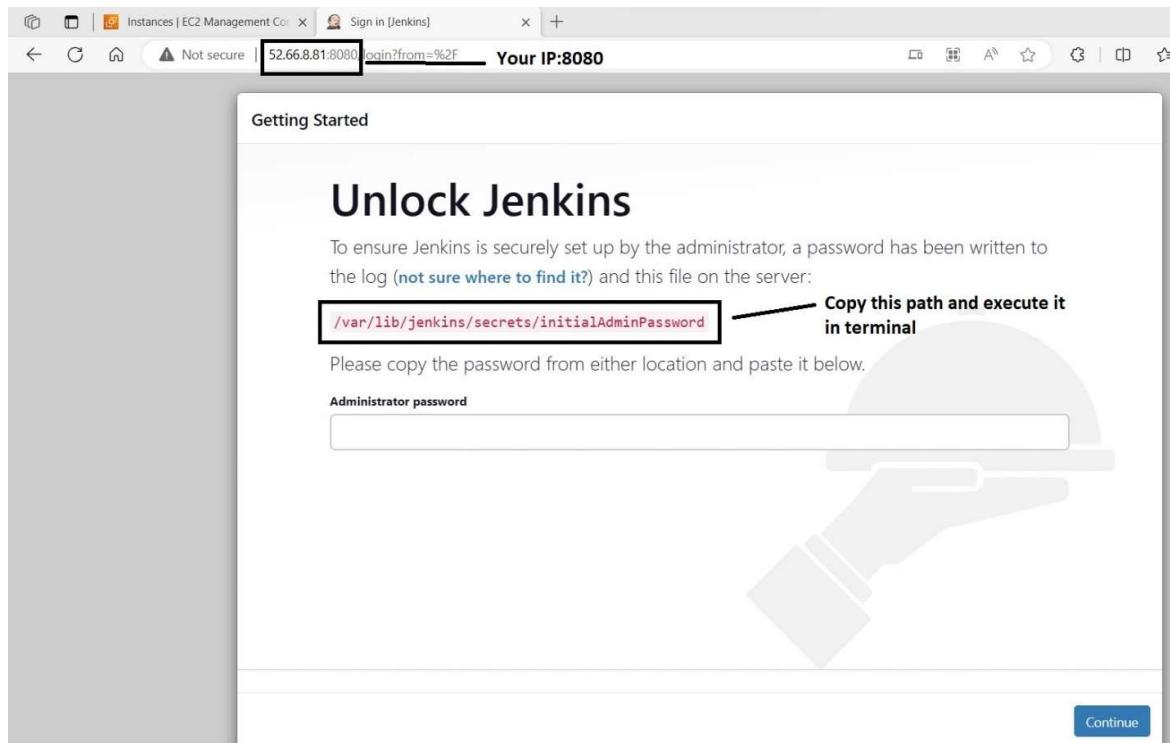
After executing the above commands, the following screen will appear.

```
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) Jenkins server is in running state  
     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.

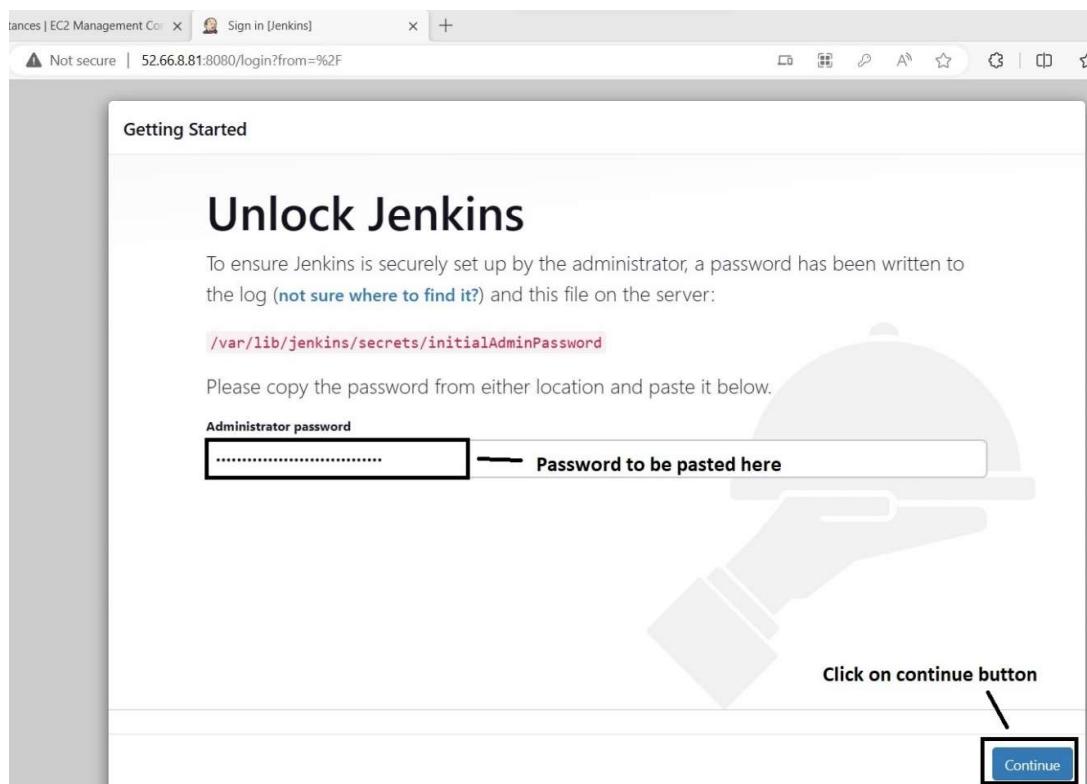
sudo cat /var/lib/jenkins/secrets/initialAdminPassword

The following output message will be displayed as shown below.

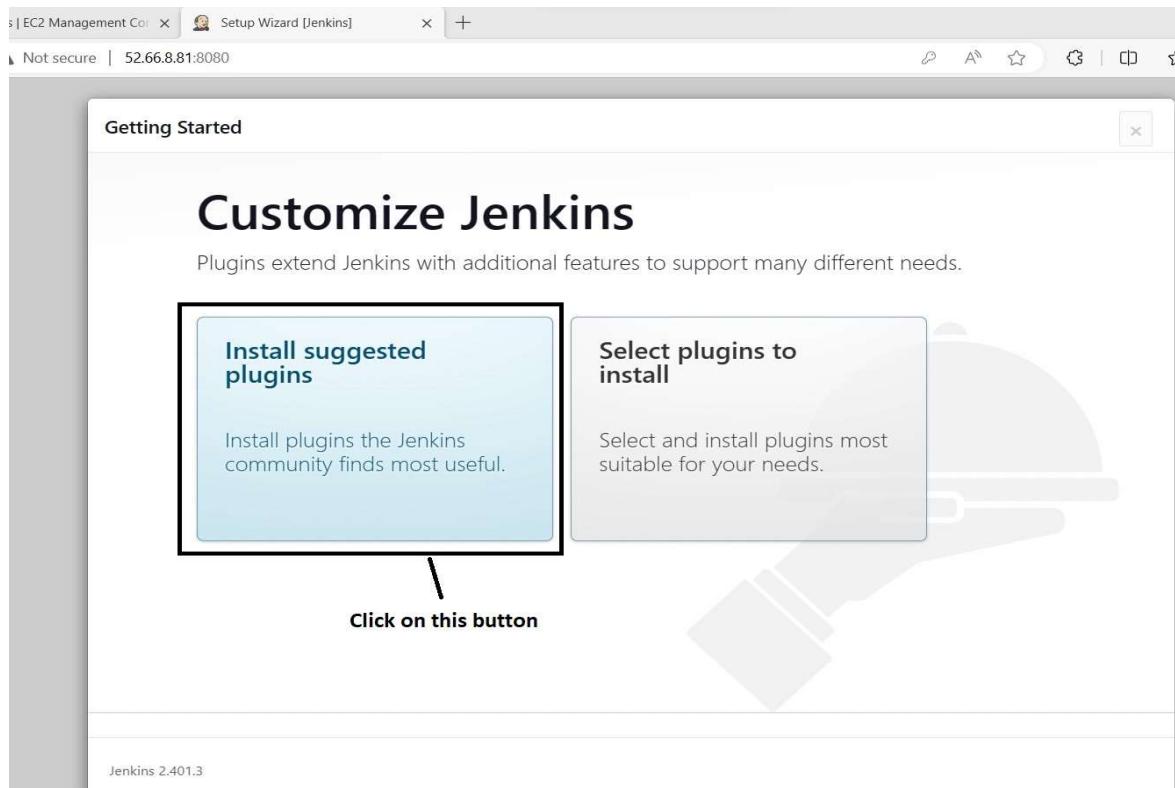
The terminal window shows the command 'sudo cat /var/lib/jenkins/secrets/initialAdminPassword' being run. The output is '005728519dce4300bff9905ce11d2a4e'. A red box highlights this output. An arrow points from the text 'Password to be copied and paste it in browser' to the highlighted password.

```
ubuntu@JenkinsServer:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
005728519dce4300bff9905ce11d2a4e
```

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	✓ OWASP Markup Formatter	○ Build Timeout	○ Credentials Binding
⌚ Timestamper	⌚ Workspace Cleanup	⌚ Ant	⌚ Gradle
⌚ Pipeline	⌚ GitHub Branch Source	⌚ Pipeline: GitHub Groovy Libraries	⌚ Pipeline: Stage View
○ Git	⌚ SSH Build Agents	⌚ Matrix Authorization Strategy	⌚ PAM Authentication
○ LDAP	⌚ Email Extension	⌚ Mailer	

Below the grid, there's a note '** - required dependency'. The entire grid is highlighted with a black rectangle.

Plugins installation under progress, shown in below screenshot.

This screenshot shows the same Jenkins 'Getting Started' page as the previous one, but with some changes. A blue bar at the top has the text 'Installed Plugin' with an arrow pointing to the 'Credentials Binding' row in the grid. Another arrow points to the 'Email Extension' row with the text 'To be installed plugin'. The grid structure is identical to the first screenshot, with the same columns and rows of plugin icons and names. The 'Credentials Binding' row is highlighted with a black rectangle, and the 'Email Extension' row is also highlighted with a black rectangle.

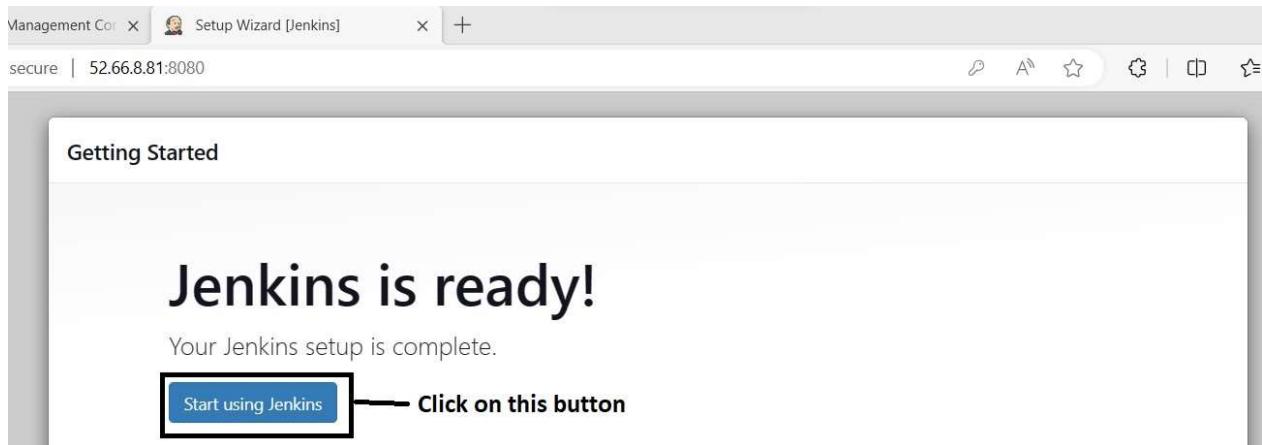
After successful installation of plugins, the following page will appear.

The screenshot shows the Jenkins Setup Wizard 'Create First Admin User' page. It has 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 all five input fields with the text 'All details to be filled'. Another callout points to the 'Save and Continue' button with the text 'Click on this button'.

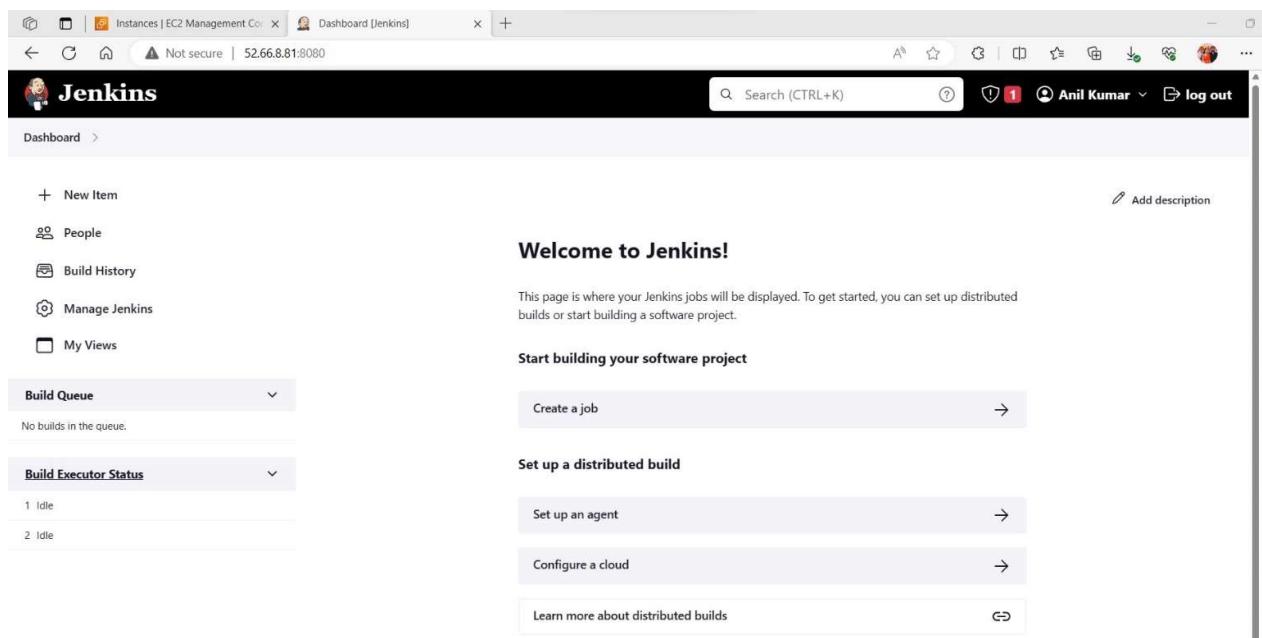
After, clicking on ‘Save and Continue’ button, the following page will appear.

The screenshot shows the Jenkins Setup Wizard 'Instance Configuration' page. It features 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

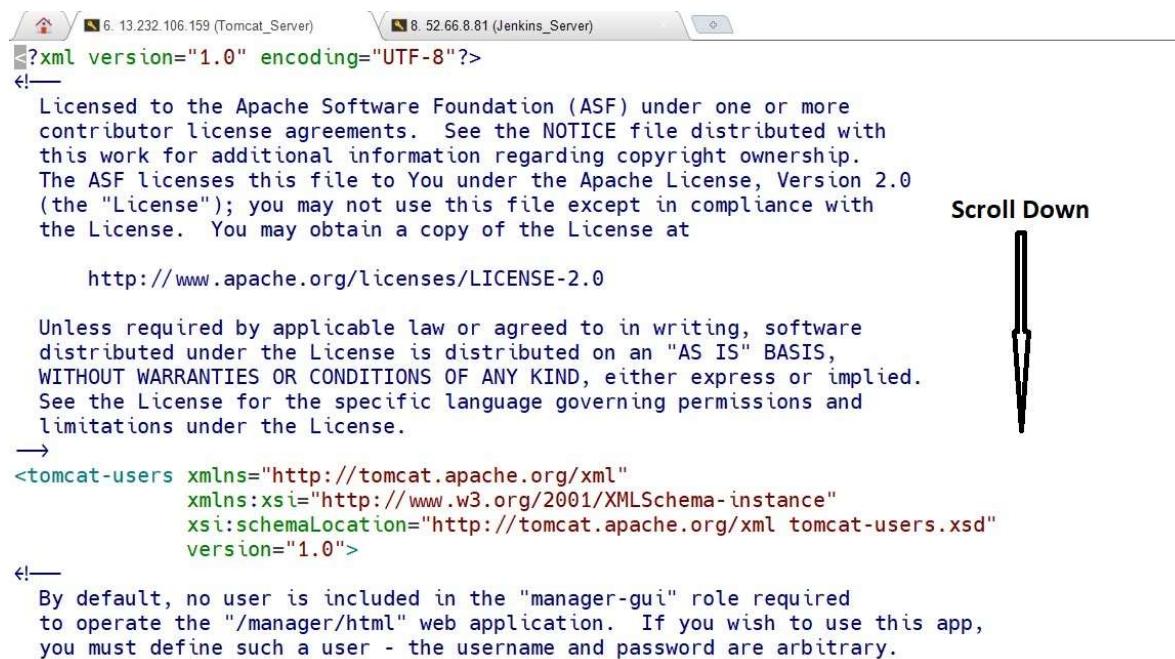
Aug 11 03:27:33 TomcatServer tomcat9[7231]: Deployment of deployment descriptor [/etc/tomcat9/Catalina/localhost/host-man
Aug 11 03:27:33 TomcatServer tomcat9[7231]: Deploying deployment descriptor [/etc/tomcat9/Catalina/localhost/manager.xml]
Aug 11 03:27:33 TomcatServer tomcat9[7231]: The path attribute 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.

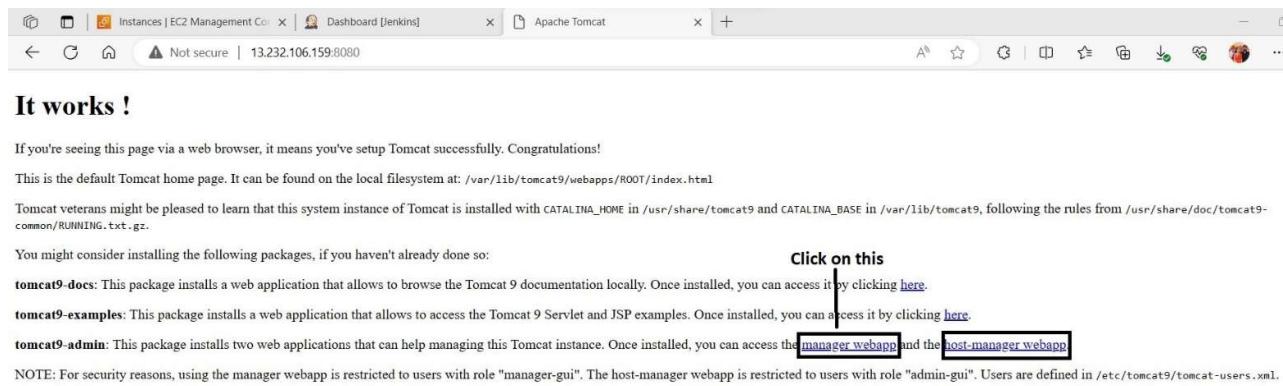
```
<!-- Usernames and Passwords should be changed before running tomcat -->
<user username="both" password="" roles="tomcat,role1"/>
<user username="role1" password="" roles="role1"/>
→
<role rolename="admin-gui"/>
<role rolename="admin-script"/>
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<role rolename="manager-status"/>
<user username="tomcat" password="tomcat" roles="manager-script,admin-gui,admin-script,manager-gui,manager-status"/>
</tomcat-users>
```

Content pasted

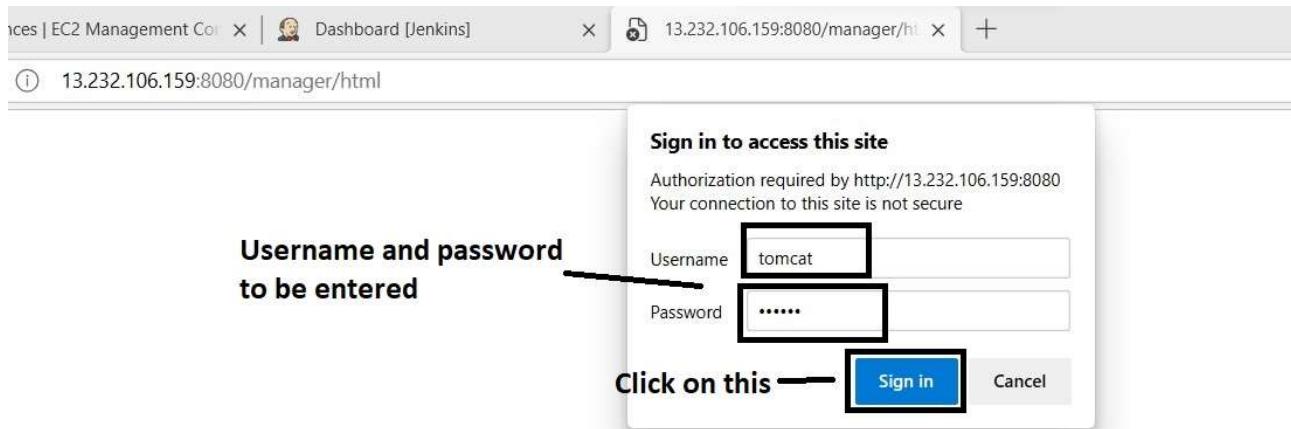
Now, enter the following URL in your browser.

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 Software Foundation logo. The main content area has a title 'Tomcat Web Application Manager'. A message box says 'Message: OK'. Below it is a 'Manager' section with tabs for 'List Applications', 'HTML Manager Help', 'Manager Help', and 'Server Status'. The 'List Applications' tab is selected, showing a table of deployed applications:

Applications					
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 application list 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. At the top, there's a header with tabs for 'Instances | EC2 Management' and 'Dashboard [Jenkins]'. Below the header, the URL is 52.66.8.81:8080. The main content area has a title 'Welcome to Jenkins!'. It says 'This page is where your Jenkins jobs will be displayed or start building a software project.' and 'Start building your software project'. 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'. At the bottom, there's a 'Build Queue' dropdown and a footer saying 'Page 12 of 59'.

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. On the left, there's a sidebar with 'Updates' (highlighted), 'Available plugins' (highlighted with a box and the instruction 'Click on'), 'Installed plugins', 'Advanced settings', and 'Download progress'. The main area is titled 'Plugins' and features 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' and 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 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 with a checked checkbox. A callout box points to this checkbox with the text 'Select the check Box'. At the bottom, there are two buttons: 'Install without restart' (highlighted with a box and a callout 'Click on') and 'Download now and install after restart'. A callout box points to the 'Install without restart' button with the text 'Click on'.

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 box and a callout 'Click on'. In the main area, the 'Download progress' section is shown. It includes a 'Preparation' list with items like 'Ionicons API', 'Folders', 'OWASP Markup Formatter', 'bouncycastle API', 'Instance Identity', 'JavaBeans Activation Framework (JAF) API', and 'javaMail API', each marked as 'Success'. To the right, a vertical scroll bar is shown with a downward arrow, and a callout box points to it with the text 'Scroll down'. Another callout box points to the 'Success' status 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' dropdown 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 page. 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 an 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 page showing the 'Add Maven' dialog. The 'Name' field is filled with 'maven 3.9.4' and is highlighted with a black box and an 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 an 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 a box containing 'Freestyle project' with its description. Other options shown are 'Pipeline', 'Multi-configuration project', and 'Folder'. At the bottom, an 'OK' button is highlighted with a 'Click on' arrow.

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. On the left, there's a sidebar with 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'. A callout arrow points to the 'Source Code Management' button with the text 'Click on'. The main area has a 'Description' field and several checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterized', 'Throttle builds', and 'Execute concurrent builds if necessary'. An 'Advanced' dropdown is also visible. At the top right, there's an 'Enabled' toggle switch which is turned on.

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. On the left, there's a sidebar with 'Source Code Management', 'Build Triggers', and 'Build Environment'. A callout arrow points to the 'Git' button with the text 'Click on'. The main area shows a radio button for 'None' which is selected, and another for 'Git' which is unselected. Below this is a 'Build Triggers' section.

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 various tabs: General, Source Code Management (which is selected and highlighted in grey), Build Triggers, Build Environment (which is highlighted with a black rectangle and has an arrow pointing to it labeled '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' input field containing 'https://github.com/mannem302/AnilKumar.git', with a note 'Paste your repository URL' next to it. In the 'Branches to build' section, there's a 'Branch Specifier (blank for 'any')' input field containing '/main', with a note 'Based on your repository branch' next to it. There are also 'Add Repository' and 'Advanced' buttons.

<https://github.com/mannem302/AnilKumar.git>

You can see the branch details, in my repository.

The screenshot shows the GitHub repository details for 'mannem302 / AnilKumar'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the header, the repository name 'AnilKumar' is shown as 'Public'. The main area displays the 'Main Branch' with a dropdown menu showing 'main'. It shows 1 branch and 0 tags. A list of files is shown: 'mannem302 Update README.md' (commit 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 'Tomcat_Deployment' job. The left sidebar has sections: General, Source Code Management, Build Triggers, **Build Environment** (which is selected and highlighted in grey), and Post-build Actions. The main content area is titled 'Configure'. Under 'Build Environment', there are several checkboxes: '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 this is a 'Build Steps' section 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 button labeled 'Add post-build action ▾'. At the bottom are 'Save' and 'Apply' buttons.

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

The screenshot shows the 'Add build step' dropdown menu from the previous screen. The menu is titled 'Add build step ▾'. It includes a 'Filter' input field and a list of options: 'Execute Windows batch command', 'Execute shell', 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets' (which is highlighted with a black rectangle and a callout arrow pointing to it with the text 'Select this option'), 'Run with timeout', and 'Set build status to "pending" on GitHub commit'.

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

Build Steps

Invoke top-level Maven targets ?

Maven Version

(Default)

Goals

Advanced ▾

Add build step ▾

A callout box with a red border and a black arrow points to the dropdown button next to the 'Maven Version' input field. The text 'Click on dropdown button' is written above the arrow.

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

Build Steps

Invoke top-level Maven targets ?

Maven Version

(Default)

(Default)
maven 3.9.4

Advanced ▾

Add build step ▾

A callout box with a red border and a black arrow points to the selected option 'maven 3.9.4' in the dropdown menu. The text 'Select this option' is written above the arrow.

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

The screenshot shows the 'Tomcat Deployment Configuration' page. On the left, a sidebar lists 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps' (which is selected), and 'Post-build Actions'. In the main area, under 'Invoke top-level Maven targets', the 'Maven Version' is set to 'maven 3.9.4'. The 'Goals' field contains 'clean package', which is highlighted with a black box and has a callout pointing to it with the text 'maven required goals to be entered'. Below this is an 'Advanced' dropdown and a 'Save' button. Under 'Post-build Actions', there is a 'Add post-build action' button, which is also highlighted with a black box and has a callout pointing to it with the text 'click on'. Below this are 'Save' and 'Apply' buttons.

maven goals ‘clean package’.

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

The screenshot shows the 'Post-build Actions' configuration page. The 'Build Steps' section is selected in the sidebar. A callout points to the 'Post-build Actions' button with the text 'Select this option'. A dropdown menu is open, showing various actions: 'Aggregate downstream test results', 'Archive the artifacts', 'Build other projects', 'Publish JUnit test result report', 'Record fingerprints of files to track usage', 'Git Publisher', 'Deploy war/ear to a container' (which is highlighted with a black box), 'E-mail Notification', 'Editable Email Notification', 'Set GitHub commit status (universal)', 'Set build status on GitHub commit [deprecated]', and 'Delete workspace when build is done'. Below the dropdown is an 'Add post-build action' button. At the bottom are 'Save' and 'Apply' buttons.

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

The screenshot shows the Jenkins 'Configure' screen with the 'Post-build Actions' tab selected. A sub-section titled 'Deploy war/ear to a container' is expanded. It includes fields for 'WAR/EAR files' containing '**/*.war' and 'Context path'. A note states: 'If context.xml file is not available in github, you have to mention here. Otherwise not required.' Below this is a 'Containers' section with a dropdown menu labeled 'Add Container ▾' which has an arrow pointing to it. There is also a checkbox for 'Deploy on failure' and a button to 'Add post-build action ▾'. At the bottom are 'Save' and 'Apply' buttons.

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

The screenshot shows a dropdown menu with various container options listed vertically. The options include: 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, and Tomcat 9.x Remote. The 'Tomcat 9.x Remote' option is highlighted with a black rectangle and has an arrow pointing to it with the text 'Select this option'. At the bottom of the dropdown is a blue 'Add Container ▾' button.

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

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 

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

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

Username ?
tomcat

Treat username as secret ?

Password ?
.....

ID ?
tomcat_user

Description ?
tomcat_user

Enter All fields

Finally

Click on

Add Cancel

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

The screenshot shows a configuration interface for a 'Tomcat 9.x Remote' container. On the left, there's a sidebar with 'Containers' and a tree view. The main area has a title 'Tomcat 9.x Remote'. Under 'Credentials', a dropdown menu is open, showing options like '- none -' and 'tomcat/***** (tomcat_user)'. A cursor arrow points to the 'tomcat/***** (tomcat_user)' option. A callout bubble says 'Click on dropdown option'.

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

The screenshot shows the same configuration interface after selecting 'tomcat/***** (tomcat_user)' from the dropdown. The dropdown is now closed, and the selected option is highlighted. A callout bubble says 'Select this user'.

After selecting that user, the following option will appear.

The screenshot shows the configuration interface with the user selected. On the left, a sidebar lists 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions' (which is currently selected). The main area shows the 'Tomcat 9.x Remote' configuration with the credential set to 'tomcat/***** (tomcat_user)', 'Tomcat URL' set to 'http://13.232.106.159:8080/', and an 'Advanced' section expanded. At the bottom, there are 'Save' and 'Apply' buttons, with a callout bubble pointing to the 'Save' button saying 'First Click on' and another callout bubble pointing to the 'Save' button saying 'Second Click on'.

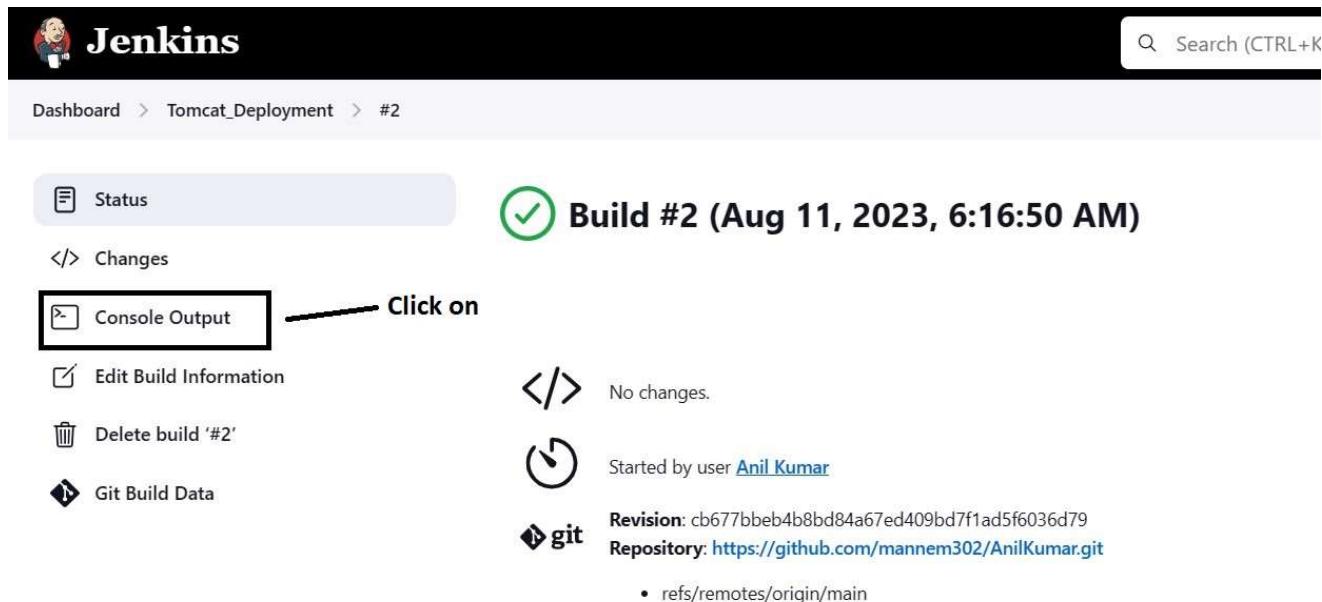
After clicking on ‘Apply’ and ‘Save’ buttons the following page will appear.

The screenshot shows the Jenkins interface for the 'Tomcat_Deployment' project. At the top, there are several browser tabs: 'Instances | EC2 Management', 'Tomcat_Deployment [Jenk]', '/manager', and 'AnilKumar/src/main/web'. Below the tabs, the Jenkins logo is displayed. The main title is 'Project Tomcat_Deployment'. On the left, a sidebar menu includes 'Status', 'Changes', 'Workspace', 'Build Now' (which is highlighted with a red box and a callout 'Click on'), 'Configure', 'Delete Project', and 'Rename'. The 'Build History' section shows a single build entry: '#1 Aug 11, 2023, 6:16 AM'. Below the build history, 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.

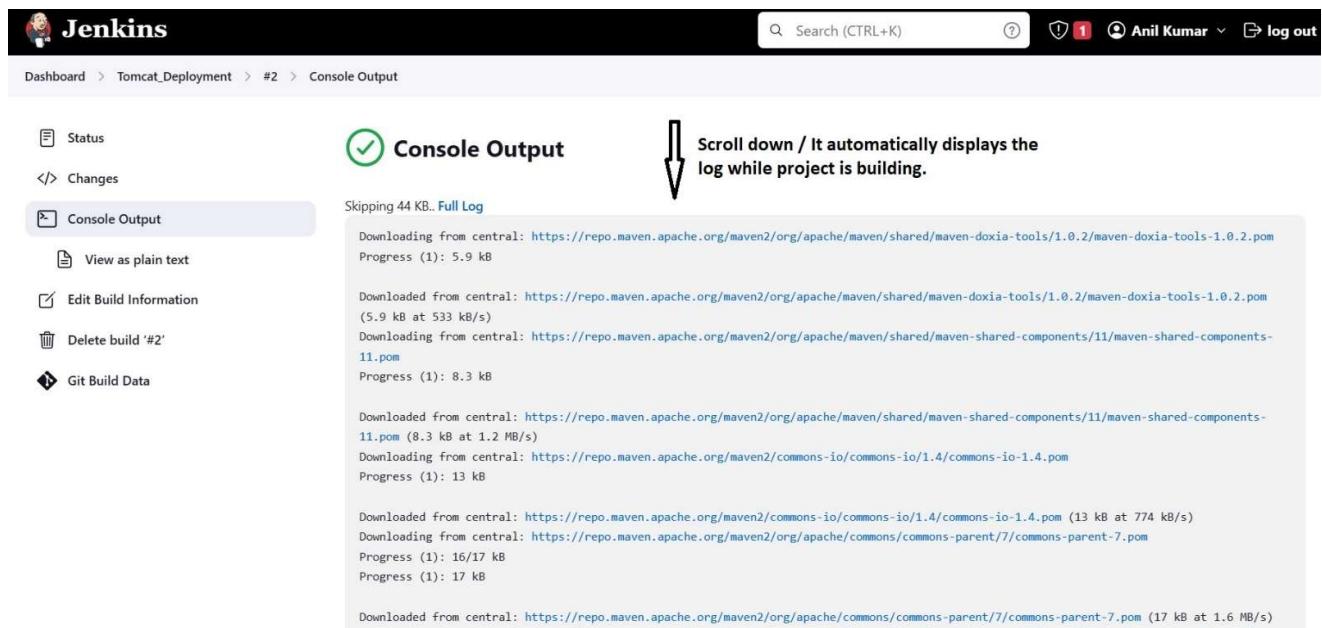
This screenshot is similar to the previous one, showing the Jenkins interface for the 'Tomcat_Deployment' project. The 'Build Now' button in the sidebar is highlighted with a red box and a callout 'Click on'. In the 'Build History' section, the first build entry is also highlighted with a red box and a callout 'Click on'. The build entry details are visible: '#1 Aug 11, 2023, 6:16 AM'. The bottom of the screen shows the same 'Atom feed' links as the previous screenshot.

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



The screenshot shows the Jenkins interface for a build named 'Tomcat_Deployment' (Build #2). The top navigation bar includes a search bar and user information. Below it, the breadcrumb path is 'Dashboard > Tomcat_Deployment > #2'. On the left, there's a sidebar with links like Status, Changes, Console Output (which is highlighted with a black border), Edit Build Information, Delete build '#2', and Git Build Data. The main content area displays the build status as 'Build #2 (Aug 11, 2023, 6:16:50 AM)' with a green checkmark icon. It shows 'No changes.' and was 'Started by user Anil Kumar'. The 'git' section indicates the revision and repository: cb677bbeb4b8bd84a67ed409bd7f1ad5f6036d79 and https://github.com/mannem302/AnilKumar.git, with a commit message 'refs/remotes/origin/main'. A callout arrow points from the text 'Click on' to the 'Console Output' button in the sidebar.

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



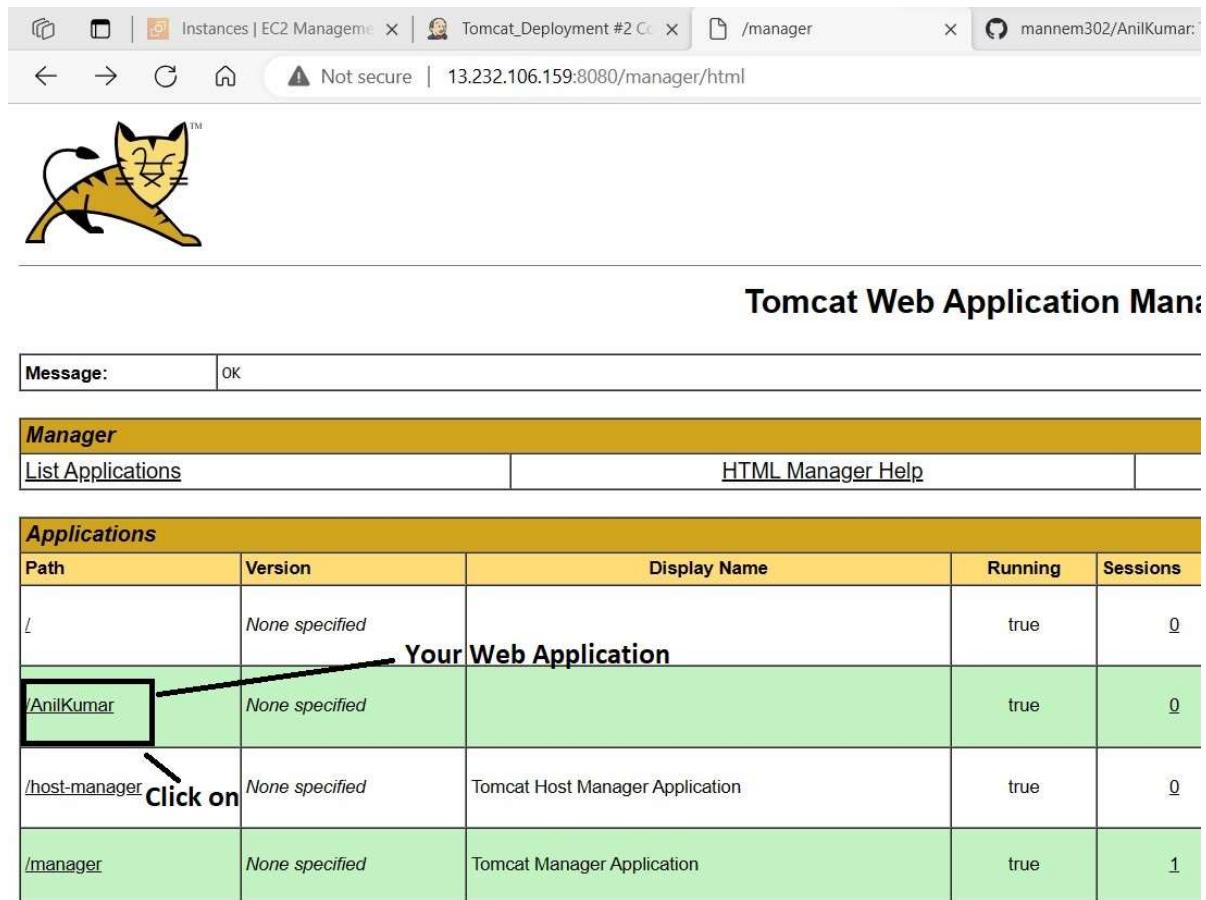
The screenshot shows the Jenkins 'Console Output' page for the same build. The top navigation bar and sidebar are identical to the previous screenshot. The main content area has a large green checkmark icon and the title 'Console Output'. It includes a note 'Scroll down / It automatically displays the log while project is building.' with a downward arrow. The log output shows Maven download progress for various dependencies from central repositories, such as maven-doxia-tools, maven-shared-components, commons-io, and commons-parent. The log ends with the download of commons-parent-7.pom.

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 web browser window with the following details:

- Address bar: `http://13.232.106.159:8080/manager/html`
- Page title: **Tomcat Web Application Manager**
- Message: `OK`
- Manager section:
 - List Applications: `/` (None specified), `/AnilKumar` (None specified), `/host-manager` (None specified), `/manager` (None specified)
 - HTML Manager Help
- Applications section:

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

A red box highlights the `/AnilKumar` row in the applications table. A red arrow points from the text "Click on" to the `/AnilKumar` link in the table.

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

Sonar



You have successfully deployed, web-application in Tomcat Server

Created By

Anil Kumar Mannem

SonarQube Installation

For code quality analysis, initially we require sonarqube server and later we have to integrate it with jenkins.

Follow step by steps for sonarqube installation.

Initially launch a t2.medium instance in AWS and then execute the following commands.

sudo apt-get update

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

```
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/main Translation-en [16.3 kB]
Get:25 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/main amd64 c-n-f Metadata [1420 B]
Get:26 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/restricted amd64 c-n-f Metadata [116 B]
Get:27 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [25.0 kB]
Get:28 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe Translation-en [16.3 kB]
Get:29 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64 c-n-f Metadata [880 B]
Get:30 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:31 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [2362 kB]
Get:32 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [373 kB]
Get:33 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [13.0 kB]
Get:34 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [2070 kB]
Get:35 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [289 kB]
Get:36 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 c-n-f Metadata [580 B]
Get:37 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [872 kB]
Get:38 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [183 kB]
Get:39 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [19.0 kB]
Get:40 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [23.6 kB]
Get:41 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [5504 B]
Get:42 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [548 B]
Fetched 28.1 MB in 5s (6076 kB/s)
Reading package lists... Done
ubuntu@ip-10-0-10-167:~$
```

packages downloaded

Then, execute the following commands for upgrading the packages.

sudo apt-get upgrade -y

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

```
See system logs and 'systemctl status snapd.mounts-pre.target' for details.
Setting up systemd-sysv (245.4-4ubuntu3.22) ...
Setting up libnss-systemd:amd64 (245.4-4ubuntu3.22) ...
Setting up libpam-systemd:amd64 (245.4-4ubuntu3.22) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for initramfs-tools (0.136ubuntu6.7) ...
update-initramfs: Generating /boot/initrd.img-5.15.0-1036-aws
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
Processing triggers for rsyslog (8.2001.0-1ubuntu1.3) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for microcode-initrd (2~20.04.0) ...
Processing triggers for dbus (1.12.16-2ubuntu2.3) ...
Processing triggers for ca-certificates (20230311ubuntu0.20.04.1) ...
Updating certificates in /etc/ssl/certs ...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done. Packages upgraded
ubuntu@ip-10-0-10-167:~$
```

To start the sonarqube server, we require java. To install java execute the following commands.

sudo apt-get install openjdk-17-jdk -y

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

```
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jstat to provide /usr/bin/jstat (jstat) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jstated to provide /usr/bin/jstated (jstated) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/serialver to provide /usr/bin/serialver (serialver) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jhsdb to provide /usr/bin/jhsdb (jhsdb) in auto mode
Setting up libgtk2.0-0:amd64 (2.24.32-4ubuntu4) ...
Setting up humanity-icon-theme (0.6.15) ...
Setting up libgail18:amd64 (2.24.32-4ubuntu4) ...
Setting up libgtk2.0-bin (2.24.32-4ubuntu4) ...
Setting up libgail-common:amd64 (2.24.32-4ubuntu4) ...
Setting up openjdk-17-jre:amd64 (17.0.8+7~20.04.2) ...
Setting up ubuntu-mono (19.04-0ubuntu3) ...
Setting up openjdk-17-jdk:amd64 (17.0.8+7~20.04.2) ...
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jconsole to provide /usr/bin/jconsole (jconsole) in auto mode
Processing triggers for ca-certificates (20230311ubuntu0.20.04.1) ...
Updating certificates in /etc/ssl/certs ...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...

done.
done. Installing Java
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
Processing triggers for systemd (245.4-4ubuntu3.22) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libgdk-pixbuf2.0-0:amd64 (2.40.0+dfsg-3ubuntu0.4) ...
ubuntu@ip-10-0-10-167:~$
```

To validate the java installation, execute the following command.

java --version

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

```
ubuntu@ip-10-0-10-167:~$ java --version
openjdk 17.0.8 2023-07-18 Displays Java Version
OpenJDK Runtime Environment (build 17.0.8+7-Ubuntu-120.04.2)
OpenJDK 64-Bit Server VM (build 17.0.8+7-Ubuntu-120.04.2, mixed mode, sharing)
```

Now execute the following command, to enter into opt directory for downloading sonarqube.

cd /opt

After entering into that directory. Execute the below command for downloading.

sudo wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.9.0.65466.zip

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

```
ubuntu@ip-10-0-10-167:/opt$ sudo wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.9.0.65466.zip
--2023-08-18 04:09:32-- https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.9.0.65466.zip
Resolving binaries.sonarsource.com (binaries.sonarsource.com) ... 54.182.0.53, 54.182.0.125, 54.182.0.110, ...
Connecting to binaries.sonarsource.com (binaries.sonarsource.com)|54.182.0.53|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 293899334 (280M) [binary/octet-stream]
Saving to: 'sonarqube-9.9.0.65466.zip'

sonarqube-9.9.0.65466.zip      100%[=====] 280.28M   101MB/s    in 2.8s

2023-08-18 04:09:34 (101 MB/s) - 'sonarqube-9.9.0.65466.zip' saved [293899334/293899334]
```

Downloading Sonarqube

As we have downloaded zip file, to unzip that folder we require unzip software and it shall be installed by executing the following commands.

sudo apt-get install unzip -y

After completion of installation, execute the following steps

sudo unzip sonarqube-9.9.0.65466.zip

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

```
inflating: sonarqube-9.9.0.65466/web/images/alm/bitbucket-white.svg
inflating: sonarqube-9.9.0.65466/web/images/alm/bitbucket.svg
inflating: sonarqube-9.9.0.65466/web/images/alm/github-white.svg
inflating: sonarqube-9.9.0.65466/web/images/SonarLint-connection-request.png
inflating: sonarqube-9.9.0.65466/web/images/check.svg
inflating: sonarqube-9.9.0.65466/web/images/saml.png
inflating: sonarqube-9.9.0.65466/web/apple-touch-icon-144x144.png
inflating: sonarqube-9.9.0.65466/web/apple-touch-icon-114x114.png
inflating: sonarqube-9.9.0.65466/web/apple-touch-icon-57x57.png
inflating: sonarqube-9.9.0.65466/web/index.html
  creating: sonarqube-9.9.0.65466/lib/jdbc/                                Unzip u/p
  creating: sonarqube-9.9.0.65466/lib/jdbc/mssql/
inflating: sonarqube-9.9.0.65466/lib/jdbc/mssql/mssql-jdbc-11.2.2.jre17.jar
  creating: sonarqube-9.9.0.65466/lib/jdbc/postgresql/
inflating: sonarqube-9.9.0.65466/lib/jdbc/postgresql/postgresql-42.5.1.jar
  creating: sonarqube-9.9.0.65466/lib/jdbc/h2/
inflating: sonarqube-9.9.0.65466/lib/jdbc/h2-2.1.214.jar
inflating: sonarqube-9.9.0.65466/lib/sonar-shutdowner-9.9.0.65466.jar
  creating: sonarqube-9.9.0.65466/elasticsearch/plugins/
```

After extracting the zip folder, rename that folder by using the following command.

sudo mv /opt/sonarqube-9.9.0.65466 /opt/sonarqube

We can't run the sonarqube as a root user. So, it's time to create new group and user to run sonarqube server. Execute the below commands.

sudo groupadd sonar

sudo useradd -c "user to run SonarQube" -d /opt/sonarqube -g sonar sonar

sudo chown -R sonar:sonar /opt/sonarqube

sudo passwd sonar (set the password)

Now edit the sonar script file by executing the following commands.

sudo vi /opt/sonarqube/bin/linux-x86-64/sonar.sh

After executing the above command, the following page will appear and we have to paste the following content in that file.

RUN_AS_USER=sonar



```
#! /bin/sh
APP_NAME="SonarQube"

# By default, java from the PATH is used, except if SONAR_JAVA_PATH env variable is set
findjava() {
    if [ -z "${SONAR_JAVA_PATH}" ]; then
        if ! command -v java 2>&1; then
            echo "Java not found. Please make sure that the environmental variable SONAR_JAVA_PATH points to a Java e:
        fi
        JAVA_CMD=java
    else
        if ! [ -x "${SONAR_JAVA_PATH}" ] || ! [ -f "${SONAR_JAVA_PATH}" ]; then
            echo "File '${SONAR_JAVA_PATH}' is not executable. Please make sure that the environmental variable SONAR_JAVA_PATH points to a Java e:
        fi
        JAVA_CMD="${SONAR_JAVA_PATH}"
    fi
}
```

After pasting that content, the page looks like as shown below.



```
RUN_AS_USER=sonar Content pasted here
APP_NAME="SonarQube"

# By default, java from the PATH is used, except if SONAR_JAVA_PATH env variable is set
findjava() {
    if [ -z "${SONAR_JAVA_PATH}" ]; then
        if ! command -v java 2>&1; then
            echo "Java not found. Please make sure that the environmental variable SONAR_JAVA_PATH points to a Java e:
        fi
        JAVA_CMD=java
    else
        if ! [ -x "${SONAR_JAVA_PATH}" ] || ! [ -f "${SONAR_JAVA_PATH}" ]; then
            echo "File '${SONAR_JAVA_PATH}' is not executable. Please make sure that the environmental variable SONAR_JAVA_PATH points to a Java e:
        fi
        JAVA_CMD="${SONAR_JAVA_PATH}"
    fi
}
```

To save the file :wq

Now, give the sudo access to the sonar user, by executing the following command.

sudo visudo

After executing the above command, the page shown in next page will appear and the following content to be pasted.

sonar ALL=(ALL:ALL) ALL

```

GNU nano 4.8                               /etc/sudoers.tmp

#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
Content to be pasted here

# See sudoers(5) for more information on "#include" directives:

#includeincludedir /etc/sudoers.d
[]

^G Get Help      ^O Write Out     ^W Where Is      ^K Cut Text      ^J Justify      ^C Cur Pos
^X Exit         ^R Read File     ^\ Replace       ^U Paste Text    ^T To Spell     ^
^_ Go To Line

```

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

```

GNU nano 4.8
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
sonar  ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "#include" directives

#includeincludedir /etc/sudoers.d
[]

^G Get Help      ^O Write Out     ^W Where Is      ^K Cut Text
^X Exit         ^R Read File     ^\ Replace       ^U Paste Text

```

press **ctrl+x** to save the file.

Now, view the default shell used for sonar user, by executing the following command.

sudo vi /etc/passwd

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

```
fwupd-refresh:x:112:116:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
ec2-instance-connect:x:113:65534::/nonexistent:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
sonar:x:1001:1001:user to run SonarQube:/opt/sonarqube:/bin/sh
~
~
~
~
~
~
"/etc/passwd" 36L, 1965C
```

default shell is sh, change it to bash

After changing sh to bash, the page looks like as shown below.

```
ec2-instance-connect:x:113:65534::/nonexistent:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
sonar:x:1001:1001:user to run SonarQube:/opt/sonarqube:/bin/bash
```

Changed to bash

press esc and :wq to save the file.

Now, before starting sonarqube service. Switch to sonar user by executing following command.

sudo su sonar

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

```
ubuntu@ip-10-0-10-167:/opt$ sudo su sonar
sonar@ip-10-0-10-167:/opt$
```

User switched to sonar

Then execute the following command, for entering into sonarqube directory and start the service.

cd /opt/sonarqube/bin/linux-x86-64/

./sonar.sh start

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

```
sonar@ip-10-0-10-167:~/bin/linux-x86-64$ ./sonar.sh start
/usr/bin/java
Starting SonarQube ...
Started SonarQube.
```

Started

Now, execute the following command to check the sonarqube status

./sonar.sh status

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

```
sonar@ip-10-0-10-167:~/bin/linux-x86-64$ ./sonar.sh status
/usr/bin/java
SonarQube is running (25735). Running status
```

After executing the above commands, check the sonarqube logs, by using following commands.

tail /opt/sonarqube/logs/sonar.log

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

```
pt/sonarqube/temp/sq-process11383705189953608995properties
2023.08.18 04:39:44 INFO app[] [o.s.a.SchedulerImpl] Process[web] is up
2023.08.18 04:39:44 INFO app[] [o.s.a.ProcessLauncherImpl] Launch process[COMPUTE_ENGINE] from [/opt/s
Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=/opt/sonarqube/temp -XX:-OmitStackTraceOnF
--add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add
ase/sun.nio.ch=ALL-UNNAMED --add-opens=java.management/sun.management=ALL-UNNAMED --add-opens=jdk.mana
hat.fips=false -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|
onarqube/lib/jdbc/h2/h2-2.1.214.jar org.sonar.ce.app.CeServer /opt/sonarqube/temp/sq-process9103369173
2023.08.18 04:39:46 WARN app[] [startup] #####
2023.08.18 04:39:46 WARN app[] [startup] Default Administrator credentials are still being used. Make
2023.08.18 04:39:46 WARN app[] [startup] #####
2023.08.18 04:39:50 INFO app[] [o.s.a.SchedulerImpl] Process[ce] is up
2023.08.18 04:39:50 INFO app[] [o.s.a.SchedulerImpl] SonarQube is operational
```

Now, stop the service and add the system service, by executing the following command.

./sonar.sh stop

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

```
sonar@ip-10-0-10-167:~/bin/linux-x86-64$ ./sonar.sh stop
/usr/bin/java
Gracefully stopping SonarQube ...
Stopped SonarQube.
sonar@ip-10-0-10-167:~/bin/linux-x86-64$
```

After stopping the service, execute the following command. To create systemd service.

sudo vi /etc/systemd/system/sonar.service

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



Content mentioned in the next page shall be pasted in above screen.

[Unit]

Description=SonarQube service

After=syslog.target network.target

[Service]

Type=forking

ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start

ExecStop=/opt/sonarqube/bin/linux-x86-64/sonar.sh stop

User=sonar

Group=sonar

Restart=always

LimitNOFILE=65536

LimitNPROC=4096

[Install]

WantedBy=multi-user.target

After pasting the content, execute the following command to start the sonarqube service.

sudo systemctl start sonar

After starting the service, check the status by executing the following commands.

sudo systemctl status sonar

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

```
ubuntu@ip-10-0-10-167:/opt$ sudo systemctl status sonar
● sonar.service - SonarQube service
   Loaded: loaded (/etc/systemd/system/sonar.service; disabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-08-18 04:56:37 UTC; 11s ago
     Process: 26230 ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start (code=exited, status=0/SUCCESS)
    Main PID: 26253 (java)
      Tasks: 100 (limit: 4686)
     Memory: 818.7M
        CPU: 0.000 CPU(s) since start
       CGroup: /system.slice/sonar.service
               └─26253 java -Xms8m -Xmx32m --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-L
                  ├─26278 /usr/lib/jvm/java-17-openjdk-amd64/bin/java -XX:+UseG1GC -Djava.io.tmpdir=/opt/sonarqube/temp -XX:ErrorFile=/c
                  ├─26388 /usr/lib/jvm/java-17-openjdk-amd64/bin/java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=/c

Aug 18 04:56:37 ip-10-0-10-167 systemd[1]: Starting SonarQube service...
Aug 18 04:56:37 ip-10-0-10-167 sonar.sh[26230]: /usr/bin/java
Aug 18 04:56:37 ip-10-0-10-167 sonar.sh[26230]: Starting SonarQube...
Aug 18 04:56:37 ip-10-0-10-167 sonar.sh[26230]: Started SonarQube.
Aug 18 04:56:37 ip-10-0-10-167 systemd[1]: Started SonarQube service.
[lines 1-17/17 (END)]
```

We can enable the sonarqube service at system startup by using the following command.

sudo systemctl enable sonar

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

```
ubuntu@ip-10-0-10-167:/opt$ sudo systemctl enable sonar
Created symlink /etc/systemd/system/multi-user.target.wants/sonar.service → /etc/systemd/system/sonar.service.
```

Now open the following URL in browser for viewing sonarqube webpage.

http://IP_ADDRESS:9000/ (Sonarqube default port is 9000)

After entering that URL, the following page will appear. In that we have to enter username and password.

Default username- **admin**

Default password- **admin**

Log in to SonarQube

Default Username is admin — admin

Default password is admin —

Click on — **Log in** Cancel

After clicking on 'log in' button, the following page will appear.

Update your password

This account should not use the default password.

Enter a new password

All fields marked with * are required

Old Password *

.....

New Password * **We have to**

..... **change password**

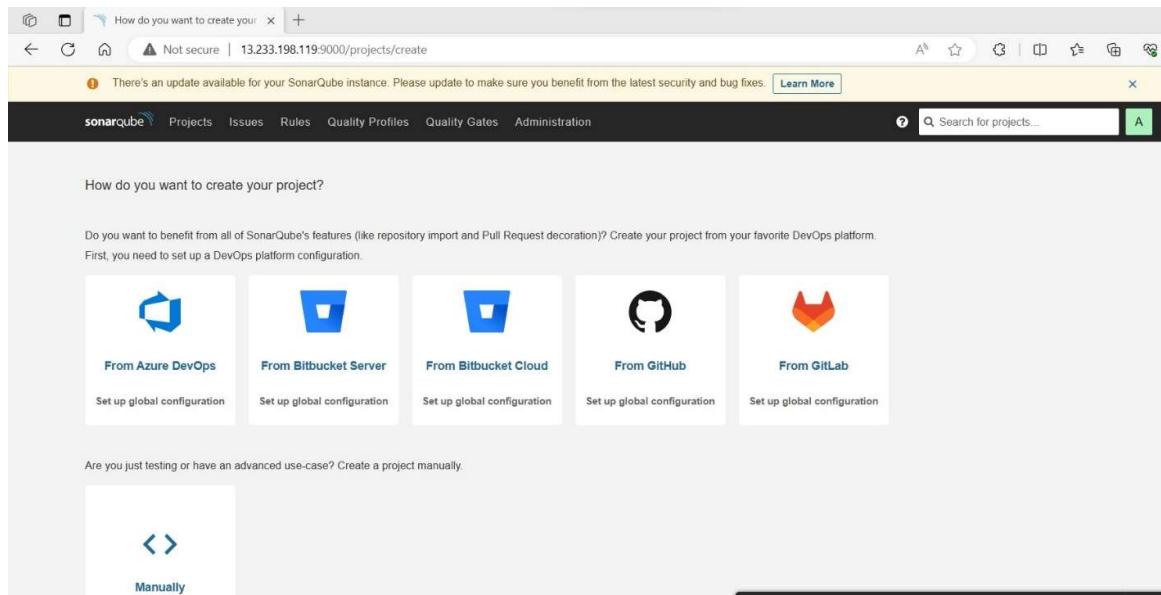
Confirm Password *

.....

Update

Click on

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



The screenshot shows the SonarQube 'Create Project' interface. At the top, there's a message about an update available. Below it, a navigation bar includes 'sonarqube' and links for 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. A search bar at the top right contains the placeholder 'Search for projects...'. The main content area is titled 'How do you want to create your project?'. It explains that users can benefit from SonarQube's features by creating a project from their favorite DevOps platform. It lists five options: 'From Azure DevOps', 'From Bitbucket Server', 'From Bitbucket Cloud', 'From GitHub', and 'From GitLab', each with a corresponding icon and a 'Set up global configuration' link below. Below these options, there's a section for manual creation with a 'Manually' button and a 'Create' button.

Sonarqube was successfully installed and in running state.

Note:- In this process we are using embedded database for testing purpose.

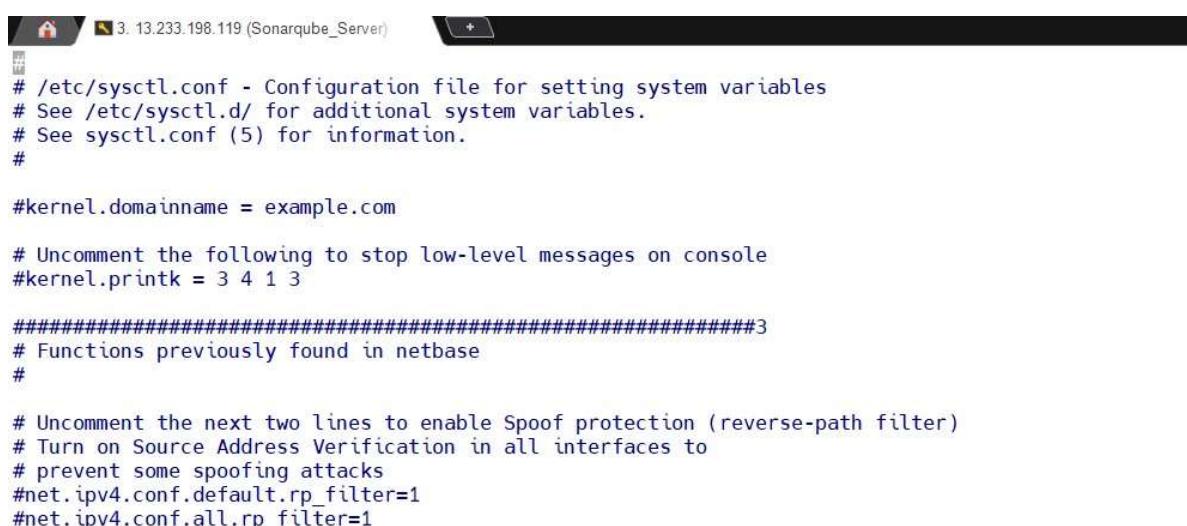
Troubleshooting Activity

Sometimes, sonarqube will not be started and displays the following error.

loaded plugin [org.elasticsearch.transport.Netty4Plugin] ERROR: [1] bootstrap checks failed. max virtual memory areas vm.max_map_count [65530] is too low, increase to at least [262144].

In that case we have to execute the following command.

sudo vi etc/sysctl.conf



```
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#
#kernel.domainname = example.com
#
# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3
#####
# Functions previously found in netbase
#
# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1
```

paste the following line in the above file

vm.max_map_count=262144

Sometimes, we need to execute the following command. If we see any error in sonarqube status

sudo systemctl restart sonar

Now, we will integrate the sonarqube server with jenkins server and execute the code quality analysis. For this we require sonarqube scanner plugin.

Follow the following steps for integrating both.

Now, again open the jenkins dashboard and click on ‘**manage jenkins**’ button. As shown in below image.

The screenshot shows the Jenkins dashboard at the URL 65.1.132.131:8080. The 'Manage Jenkins' button is highlighted with a red box and a 'Click on' arrow pointing to it. The dashboard includes sections for New Item, People, Build History, Project Relationship, Check File Fingerprint, and a main Jenkins instance table for 'Tomcat_Deployment'. Below the table are build queue and executor status sections.

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

The screenshot shows the Jenkins System Configuration page. The 'Plugins' section is highlighted with a red box and a 'Click on' arrow pointing to it. Other sections visible include System Configuration, Tools, Nodes and Clouds, and Build Executor Status.

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

The screenshot shows the Jenkins 'Available plugins' page. A search bar at the top right contains the text 'sonar'. Below it, a list of available plugins is shown. The first item is 'SonarQube Scanner 2.15', which has a checked checkbox next to its name. A callout bubble points to this checkbox with the text 'Click on'. Another callout bubble points to the 'Search for sonar' input field with the text 'Search for sonar'. On the left sidebar, there are links for 'Updates', 'Available plugins' (which is highlighted with a red box and a 'Click on' callout), 'Installed plugins', and 'Advanced settings'. At the bottom of the page, there are two buttons: 'Install without restart' (highlighted with a red box and a 'Click on' callout) and 'Download now and install after restart'. A note indicates that information was obtained 5 hours and 31 minutes ago, and a 'Check now' link is provided.

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

The screenshot shows the Jenkins 'Download progress' page. A callout bubble points to the 'Download progress' link in the left sidebar with the text 'Click on'. The main area displays the 'Download progress' section for the 'SonarQube Scanner' plugin. It shows two status entries: 'SonarQube Scanner' and 'Loading plugin extensions', both marked as 'Success' with green checkmarks. A callout bubble points to these success messages with the text 'Plugin successfully installed'. Below this, there are two links: 'Go back to the top page' (with a note '(you can start using the installed plugins right away)') and 'Restart Jenkins when installation is complete and no jobs are running' (with an unchecked checkbox).

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', 'Project Relationship', 'Check File Fingerprint', and 'Manage Jenkins'. Below this is a 'Build Queue' section showing 'No builds in the queue.' The main area is titled 'Manage Jenkins' and contains sections for 'System Configuration' (with 'System' and 'Nodes and Clouds' options) and 'Tools'. A callout box with an arrow points to the 'Tools' section, which is described as 'Configure tools, their locations and automatic installers.'

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

The screenshot shows the Jenkins 'Tools' configuration page. It includes sections for 'Maven Configuration' (with 'Default settings provider' and 'Use default maven settings') and 'Default global settings provider' (with 'Use default maven global settings'). At the bottom, there's a note: 'Scroll down the page, to find sonarqube scanner' with a downward-pointing arrow. The URL in the browser bar is 65.1.132.131:8080/manage/configureTools/.

After scrolling down the page, you will find the following screen.



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

Dashboard > Manage Jenkins > Tools

List of SonarQube Scanner installations on this system

Add SonarQube Scanner

≡ SonarQube Scanner

Name Enter Name
sonar_scanner

Install automatically ?

≡ Install from Maven Central

Version
SonarQube Scanner 5.0.1.3006

Add Installer ▾

Click on dropdown icon

After clicking on dropdown icon as shown in above screenshot. The following page will appear.

≡ SonarQube Scanner

Name
sonar_scanner

Install automatically ?

≡ Install from Maven Central

Version
SonarQube Scanner 4.8.0.2856

Add Installer ▾

Add SonarQube Scanner

Second click on
Save

First click on
Apply

After clicking on ‘Save’ button the following 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', 'Project Relationship', 'Check File Fingerprint', and 'Manage Jenkins'. Below this are dropdown menus for 'Build Queue' (No builds in the queue) and 'Build Executor Status'. The main area is titled 'Manage Jenkins' and contains sections for 'System Configuration' and 'Tools'. A callout bubble points to the 'System' link under 'System Configuration' with the instruction 'Click on'. The 'System' section is highlighted with a black border. Another callout bubble points to the 'Tools' section with the instruction 'Click on'. The 'Tools' section includes a link to 'Configure tools, their locations and automatic installers.'

After clicking on ‘System’ button. The following page will appear.

The screenshot shows the Jenkins System configuration page. At the top, it says 'System [Jenkins]'. The main content area is titled 'System'. It has a 'Home directory' section with a default value of '/var/lib/jenkins'. Below this is a 'System Message' section with a placeholder for posting notifications to users. A callout bubble with a downward arrow points to the bottom of the page with the instruction 'Scroll down the page, to find SonarQube servers'.

After scrolling down the page, you will find the sonarqube server option, as shown below.

The screenshot shows the Jenkins System configuration page. At the top, there are three checkboxes: 'Disable deferred wipeout on this node' (unchecked), 'Environment variables' (unchecked), and 'Tool Locations' (unchecked). Below this is a section titled 'SonarQube servers'. It contains a note: 'If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.' followed by a checkbox labeled 'Environment variables' which is unchecked. A link 'Enable injection of SonarQube server configuration as build environment variables' is provided. Under 'SonarQube installations', it says 'List of SonarQube installations' and shows a button 'Add SonarQube' which is highlighted with a black rectangle and a callout arrow pointing to it with the text 'Click on'.

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

The screenshot shows the 'Add SonarQube' configuration page. It has a dashed border on the left. Inside, there is a 'Name' field containing 'sonarqube' with a callout arrow pointing to it labeled 'Click on'. Next is a 'Server URL' field with a placeholder 'Default is http://localhost:9000' and a value 'http://13.233.198.119:9000/' which is also highlighted with a black rectangle and a callout arrow labeled 'Click on'. Below is a 'Server authentication token' field with a note: 'SonarQube authentication token. Mandatory when anonymous access is disabled.' and an empty input field. At the bottom left is an 'Advanced' dropdown menu with a callout arrow labeled 'Click on'.

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

SonarQube installations

List of SonarQube installations

Name

sonarqube

Server URL

Default is http://localhost:9000

http://13.233.198.119:9000/

Server authentication token

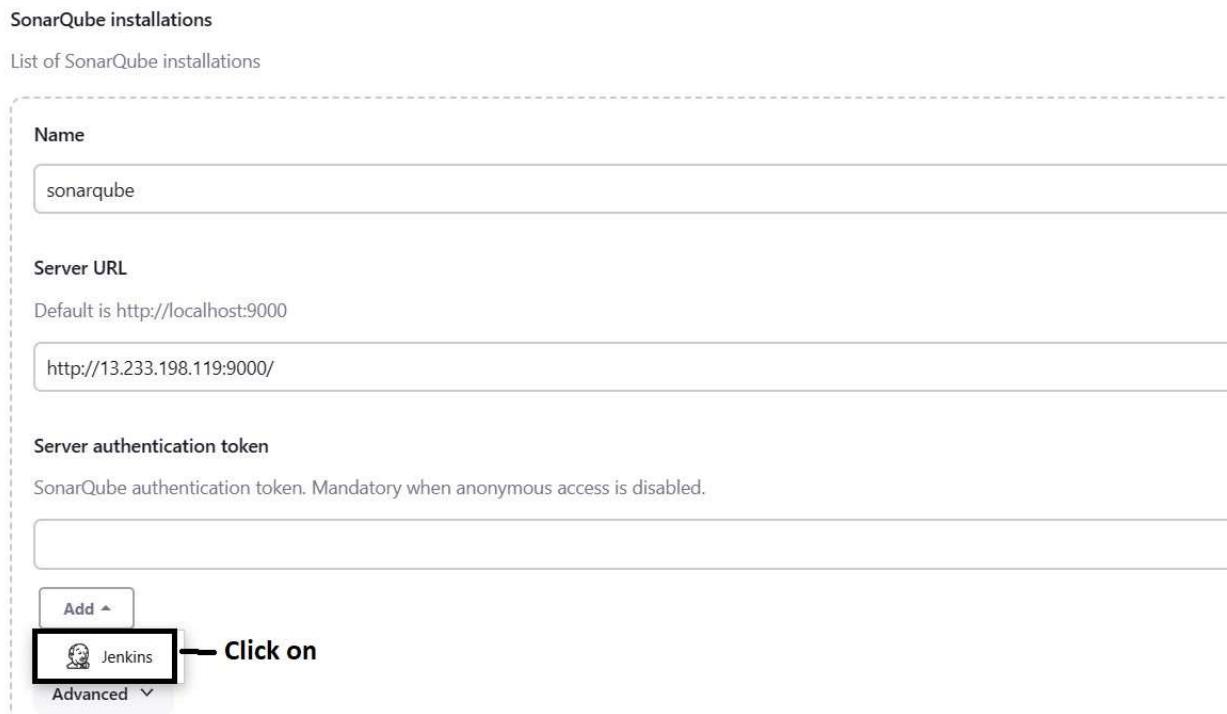
SonarQube authentication token. Mandatory when anonymous access is disabled.

Add ▲

Jenkins

Advanced ▼

Click on Jenkins



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

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

Username with password

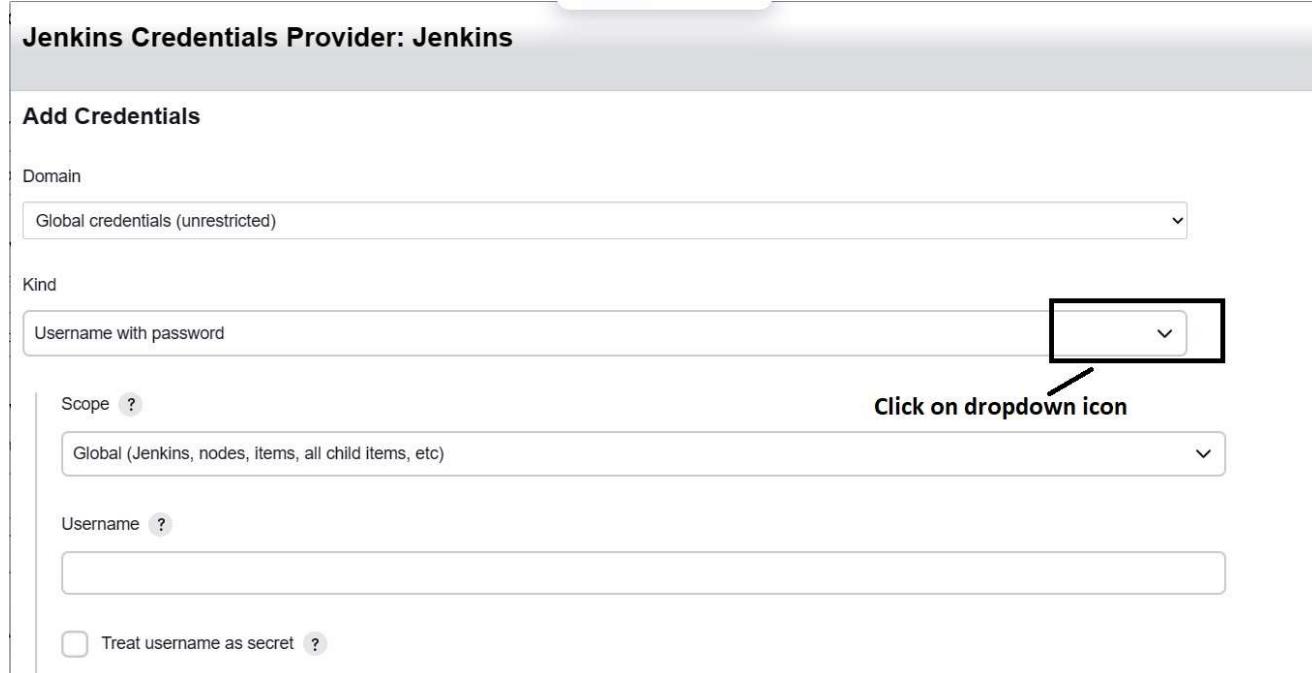
Scope ?

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

Username ?

Treat username as secret ?

Click on dropdown icon



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

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

Username with password

GitHub App

SSH Username with private key

Secret file

Secret text

Certificate

Select this option

Username ?

After selecting ‘Secret text’ option the following page will appear.

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

Secret text

Scope ?

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

Secret

We have to paste the token, Which is generated in
sonarqube server

ID ?

As shown in previous page screenshot, we have to paste the token in Secret box.
To generate the sonarqube token follow the below steps.

First open the sonarqube server webpage, as shown in below screenshot.

The screenshot shows a web browser window for 'sonarqube' at '13.233.198.119:9000/projects/create'. The 'Administration' tab is highlighted with a red box and a cursor arrow pointing to it. A yellow status bar at the top right says 'There's an update available for your SonarQube instance. Please update to make sure you benefit from the latest security and bug fixes.' Below the tabs, there's a section titled 'How do you want to create your project?' with five options: 'From Azure DevOps', 'From Bitbucket Server', 'From Bitbucket Cloud', 'From GitHub', and 'From GitLab'. Each option has a small icon and a 'Set up global configuration' link below it.

After clicking on ‘Administration’ button. The following page will appear.

The screenshot shows a web browser window for 'sonarqube' at '13.233.198.119:9000/admin/settings'. The 'Administration' tab is highlighted with a blue bar. Below the tabs, there's a 'General Settings' section with a 'Click on' callout pointing to the 'Security' tab in the navigation bar. The 'Security' tab is highlighted with a red box and a cursor arrow pointing to it. On the left, there's a sidebar with 'Analysis Scope', 'Authentication', and 'DevOps Platform Integrations'. On the right, there's a 'Duplications' section with a 'Cross project duplication detection' toggle switch set to 'off' (default).

After clicking on ‘Security’ dropdown menu, the following page will appear.

The screenshot shows the SonarQube Administration interface. At the top, there is a banner indicating an update available. Below the banner, the navigation bar includes links for sonarqube, Projects, Issues, Rules, Quality Profiles, Quality Gates, and Administration. The Administration link is underlined, indicating it is selected. A sub-menu for Administration is open, showing options: General, Users (which is highlighted with a red box and has an arrow pointing to it labeled "Select this option"), Groups, Global Permissions, and Permission Templates. Below the sub-menu, there are sections for Analysis Scope, Authentication, and Duplications. A note about Cross project duplication detection is present, mentioning it is deprecated. A toggle switch is shown next to the note.

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

The screenshot shows the SonarQube Users Administration page. The browser title is "Users - Administration". The main content area is titled "Users" and describes the task of creating and administering individual users. A search bar is available. A table lists a single user: "Administrator admin". The table columns include SCM Accounts, Last connection, Groups, and Tokens. The "Groups" column shows "sonar-administrators" and "sonar-users". The "Tokens" column shows a gear icon with a plus sign and a dropdown arrow. A callout bubble with the text "Click on" points to the gear icon.

After clicking on that button shown in earlier screenshot, which is located under Tokens. The following page will appear.

The screenshot shows the 'Tokens of Administrator' page in the SonarQube Administration section. A modal dialog is open for generating a token. The 'Generate Tokens' form contains fields for 'Name' (with placeholder 'Enter Token Name'), 'Expires in' (set to '30 days'), and a 'Generate' button. Below the form, a table lists tokens. The first row, labeled 'No tokens', has a 'Name' field containing 'Enter name' and a note 'You can select or leave it default'. A cursor arrow points to the 'Generate' button with the text 'click on' above it. At the bottom right of the modal is a 'Done' button.

After clicking on ‘Generate’ button. The following page will appear.

The screenshot shows the same 'Tokens of Administrator' page after the 'Generate' button was clicked. The modal now displays a success message: 'New token "sonar_token" has been created. Make sure you copy it now, you won't be able to see it again!' Below the message, a 'Copy' button is highlighted with a cursor arrow pointing to it, and the text 'Click on' is placed above the button. The table below shows the newly created token 'sonar_token' with details: Type 'User', Project 'squ_3ec2b519a7595e0724cbd341b9739caf1e6c4d1e', Last use 'Never', Created 'August 18, 2023', and Expiration 'September 17, 2023'. A 'Revoke' button is at the bottom right of the table. The 'Done' button is also visible at the bottom right of the modal.

Finally sonarqube token was generated.

After clicking on ‘Copy’ button, the token will be copied and it shall be pasted in the secret box as shown below.

Kind

Secret text

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

Secret
Generated token to be pasted here

ID ?
sonar_credentials **Enter ID**

Description ?
sonar_credentials **Enter Description**

Add **Click on** Cancel

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

Name
sonarqube

Server URL
Default is http://localhost:9000
http://13.233.198.119:9000/

Server authentication token
SonarQube authentication token. Mandatory when anonymous access is disabled.
Click on

- none -
sonar_credentials **Select this option**

Advanced ▾

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

Server authentication token

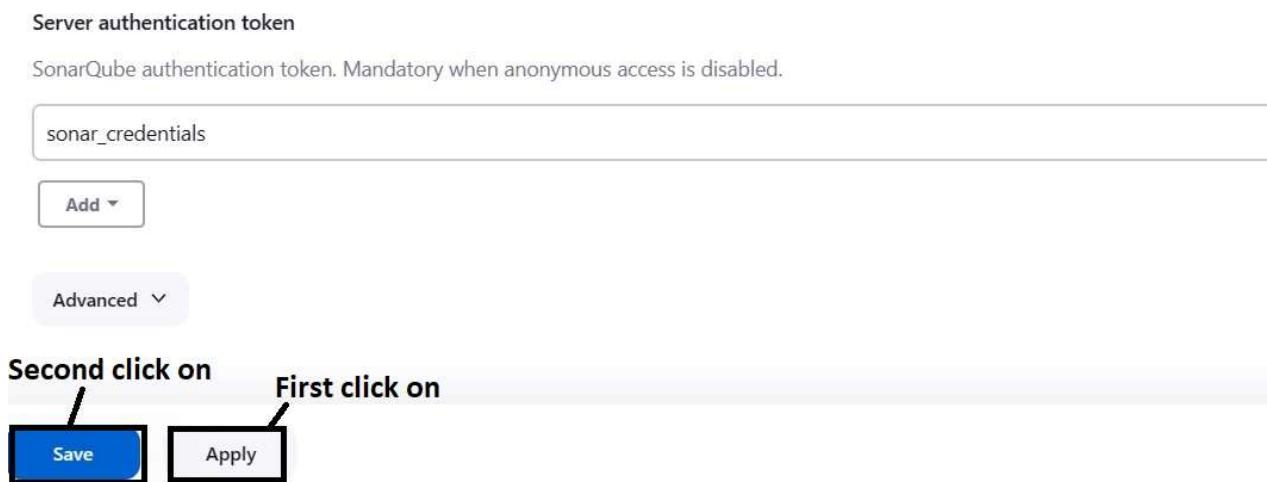
SonarQube authentication token. Mandatory when anonymous access is disabled.

sonar_credentials

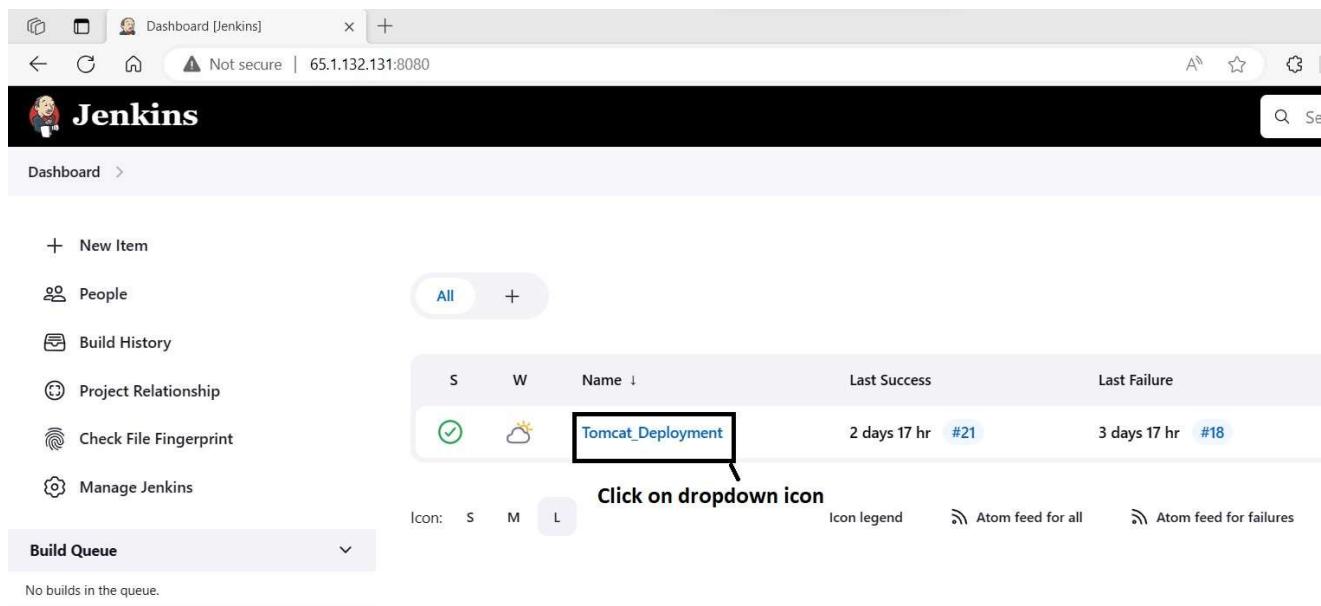
Add ▾

Advanced ▾

Second click on **Save**
First click on **Apply**



After clicking on ‘Save’ button, the following page will appear. Now we will configure the existing job to run code quality analysis.



The screenshot shows the Jenkins dashboard with the following details:

- Header: Dashboard [Jenkins] - Not secure | 65.1.132.131:8080
- Top navigation: New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins.
- Search bar: All
- Job list:
 - Tomcat_Deployment (highlighted with a box)
 - Last Success: 2 days 17 hr #21
 - Last Failure: 3 days 17 hr #18
- Job details for Tomcat_Deployment:
 - Icon: Green checkmark
 - Icon: Cloud with sun
 - Name: Tomcat_Deployment
 - Last Success: 2 days 17 hr #21
 - Last Failure: 3 days 17 hr #18
- Bottom navigation: Build Queue (dropdown), Icon legend, Atom feed for all, Atom feed for failures.

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

The screenshot shows the Jenkins dashboard with the 'Tomcat Deployment' project selected. A context menu is open over the project name, listing options: Changes, Workspace, Build Now, Configure (which is highlighted with a red box and has an arrow pointing to it), Delete Project, and Rename. The 'Configure' option is the target of the instruction.

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

The screenshot shows the Jenkins configuration page for the 'Tomcat Deployment' project. The 'General' tab is selected. On the left, a sidebar lists 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps' (which is highlighted with a red box and has an arrow pointing to it), and 'Post-build Actions'. The 'General' tab contains fields for 'Description' (with a preview link) and several checkboxes for build behaviors: 'Discard old builds', 'GitHub project', 'This project is parameterized', 'Throttle builds', and 'Execute concurrent builds if necessary'. An 'Advanced' dropdown is also present.

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

Configure

Build Steps

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Invoke top-level Maven targets ?

Maven Version

maven 3.9.4

Goals

clean package

Advanced ▾

Add build step ▾ Click on

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

Build Environment

Build Steps

Post-build Actions

Goals

clean package

Advanced ▾

Add build step ▾ Select this option

Filter

Execute SonarQube Scanner

Execute Windows batch command

Execute shell

Invoke Ant

Invoke Gradle script

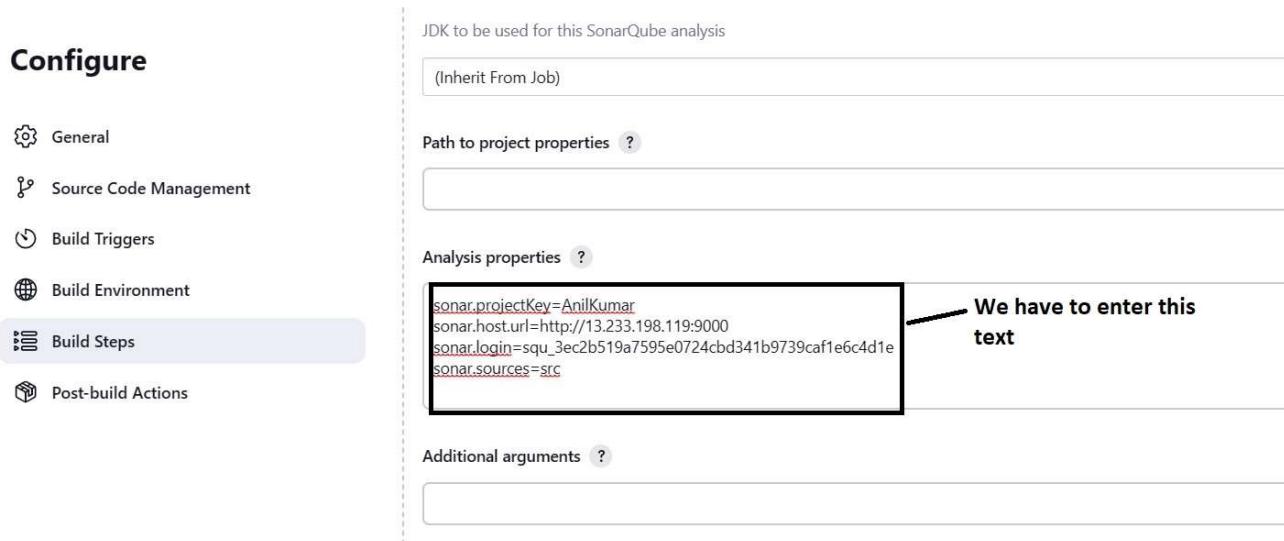
Invoke top-level Maven targets

Run with timeout

After selecting ‘Execute SonarQube Scanner’ option, the following page will appear.



After scrolling down to Analysis Properties option, we have to enter the following line in that empty field.



sonar.projectKey=AnilKumar
sonar.host.url= sonarqube url to be pasted
sonar.login= sonarqube token to be pasted
sonar.sources=src

After entering the above-mentioned text. click on apply and save as shown in below screenshot.

The screenshot shows the Jenkins configuration page for the 'Tomcat_Deployment' job. On the left, there's a sidebar with options: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. 'Post-build Actions' is currently selected and highlighted in grey. In the main panel, there are three sections: 'Analysis properties', 'Additional arguments', and 'JVM Options'. Under 'Analysis properties', the following SonarQube configuration is listed:

```
sonar.projectKey=AnilKumar  
sonar.host.url=http://13.233.198.119:9000  
sonar.login=squ_3ec2b519a7595e0724cbd341b9739caf1e6c4d1e  
sonar.sources=src
```

Below these sections is a button labeled 'Add build step ▾'. At the bottom of the configuration area, there are two buttons: 'Save' and 'Apply'. A callout arrow points from the text 'Second click on' to the 'Save' button. Another callout arrow points from the text 'First click on' to the 'Apply' button.

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

The screenshot shows the Jenkins project page for 'Tomcat_Deployment'. The top navigation bar includes links for Dashboard, Tomcat_Deployment, and Jenkins. Below the header, there's a status summary: 'Not secure | 65.1.132.131:8080/job/Tomcat_Deployment/'. The main content area has a dark header with the Jenkins logo and the project name 'Tomcat_Deployment'. On the left, there's a sidebar with links: Status, Changes, Workspace, Build Now (which is highlighted with a callout 'Click on'), Configure, Delete Project, SonarQube, and Rename. To the right of the sidebar, there's a 'SonarQube' section with a 'Permalinks' table containing the following data:

Link	Description
Last build (#23)	8 min 38 sec ago
Last stable build (#23)	8 min 38 sec ago
Last successful build (#23)	8 min 38 sec ago
Last failed build (#18)	3 days 18 hr ago
Last unsuccessful build (#18)	3 days 18 hr ago

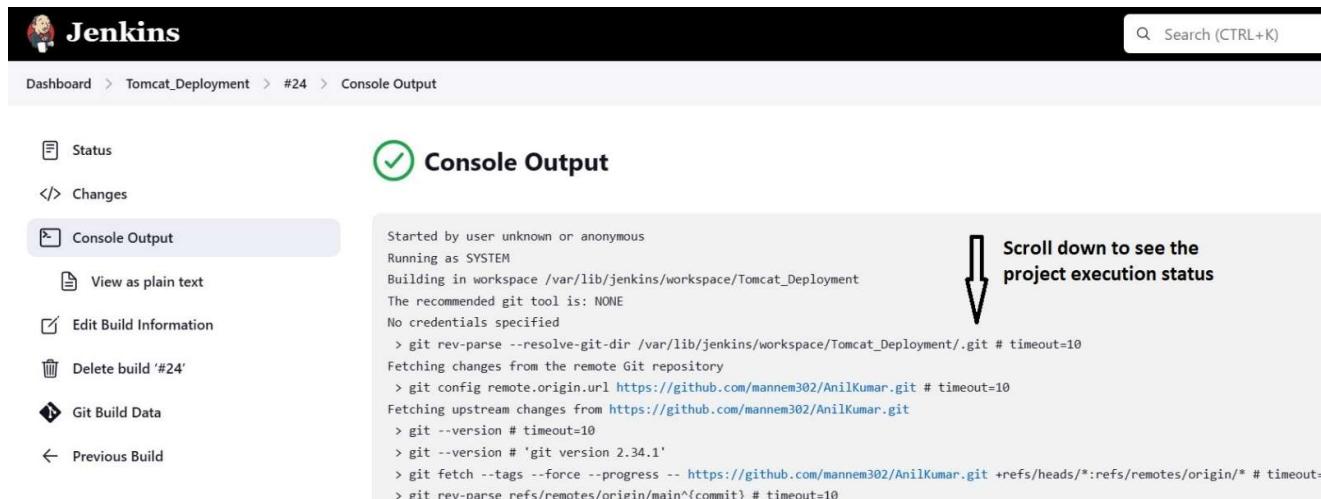
After clicking on ‘Build Now’ button, the following page will appear.

The screenshot shows the Jenkins dashboard for the project 'Tomcat_Depl'. On the left, there's a sidebar with various options: Status, Changes, Workspace, Build Now (which is highlighted with a red box), Configure, Delete Project, SonarQube, and Rename. Below this is a 'Build History' section with a 'trend' dropdown and a 'Filter builds...' input field. A specific build entry for '#24 (Aug 18, 2023, 11:39 AM)' is highlighted with a red box and has an arrow pointing to it from the text 'Click on'.

After clicking on that button, we can see console output, as shown in below screenshot.

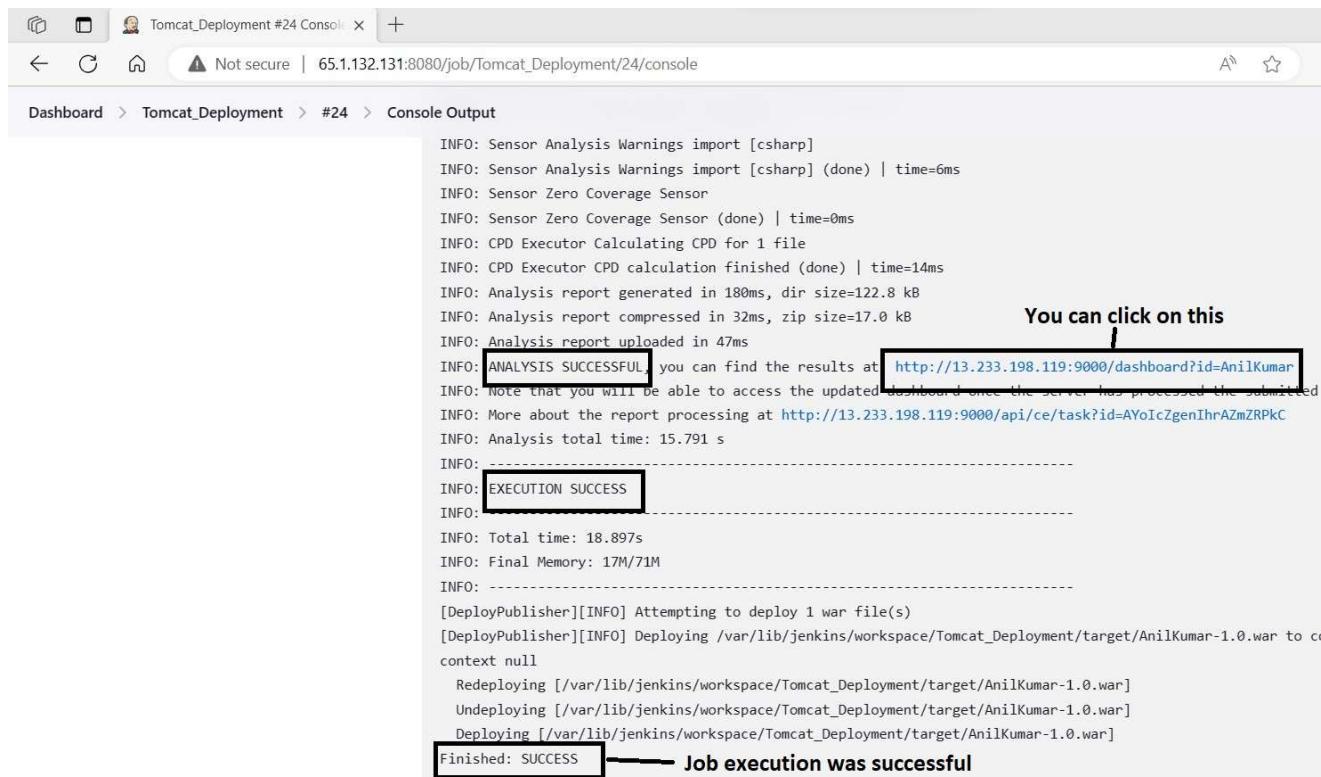
The screenshot shows the Jenkins build details for '#24 (Aug 18, 2023, 11:39:10 AM)'. At the top, there's a green checkmark icon and the text 'Build #24 (Aug 18, 2023, 11:39:10 AM)'. Below this is a 'Status' bar with 'Status' and 'Changes' tabs. A 'Console Output' link is highlighted with a red box and has an arrow pointing to it from the text 'Click on'. To the right, there's information about the build: 'No changes.', 'Started by anonymous user', 'Revision: ef10ed3d341cde14ef26e8d7ebe1c9a6525de2eb', 'Repository: https://github.com/mannem302/AnilKumar.git', and a list of commits: 'refs/remotes/origin/main'.

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



The screenshot shows the Jenkins interface for a build named 'Tomcat_Deployment #24'. The left sidebar has links for Status, Changes, Console Output (which is selected and highlighted in grey), View as plain text, Edit Build Information, Delete build '#24', Git Build Data, and Previous Build. The main content area is titled 'Console Output' with a checkmark icon. It displays the command-line output of the build process, starting with 'Started by user unknown or anonymous' and ending with 'git rev-parse refs/remotes/origin/main^{commit} # timeout=10'. A callout bubble on the right says 'Scroll down to see the project execution status' with an arrow pointing downwards.

After scrolling down the page, at the bottom you can find the project status.



The screenshot shows the Jenkins interface for a build named 'Tomcat_Deployment #24'. The top navigation bar includes icons for refresh, stop, and restart, followed by the build name 'Tomcat_Deployment #24 Console' and a '+' sign. Below the bar are standard browser controls (back, forward, home, search). The main content area shows the build history with the current step being 'Console Output'. The log output is displayed in a monospaced font. A callout bubble on the right says 'You can click on this' with an arrow pointing to a link in the log: 'you can find the results at <http://13.233.198.119:9000/dashboard?id=AnilKumar>'. At the bottom of the log, there is a box around the text 'Finished: SUCCESS' with an arrow pointing to it from the right, and another arrow pointing to the text 'Job execution was successful'.

Now, finally open the sonarqube server to see the report. On clicking on above shown link or directly open sonarqube URL. The page looks like as shown below.

The screenshot shows the SonarQube web interface. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and a search bar. A message at the top left says, "There's an update available for your SonarQube instance. Please update to make sure you benefit from the latest security and bug fixes." Below the navigation, there's a sidebar titled "Filters" with sections for "Quality Gate" (Passed: 1, Failed: 0) and "Reliability" (A rating: 1, B rating: 0, C rating: 0, D rating: 0). The main content area shows "1 project(s)" named "AnilKumar" with a "Passed" status. A callout bubble points to the project name with the text "Click on". Below the project name, there are metrics: Bugs (0), Vulnerabilities (0), Hotspots Reviewed (-), Code Smells (0), Coverage (-), Duplications (0.0%), and Lines (14, HTML, XML). The status bar at the bottom right indicates "Last analysis: 3 minutes ago".

Click on project name to see detailed information about bugs, vulnerabilities, code smells, code coverage and duplications.