

WEB APPLICATION CI/CD PROCESS (UBUNTU 22.04).

To deploy the web application in tomcat server using Jenkins. In parallel code quality analysis by using SonarQube server and delivering the build artifacts to Jfrog server follow the below mentioned procedure.

Prerequisites are

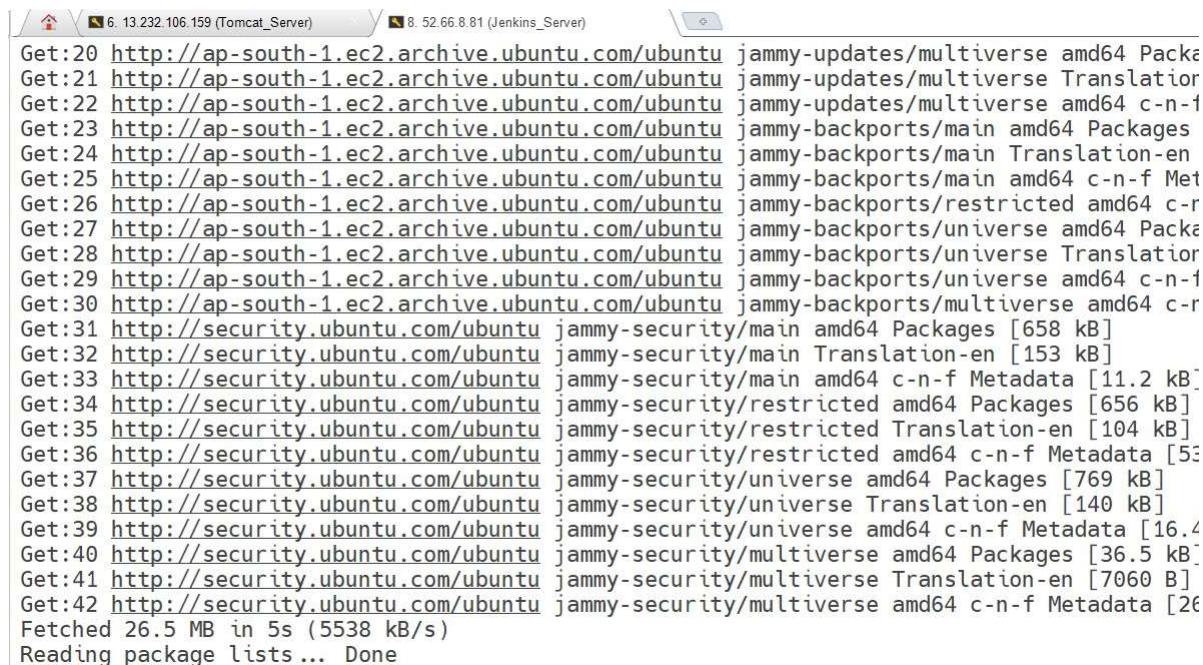
- 1) Jenkins Server. (t2.Micro – Instance1)**
- 2) Tomcat Server. (t2.Micro – Instance2)**
- 3) SonarQube Server. (t2.Medium – Instance3)**
- 4) Jfrog Server. (t2.Medium – Instance4)**

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.



```
[13:23:106.159 (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-1
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 Metadata
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-1
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
```

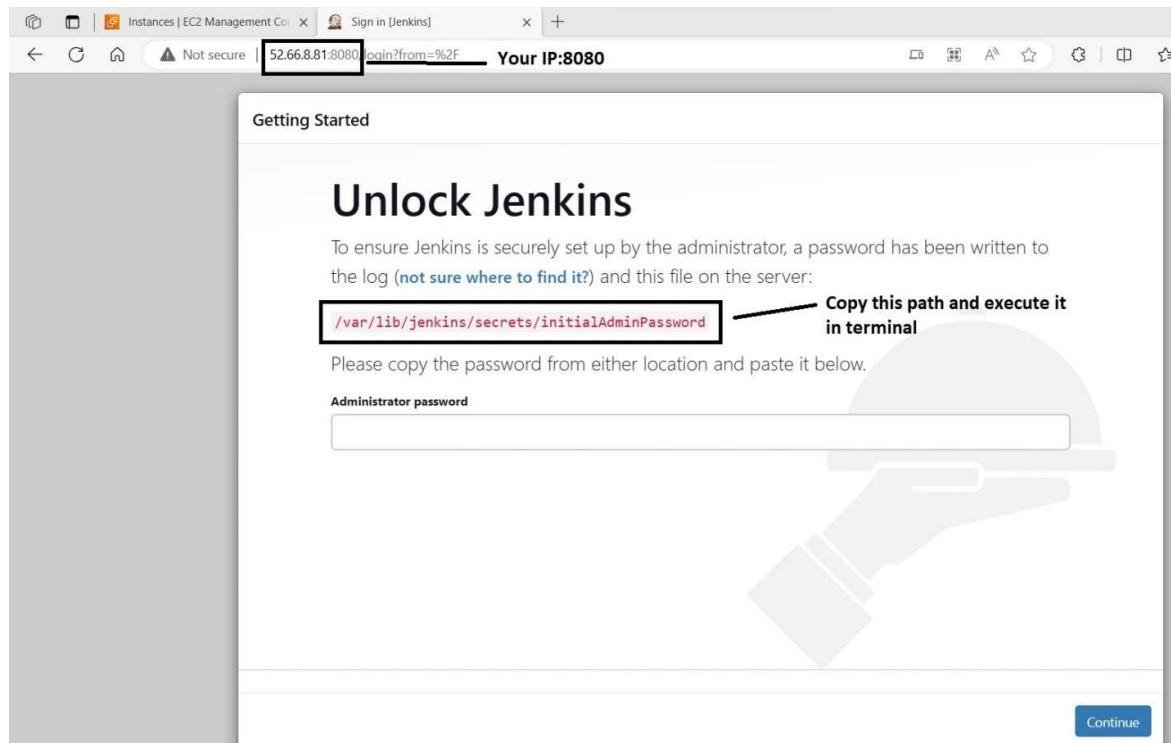
After executing the above commands, the following screen will appear.

```
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.



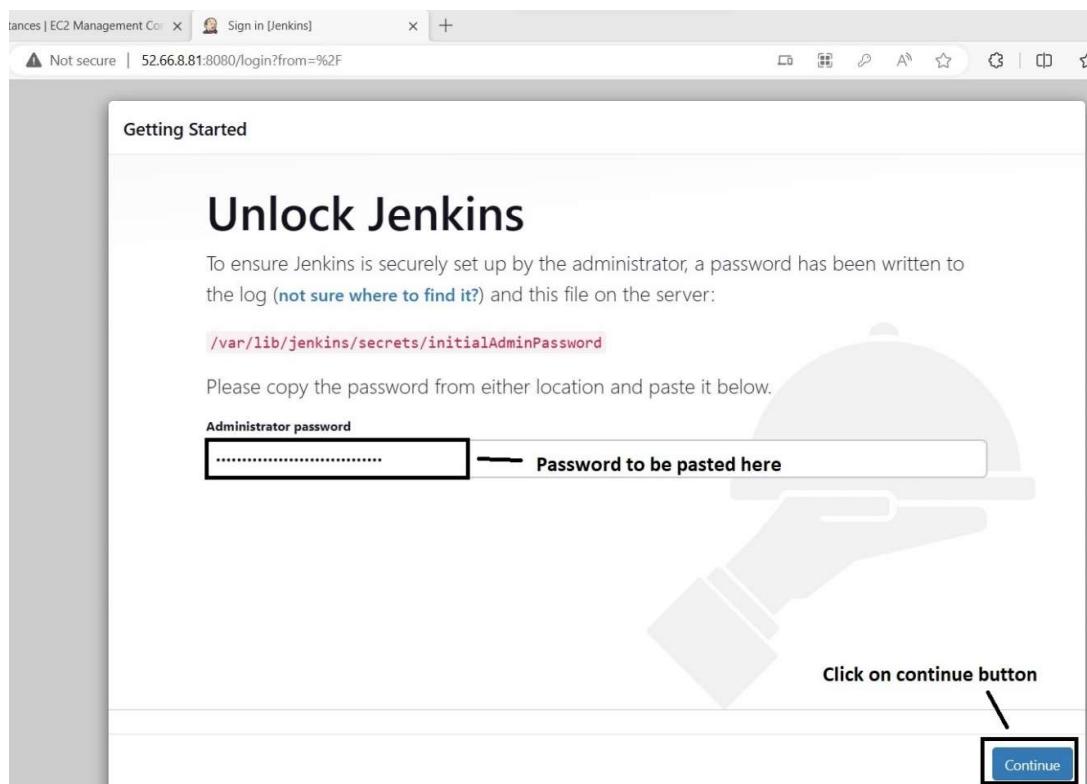
Then you have to copy that given path and execute the following commands in the terminal for initial admin password.

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

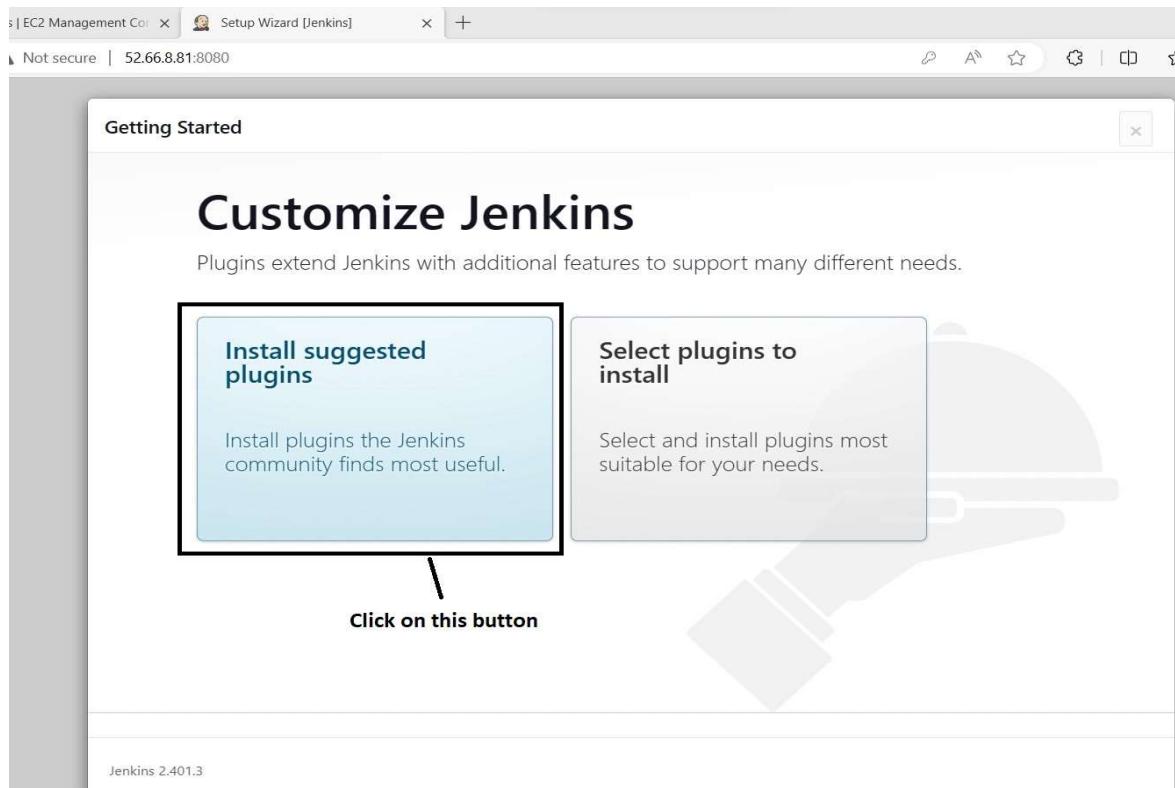
The following output message will be displayed as shown below.

The terminal window shows the command 'sudo cat /var/lib/jenkins/secrets/initialAdminPassword' being run. The output is '005728519dce4300bff9905ce11d2a4e'. A callout points to this output with the text 'Password to be copied and paste it in browser'.

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



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



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

The screenshot shows the Jenkins 'Getting Started' page. At the top, there's a header with tabs like 'Management Center' and 'Setup Wizard [Jenkins]'. Below the header, it says 'secure | 52.66.8.81:8080'. The main area has a title 'Getting Started' and a sub-section 'Plugins installation under progress'. A progress bar is shown below the sub-section. A table lists various Jenkins plugins. The first row of the table is highlighted with green checkmarks in all columns. The second row has a red box around the 'Email Extension' column. The third row has a red box around the 'GitHub Branch Source' column. The fourth row has a red box around the 'SSH Build Agents' column. The fifth row has a red box around the 'LDAP' column. To the right of the table, a sidebar lists 'Ionicons API', 'Folders', 'OWASP Markup Formatter', and '** - required dependency'. The bottom right corner of the sidebar also has a red box.

Plugins installation under progress, shown in below screenshot.

This screenshot is similar to the previous one, showing the Jenkins 'Getting Started' page. It features the 'Getting Started' title and 'Plugins installation under progress' section. The table of plugins is identical to the previous screenshot. A red box highlights the 'Email Extension' plugin in the second row. Another red box highlights the 'GitHub Branch Source' plugin in the third row. A callout arrow points from the text 'Installed Plugin' to the 'Credentials Binding' plugin in the fourth row, which has a green checkmark in its status column. Another callout arrow points from the text 'To be installed plugin' to the 'Email Extension' plugin in the second row, which has a red box around it. The sidebar on the right side of the table is identical to the previous screenshot, listing 'Ionicons API', 'Folders', 'OWASP Markup Formatter', and '** - required dependency'.

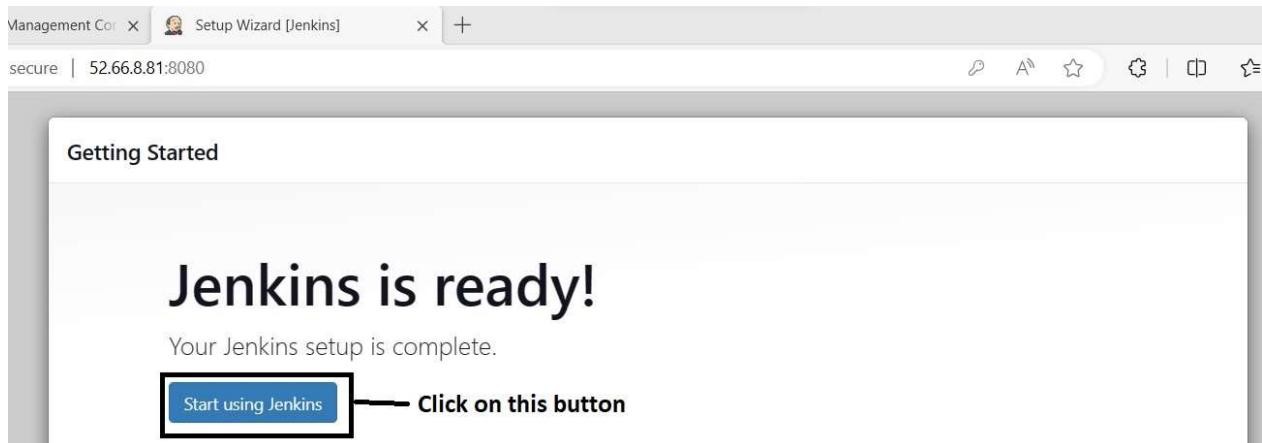
After successful installation of plugins, the following page will appear.

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'.

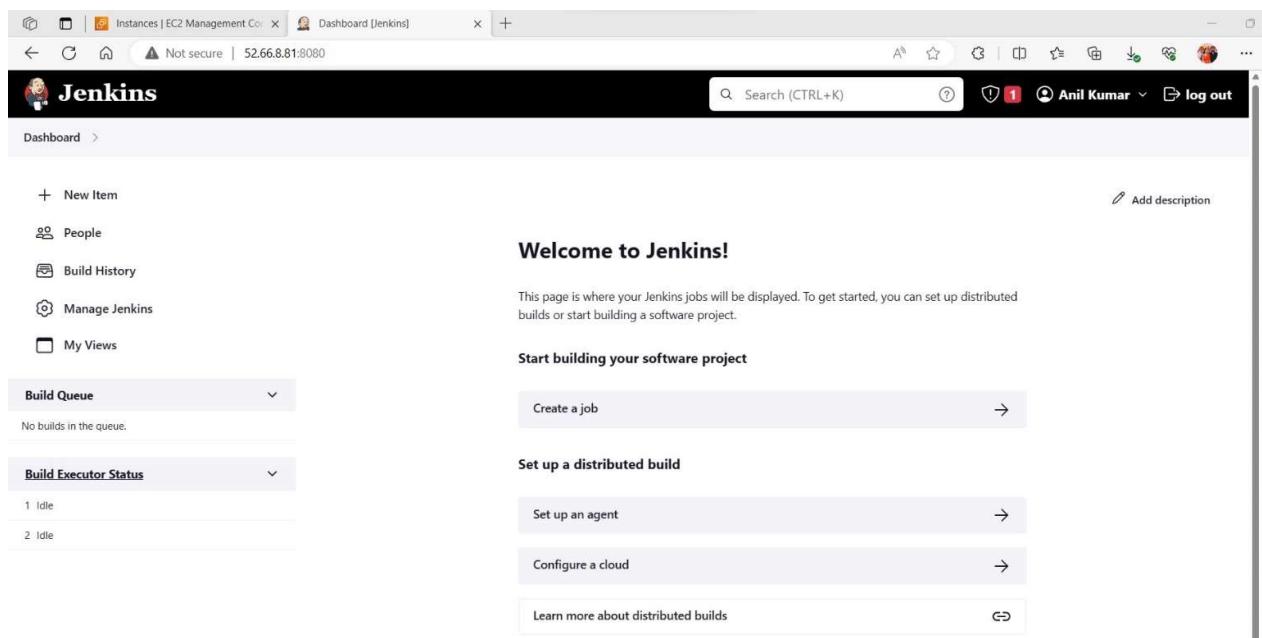
After, clicking on ‘Save and Continue’ button, the following page will appear.

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

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



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



Finally, jenkins setup was completed successfully.

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

Step by Step Procedure For Tomcat Server Setup.

First execute the following command in terminal.

sudo apt-get update

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

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

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

sudo apt-get install tomcat9 tomcat9-admin -y

After, executing the above command tomcat9 will be installed.

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

sudo systemctl status tomcat9

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

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

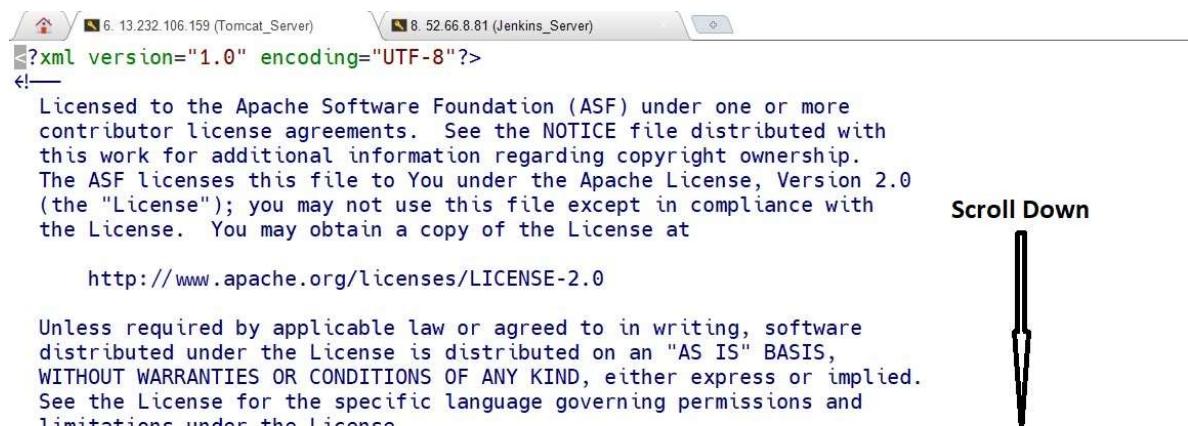
Aug 11 03:27:33 TomcatServer tomcat9[7231]: Deployment of deployment descriptor [/etc/tomcat9/Catalina/localhost/host-man
Aug 11 03:27:33 TomcatServer tomcat9[7231]: Deploying deployment descriptor [/etc/tomcat9/Catalina/localhost/manager.xml]
Aug 11 03:27:33 TomcatServer tomcat9[7231]: The path attribute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

Tomcat server installation completed successfully.

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

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

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



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

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

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

After scrolling down, the following mentioned content will appear.

```
<!--
<role rolename="tomcat"/>
<role rolename="role1"/>
<user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
<user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
<user username="role1" password="<must-be-changed>" roles="role1"/>
-->
We have to paste the content here, before </tomcat-users> tag
</tomcat-users>
```

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

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

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

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

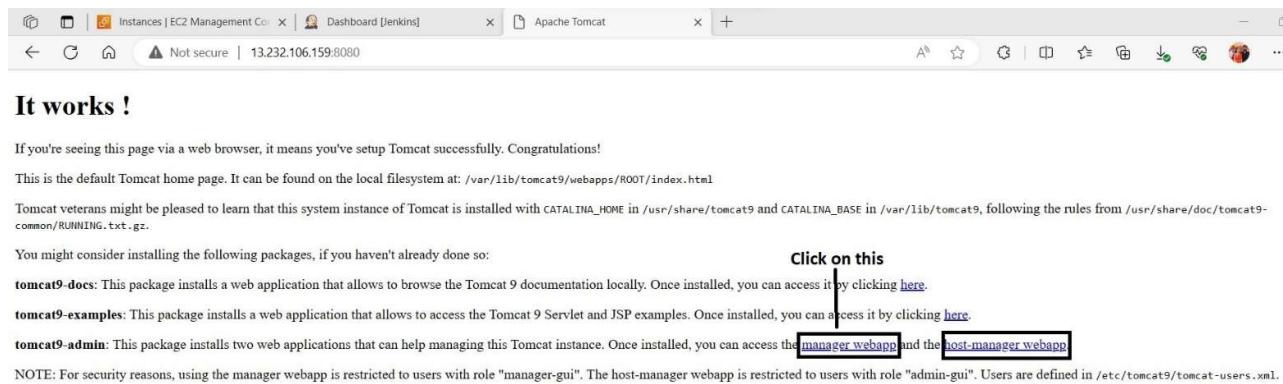
```
<!-- user name= both password= must be changed roles= tomcat /-->
<user username="both" password="" roles="tomcat,role1"/>
<user username="role1" password="" roles="role1"/>
→
<role rolename="admin-gui"/>
<role rolename="admin-script"/>
<role rolename="manager-gui"/>
<role rolename="manager-script"/>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

Content pasted

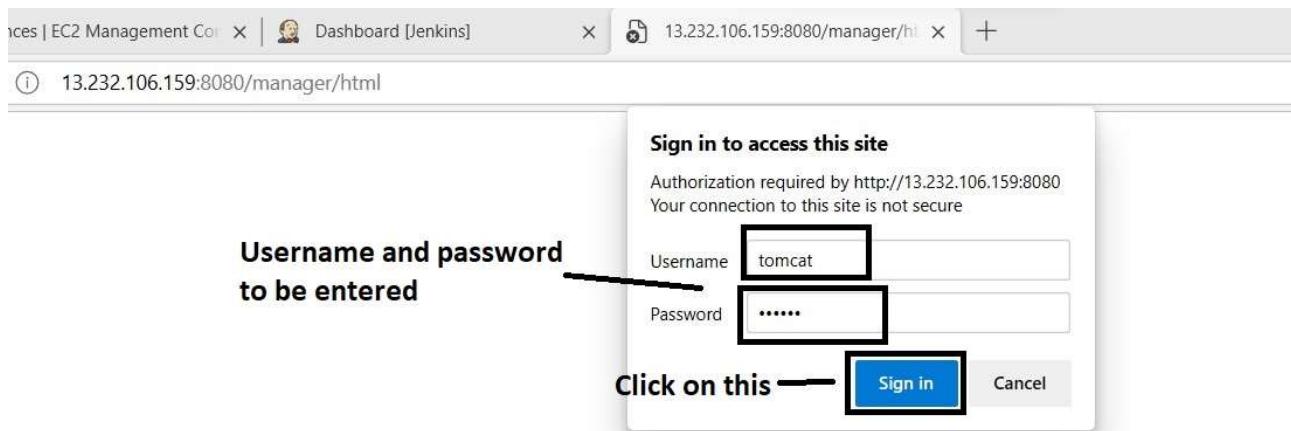
Now, enter the following URL in your browser.

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, a message box says 'Message: OK'. The main area has a title 'Tomcat Web Application Manager'. It features a 'Manager' section with tabs for 'List Applications', 'HTML Manager Help', 'Manager Help', and 'Server Status'. A 'Deploy' section allows for deploying a directory or WAR file with fields for 'Context Path', 'Version (for parallel deployment)', 'XML Configuration file path', and 'WAR or Directory path', followed by a 'Deploy' button. A cartoon cat icon is visible in the top left corner.

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

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

Step by step process for Integrating jenkins and Tomcat server

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

The screenshot shows the Jenkins dashboard. The URL in the browser bar is '52.66.8.81:8080'. The main header says 'Jenkins'. On the left, there's a sidebar with links: '+ New Item', 'People', 'Build History', 'Manage Jenkins' (which is highlighted with a red box and a callout 'Click on'), and 'My Views'. In the center, it says 'Welcome to Jenkins!' with a sub-instruction: 'This page is where your Jenkins jobs will be displayed or start building a software project.' At the bottom, there's a link 'Start building your software project'. At the very bottom, it says 'Build Queue' and 'Page 12 of 76'.

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 nodes and clouds). A prominent 'Plugins' link is highlighted with a box and the instruction 'Click on'. At the top right, there are buttons for 'Set up agent', 'Set up cloud', and 'Dismiss'.

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

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

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

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

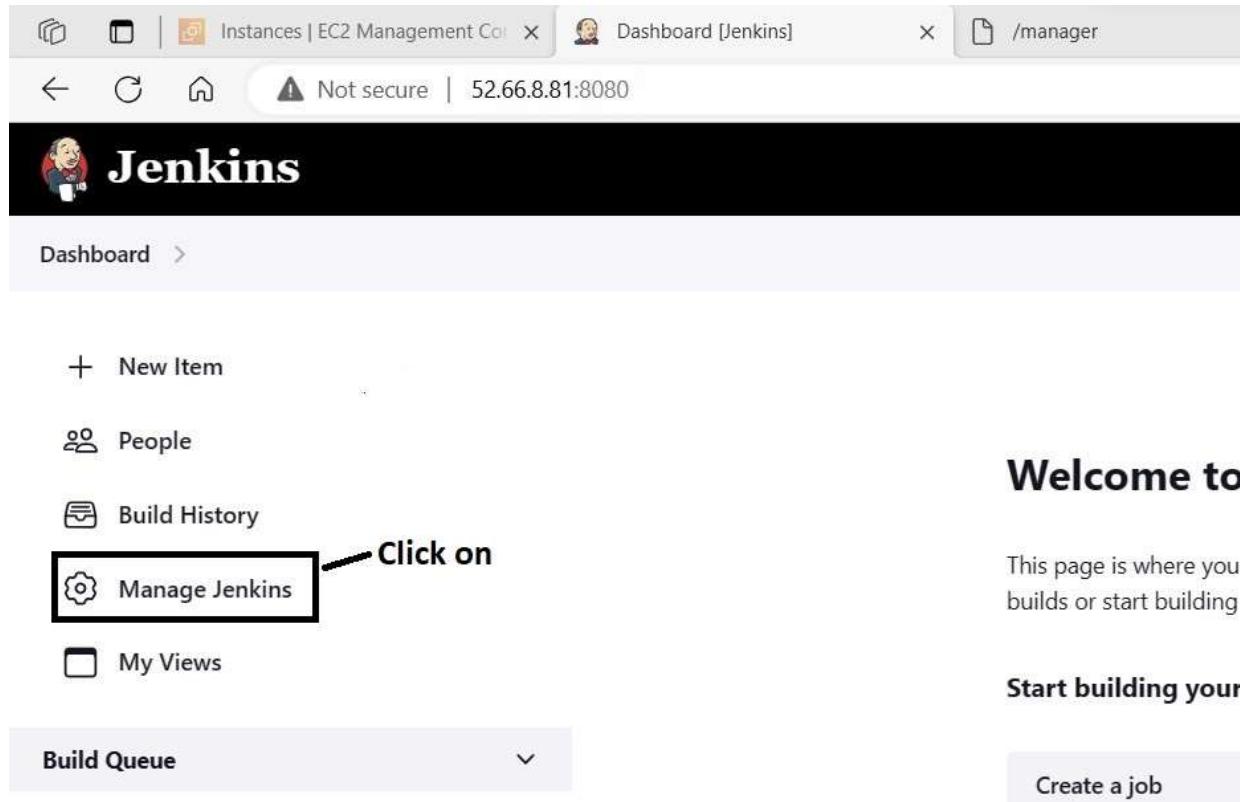
Note:- Deploy to container plugin is required to integrate jenkins and tomcat.

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

The screenshot shows the Jenkins 'Download progress' page. The 'Dashboard' link in the top navigation bar is highlighted with a black box and a callout arrow pointing to it with the text 'Click on'. The 'Available plugins' link in the sidebar is also highlighted with a black box and a callout arrow pointing to it with the text 'Click on'. In the main area, the 'Download progress' section is titled 'Download progress'. It shows a list of plugins under 'Preparation': Ionicons API, Folders, OWASP Markup Formatter, bouncycastle API, Instance Identity, JavaBeans Activation Framework (JAF) API, and javaMail API. To the right of the list, a column shows the status of each plugin: Success, Success, Success, Success, Success, Success, and Success. A callout arrow points to this column with the text 'Scroll down'. Another callout arrow points to the status column with the text 'We can see the status of plugin installation'.

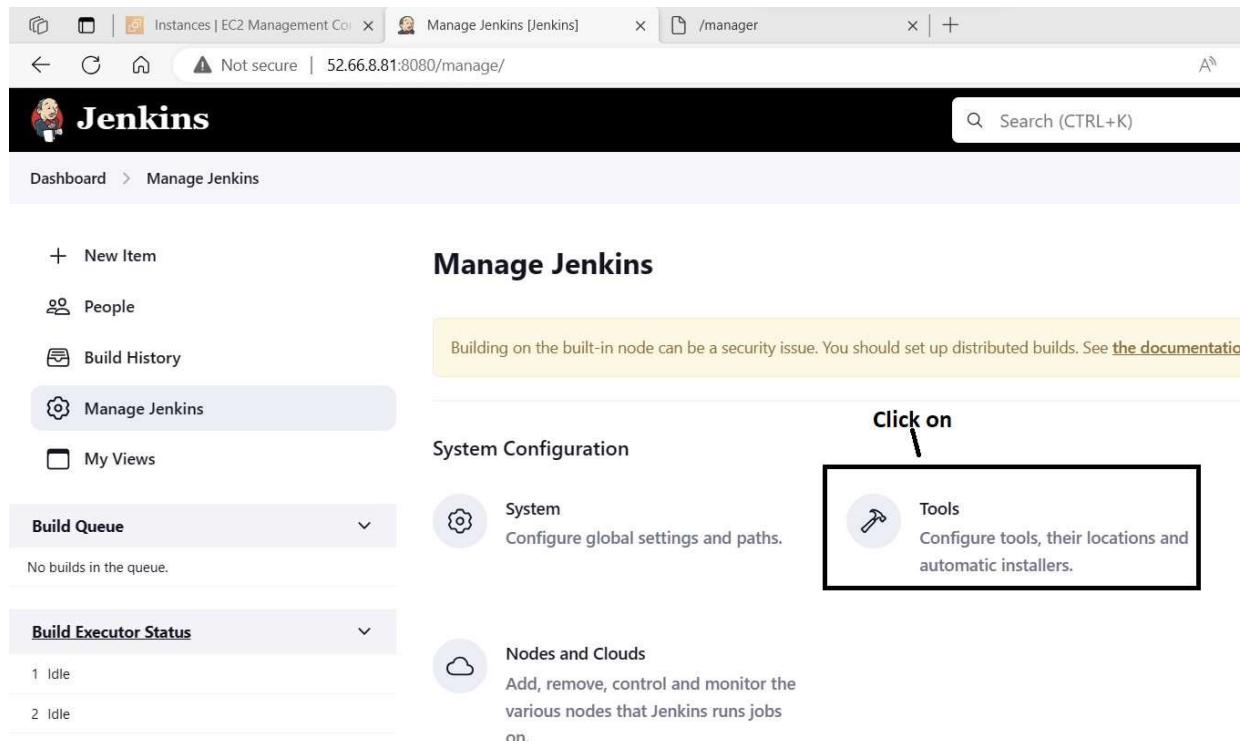
At the bottom, we can see plugin installation status.

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



The screenshot shows the Jenkins dashboard. On the left, there is a sidebar with links: '+ New Item', 'People', 'Build History', 'Manage Jenkins' (which is highlighted with a black border and a callout arrow pointing to it), and 'My Views'. On the right, there is a main content area with the heading 'Welcome to Jenkins'. It says 'This page is where you build or start building' and 'Start building your first job'. There is a 'Create a job' button at the bottom right. A callout arrow points to the 'Manage Jenkins' link in the sidebar.

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



The screenshot shows the 'Manage Jenkins' page. On the left, there is a sidebar with links: '+ New Item', 'People', 'Build History', 'Manage Jenkins' (which is highlighted with a black border and a callout arrow pointing to it), and 'My Views'. Below the sidebar is a 'Build Queue' dropdown showing 'No builds in the queue.' On the right, there is a 'System Configuration' section with three items: 'System' (Configure global settings and paths.), 'Tools' (Configure tools, their locations and automatic installers.) (which is highlighted with a black border and a callout 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 Con...', 'Tools [Jenkins]', and '/manager'. Below the header, there's a Jenkins logo and the word 'Jenkins'. A navigation bar shows 'Dashboard > Manage Jenkins > 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 these sections, 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 a callout arrow pointing to it, with the text 'Click on' next to it. Below the button 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 a callout arrow pointing to it, with the text 'You can give, your own name to identify the maven' next to it. The 'Install automatically' checkbox is checked. Below it, the 'Install from Apache' section is expanded, showing the 'Version' field with '3.9.4' selected and highlighted with a black box and a callout arrow pointing to it, with the text 'You can select, supported version.' next to it. At the bottom of the dialog are 'Add Maven', 'Save', and 'Apply' buttons. The 'Save' button is highlighted with a black box and a callout arrow pointing to it, with the text 'Click on' next to it.

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 note 'You can define your own name' and a 'Required field' note. A large callout box highlights the 'Freestyle project' option, which is described as the central feature of Jenkins, combining any SCM with any build system. Other options shown are 'Pipeline' (orchestrating long-running activities), 'Multi-configuration project' (suitable for multiple environments), and 'Folder' (creating containers for nested items). At the bottom, there's an 'OK' button and a 'Multibranch Pipeline' link. A callout points to the 'Freestyle project' section with the text 'Select ——'.

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

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

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

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

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

The screenshot shows the Jenkins configuration interface for a Git repository. On the left, there's a sidebar with several tabs: General, Source Code Management (which is selected and highlighted in grey), Build Triggers, Build Environment (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 an annotation 'Paste your repository URL' next to it. Below it is a 'Credentials' section with a dropdown set to '- none -'. In the 'Branches to build' section, there's a 'Branch Specifier' input field containing '/main', with an annotation 'Based on your repository branch' next to it. At the bottom right of the main panel, there's a 'Add Repository' button.

<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. The repository name 'AnilKumar' is displayed with a 'Public' badge. Below the repository name, there's a summary showing 'main', '1 branch', and '0 tags'. A 'Main Branch' button is also present. The main content area lists the files in the 'main' branch: 'mannem302 Update README.md' (committed 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 'Build Environment' tab is active. On the right, there's a 'Build Steps' section with a 'Add build step ▾' button highlighted with a callout pointing to it. Below it is a 'Post-build Actions' section with a 'Add post-build action ▾' button. At the bottom are 'Save' and 'Apply' buttons.

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

The screenshot shows the 'Build Steps' configuration page. The 'Add build step ▾' button is clicked, opening a dropdown menu. The 'Invoke top-level Maven targets' option is highlighted with a callout pointing to it. Other visible options include 'Execute Windows batch command', 'Execute shell', 'Invoke Ant', 'Invoke Gradle script', '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 sidebar on the left shows 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps' (selected), and 'Post-build Actions'. A callout points to the 'Build Steps' option with the text 'Select this option'. A dropdown menu titled 'Filter' is open, listing 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 and has a callout pointing to it with the text 'Select this option'), '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'. At the bottom of the dropdown is an 'Add post-build action ▾' button. Below the dropdown 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 from the text 'Click on'. There is also a checkbox for 'Deploy on failure' and a button '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 on the left: 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 from the text 'Select this option'. The 'Add Container ▾' button at the bottom is also highlighted with a blue rectangle.

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 'Containers' section with a 'Tomcat 9.x Remote' configuration. Under 'Credentials', there is a dropdown menu with the option '- none -' selected. A cursor arrow points to the 'Add' button below the dropdown. The text 'Click on dropdown option' is overlaid next to the dropdown menu.

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

The screenshot shows the same 'Containers' section with the 'Tomcat 9.x Remote' configuration. The dropdown menu now shows a list of users, with 'tomcat/***** (tomcat_user)' selected. A cursor arrow points to this selected item, and the text 'Select this user' is overlaid next to it.

After selecting that user, the following option will appear.

The screenshot shows a 'Configure' screen for a Jenkins job. On the left, a sidebar lists 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'. 'Post-build Actions' is currently selected. The main area displays the 'Tomcat 9.x Remote' configuration with the user 'tomcat/***** (tomcat_user)' selected in the dropdown. Below it, the 'Tomcat URL' is set to 'http://13.232.106.159:8080/'. There is an 'Advanced' dropdown and a checkbox for 'Deploy on failure'. At the bottom, there are 'Save' and 'Apply' buttons. A cursor arrow points to the 'Save' button, and the text 'First Click on' is above it. Another cursor arrow points to the 'Apply' button, and the text 'Second Click on' is above it.

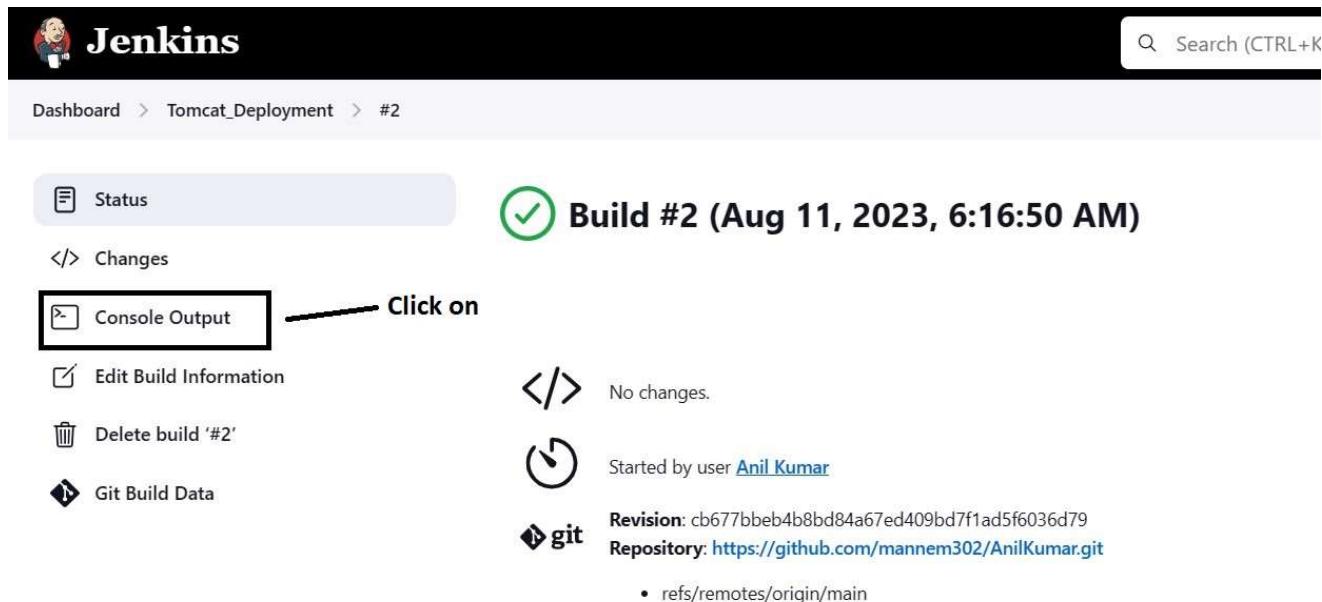
After clicking on ‘Apply’ and ‘Save’ buttons the following page will appear.

The screenshot shows the Jenkins interface for the 'Tomcat_Deployment' project. At the top, there are several tabs: 'Instances | EC2 Management', 'Tomcat_Deployment [Jenkins]', and '/manager'. Below the tabs, the Jenkins logo and the word 'Jenkins' are displayed. A search bar is on the right. The main title is 'Project Tomcat_Deployment'. On the left, there's a sidebar with links: '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.

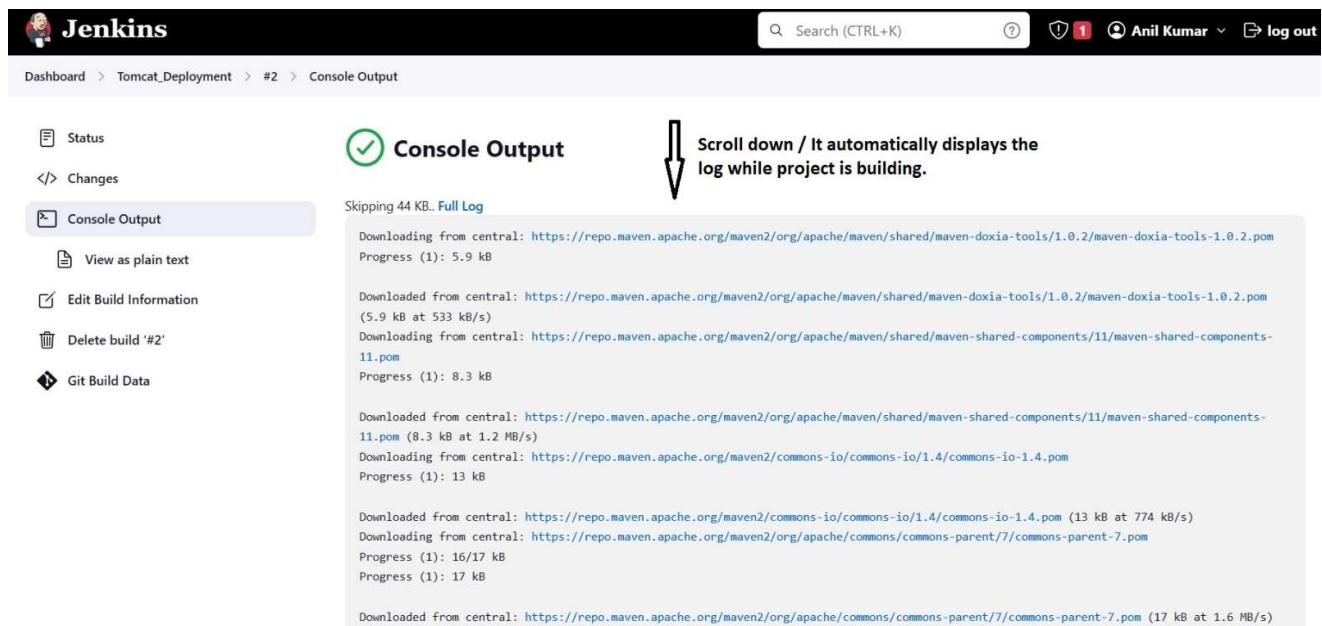
This screenshot is similar to the previous one, showing the Jenkins interface for the 'Tomcat_Deployment' project. The 'Build Now' button in the sidebar is highlighted with a red box and a callout 'Click on'. In the 'Build History' section, the first build entry is shown with a red box around it, and a callout 'Click on' points to the build number '#1'. The build details show the date and time: '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 that build, the following page will appear.



The screenshot shows the Jenkins interface for a build named 'Tomcat_Deployment' (Build #2). The 'Console Output' button is highlighted with a black border and a callout arrow pointing to it from the text 'Click on'. The main content area displays the build status: 'No changes.' and 'Started by user Anil Kumar'. It also shows the revision and repository details: 'Revision: cb677bbeb4b8bd84a67ed409bd7f1ad5f6036d79' and 'Repository: https://github.com/mannem302/AnilKumar.git'. A note indicates that the log is automatically displayed while the project is building.

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



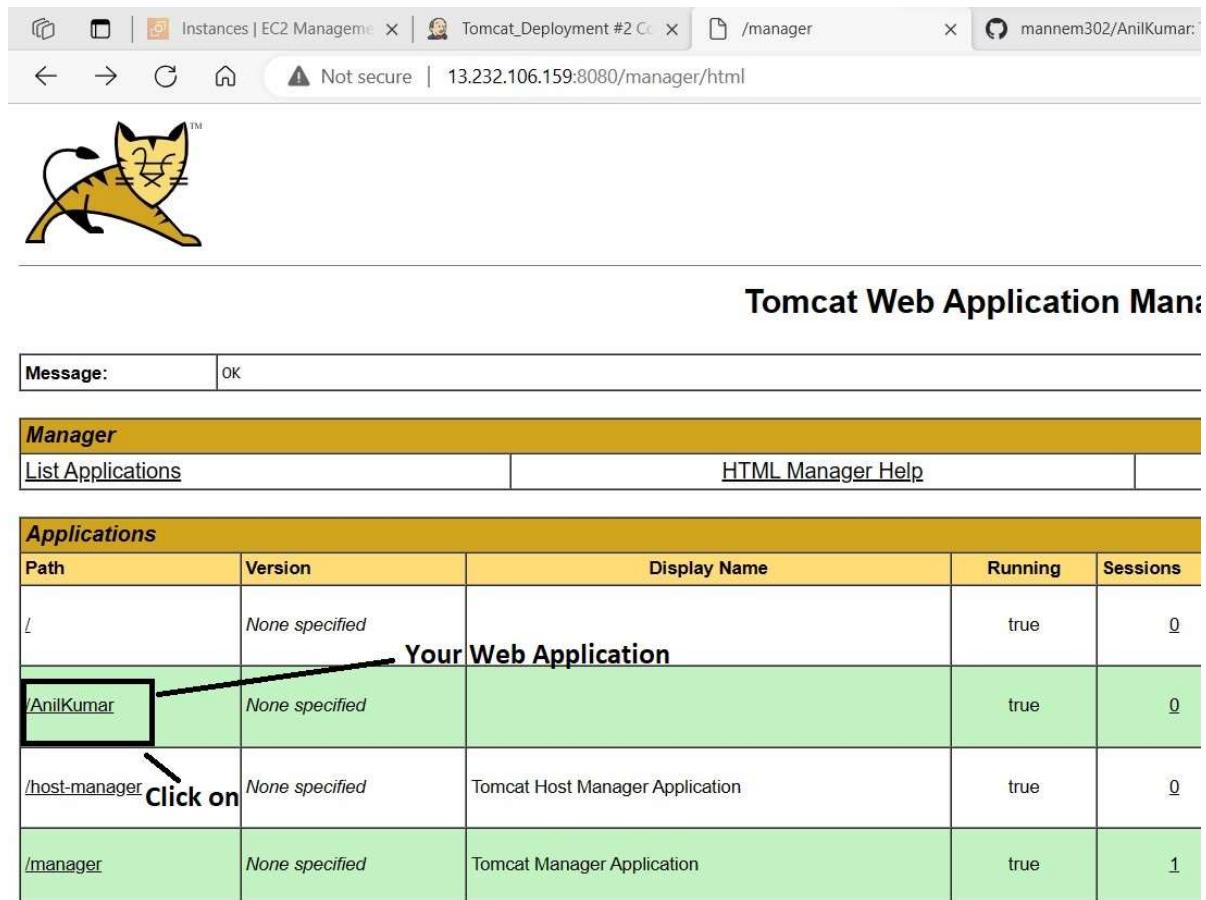
The screenshot shows the Jenkins 'Console Output' page for the same build. The 'Console Output' button is again highlighted with a black border and a callout arrow pointing to it from the text 'Click on'. The main content area shows the log output, which includes progress messages for downloading various Maven dependencies from central repositories. A note indicates that the log is automatically displayed while the project is building.

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 URL `13.232.106.159:8080/manager/html`. The title bar says "Tomcat Web Application Manager". The page displays a table of deployed applications:

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

A callout arrow points from the text "Click on" to the row containing the path `/AnilKumar`.

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+7-1~20.04.2) ...  
Setting up ubuntu-mono (19.04-0ubuntu3) ...  
Setting up openjdk-17-jdk:amd64 (17.0.8+7-1~20.04.2) ...  
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 Displays Java Version  
openjdk 17.0.8 2023-07-18  
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]
```

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/
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/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.

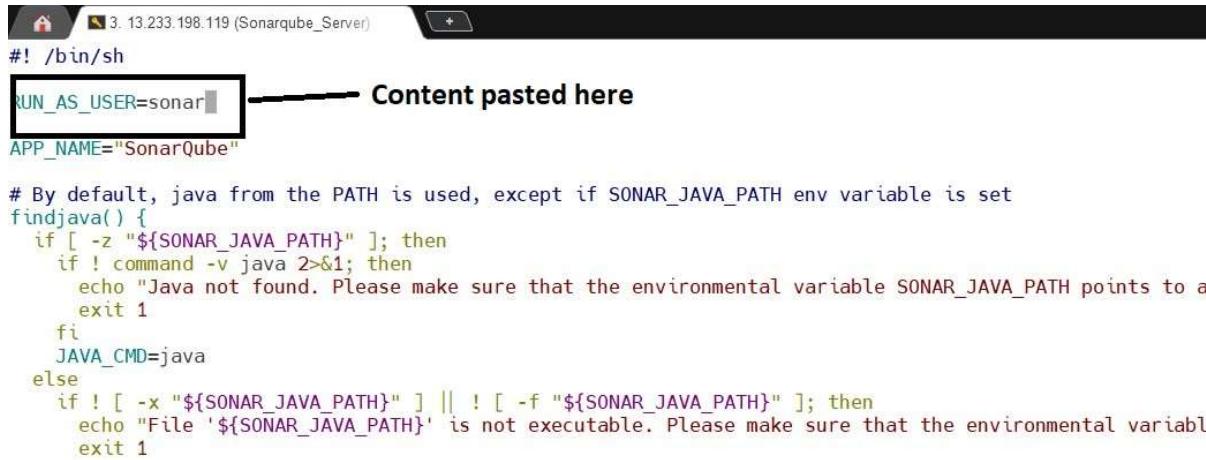
RUN_AS_USER=sonar



```
#!/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 "${SONAR_JAVA_PATH}" ]; then
        if ! command -v java 2>&1; then
            echo "Java not found. Please make sure that the environmental variable SONAR_JAVA_PATH points to a Java e"
            exit 1
        fi
        JAVA_CMD=java
    else
        if ! [ -x "${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"
            exit 1
        fi
        JAVA_CMD="${SONAR_JAVA_PATH}"
    fi
}
```

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



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

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

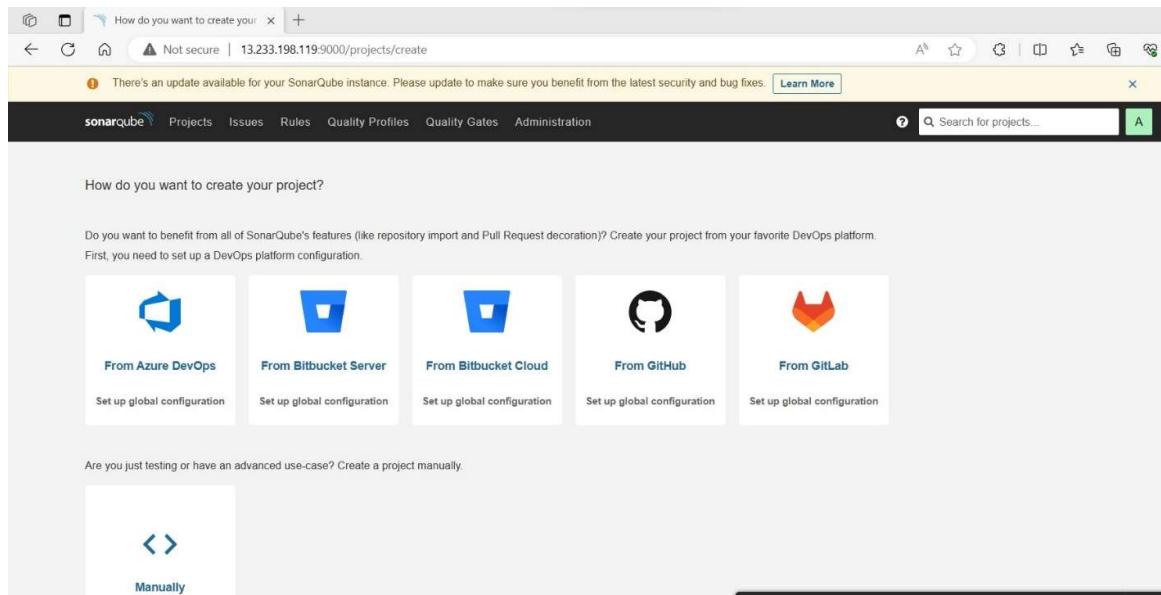
Confirm Password *

.....

Update

Click on

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



The screenshot shows the SonarQube interface for creating a new project. At the top, there is a message about an update available. Below the header, there is a navigation bar with links for 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. A search bar is also present. The main content area is titled 'How do you want to create your project?'. It provides several options for importing projects from different platforms:

- From Azure DevOps: Set up global configuration
- From Bitbucket Server: Set up global configuration
- From Bitbucket Cloud: Set up global configuration
- From GitHub: Set up global configuration
- From GitLab: Set up global configuration

Below these options, there is a link for 'Manually' which is highlighted with a blue border. The URL in the browser's address bar is 13.233.198.119:9000/projects/create.

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 table for Tomcat_Deployment.

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 the information was obtained 5 hours and 31 minutes ago.

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 a table of download progress. The first row shows 'SonarQube Scanner' with a green checkmark and the word 'Success'. A callout bubble points to this entry with the text 'Plugin successfully installed'. Below the table, there are two links: 'Go back to the top page' and 'Restart Jenkins when installation is complete and no jobs are running'.

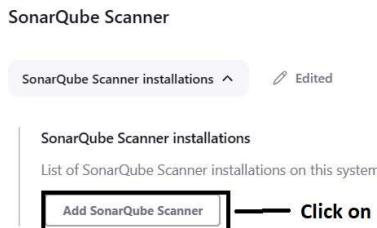
After clicking on ‘Manage Jenkins’ button. The following page will appear.

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.

A screenshot of a configuration page for adding a SonarQube Scanner installation. The page shows a form with fields for 'Name' (set to 'sonar_scanner') and 'Install automatically' (checked). A section titled 'Install from Maven Central' is expanded, showing a dropdown menu for 'Version' with 'SonarQube Scanner 5.0.1.3006' selected. A callout arrow points to the dropdown icon with the text 'Click on dropdown icon'.

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

A screenshot of the configuration page after selecting a version. The 'Version' field now contains 'SonarQube Scanner 4.8.0.2856'. A callout arrow points to this field with the text 'select this version'. At the bottom of the page, there are two buttons: 'Save' and 'Apply'. A callout arrow points to the 'Save' button with the text 'Second click on' and another arrow points to the 'Apply' button with the text 'First click on'.

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' (with a callout 'Click on System'), 'Nodes and Clouds', and 'Tools'. The 'System' link in the 'System Configuration' section is highlighted with a black rectangle and a callout pointing to it.

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

The screenshot shows the Jenkins System configuration page. At the top, it says 'System' and 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 large downward-pointing arrow on the right side of the page has a callout text: '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'. A note says: 'If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.' There is a checkbox labeled 'Environment variables' which is unchecked. Under 'SonarQube installations', it says 'List of SonarQube installations' and shows a button labeled '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 'SonarQube installations' configuration page. It has a header 'SonarQube installations' and a sub-header 'List of SonarQube installations'. A dashed box surrounds the 'Name' and 'Server URL' sections. In the 'Name' section, there is a text input field with 'sonarqube' typed into it, which is highlighted with a black rectangle and a callout arrow pointing to it with the text 'Click on'. In the 'Server URL' section, there is a text input field with 'Default is http://localhost:9000' and 'http://13.233.198.119:9000/' pasted into it, which is also highlighted with a black rectangle and a callout arrow pointing to it with the text 'Paste your sonarqube server URL'. Below these sections is a 'Server authentication token' section with a text input field containing a long token, which is highlighted with a black rectangle and a callout arrow pointing to it with the text 'Click on'. At the bottom left is a button labeled 'Advanced ▾'.

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

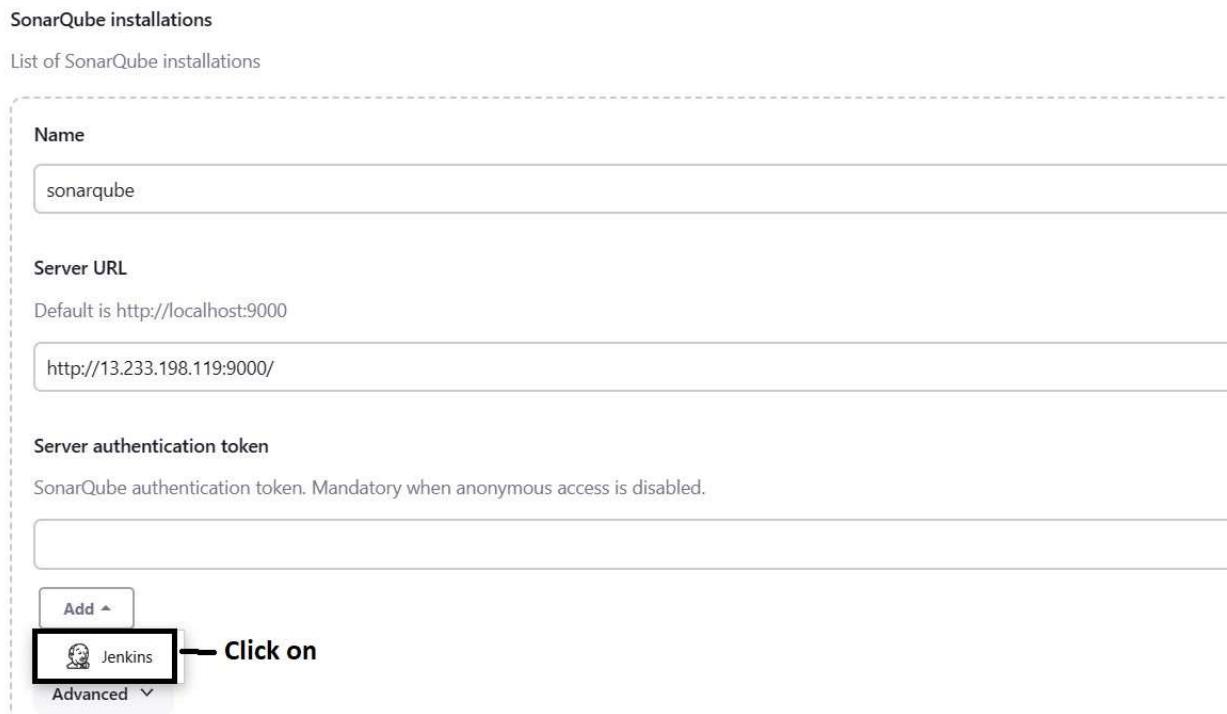
SonarQube authentication token. Mandatory when anonymous access is disabled.

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

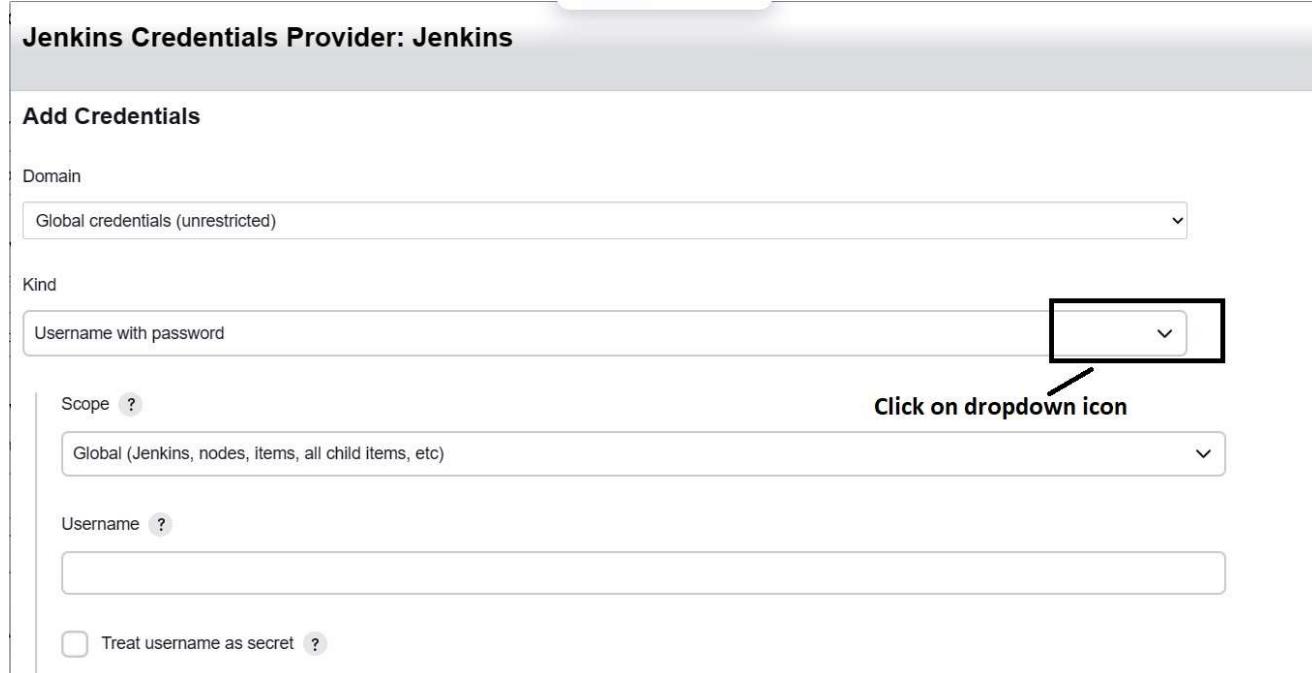
Scope ?

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 title bar says 'How do you want to create your project?'. A navigation bar at the top includes 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. An 'Administration' button is highlighted with a red box and a cursor arrow pointing to it. Below the navigation bar, a message says 'There's an update available for your SonarQube instance. Please update to make sure you benefit from the latest security and bug fixes.' A 'Learn More' link is next to it. The main content area asks 'How do you want to create your project?' and provides five options: 'From Azure DevOps', 'From Bitbucket Server', 'From Bitbucket Cloud', 'From GitHub', and 'From GitLab'. Each option has a 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 title bar says 'General Settings - Administration'. A navigation bar at the top includes 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. An 'Administration' button is highlighted with a blue bar. Below the navigation bar, a message says 'There's an update available for your SonarQube instance. Please update to make sure you benefit from the latest security and bug fixes.' A 'Learn More' link is next to it. The main content area is titled 'Administration' and shows a 'General Settings' section. It includes a 'Find in Settings' search bar and a sidebar with 'Analysis Scope', 'Authentication', and 'DevOps Platform Integrations'. A 'Security' tab is highlighted with a red box and a cursor arrow pointing to it. To the right, there is a 'Duplications' section with a 'Cross project duplication detection' toggle switch set to 'off' (default). A note states: 'DEPRECATED - By default, SonarQube detects duplications at project level. This means that a block duplicated on two different projects won't be reported. Setting this parameter to "true" allows to detect duplicates across projects. Note that activating this'.

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 creating and administering individual users. A search bar is available. A table lists users, showing one entry: "Administrator admin". The table includes columns for 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 "Click on" label. A "Create User" button is located at the top right of the user list area.

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, for 'sonar_token', has its 'Name' field highlighted with a red box and labeled 'Enter name'. The 'Type' field is labeled 'User'. The 'Project' field is labeled 'You can select or leave it default'. The 'Last use' column shows 'Never', 'Created' is 'August 18, 2023', and 'Expiration' is 'September 17, 2023'. A 'Done' button is at the bottom right of the modal.

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!' with a 'Copy' button next to the token value 'squ_3ec2b519a7595e0724cbd341b9739caf1e6c4d1e'. The table below shows the token details: 'sonar_token' (Name), 'User' (Type), 'Never' (Last use), 'August 18, 2023' (Created), and 'September 17, 2023' (Expiration). A 'Revoke' button is visible for the token row. A 'Done' button is 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

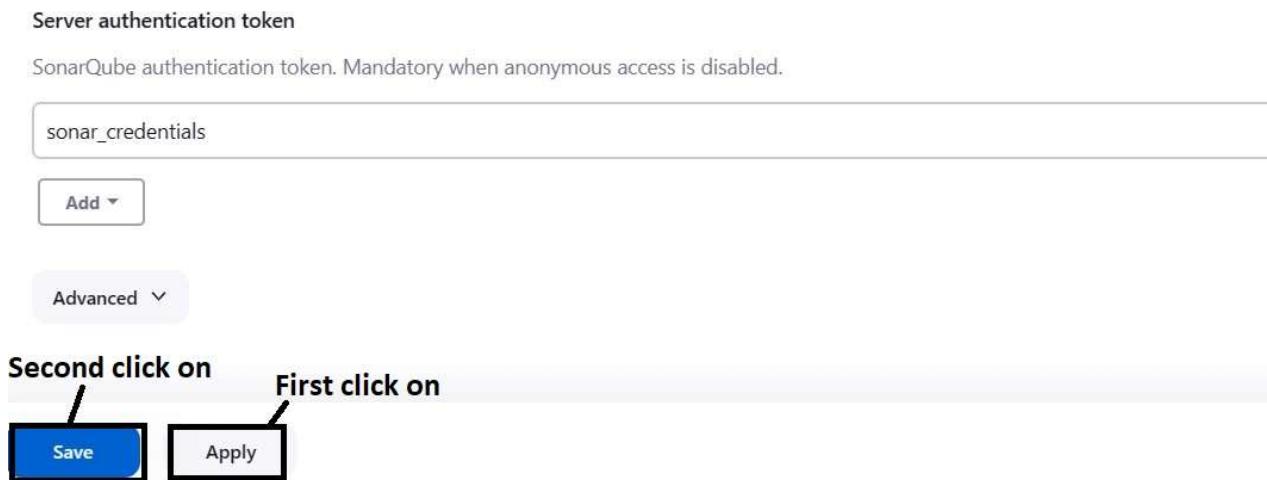
SonarQube authentication token. Mandatory when anonymous access is disabled.

sonar_credentials

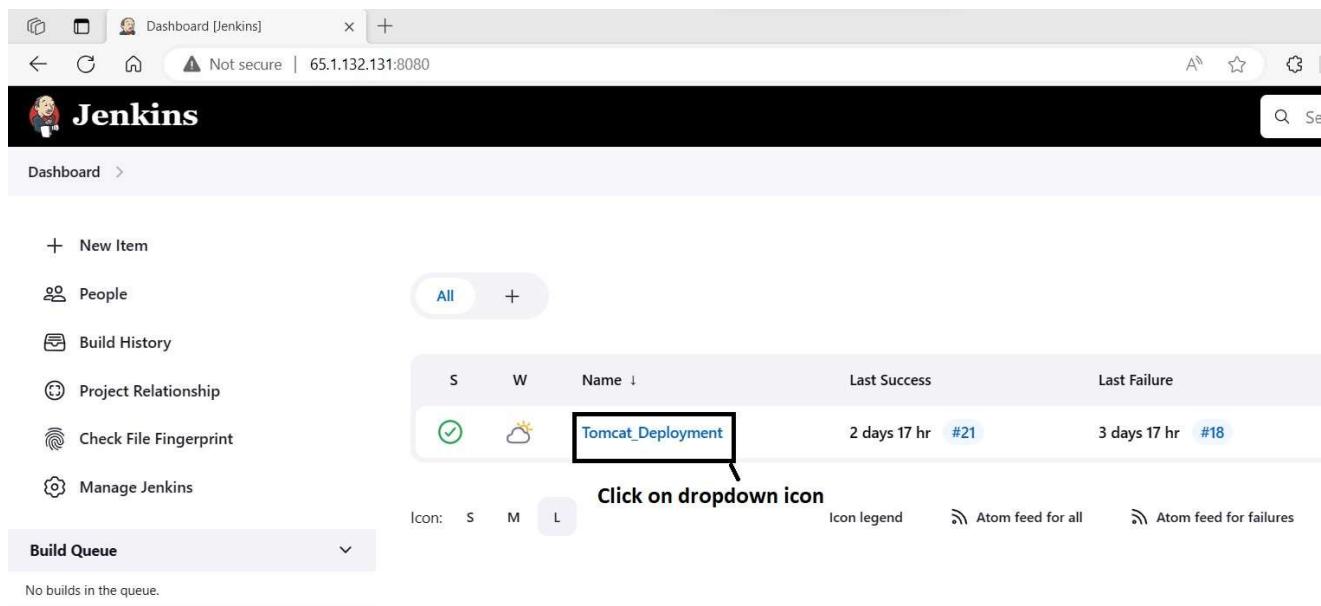
Add ▾

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 callout pointing to the dropdown icon)
 - 2 days 17 hr #21
 - 3 days 17 hr #18
- Job details for Tomcat_Deployment:
 - Icon: Green checkmark
 - Status: Success
 - 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 from the text 'Select this option'), Delete Project, and Rename. The dashboard also displays sections for New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, Build Queue (empty), and Build Executor Status (2 Idle).

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 from the text 'Click on'), and Post-build Actions. The General tab contains fields for Description (empty) and Advanced settings. Under Advanced, there are checkboxes for Discard old builds, GitHub project, This project is parameterized, Throttle builds, and Execute concurrent builds if necessary.

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

Configure

Build Steps

The screenshot shows the 'Configure' screen with the 'Build Steps' tab selected. On the left, there is a sidebar with icons for General, Source Code Management, Build Triggers, Build Environment, Build Steps (selected), and Post-build Actions. The main area is titled 'Build Steps' and contains sections for 'Invoke top-level Maven targets', 'Maven Version' (set to maven 3.9.4), 'Goals' (set to clean package), and an 'Advanced' dropdown. At the bottom, there is a button labeled 'Add build step' with a callout pointing to it.

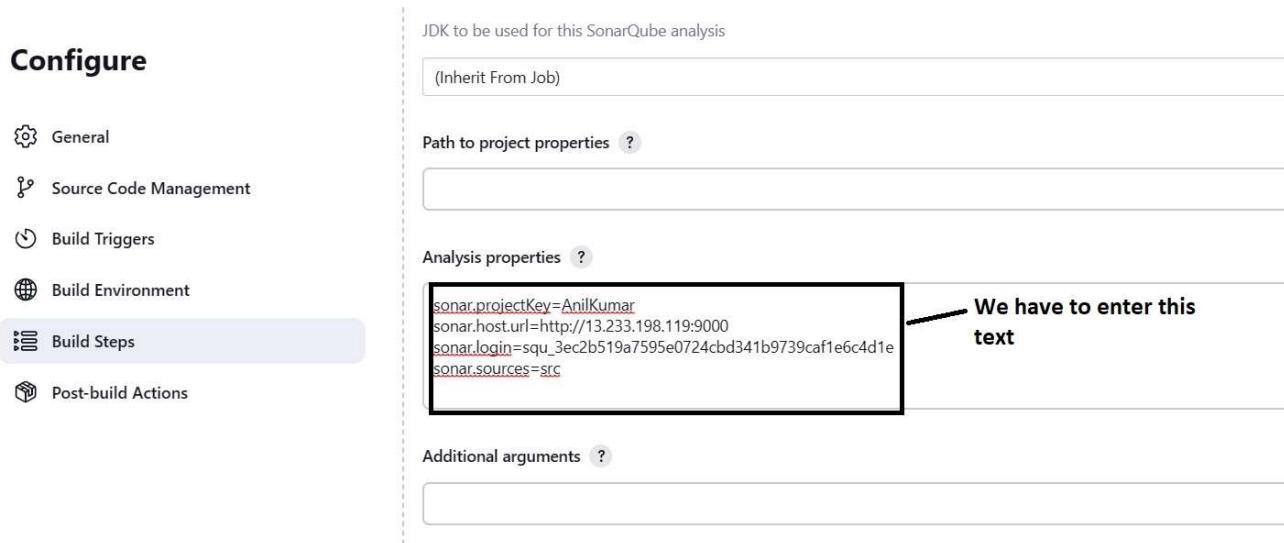
After clicking on ‘Add build step’ button, the following page will appear.

The screenshot shows the 'Build Steps' configuration screen. The 'Build Steps' tab is selected in the sidebar. In the main area, under 'Goals', 'clean package' is listed. Below it is an 'Advanced' dropdown. A callout points to the 'Add build step' button, which is highlighted with a blue border. A dropdown menu is open, showing options: 'Filter' (selected), 'Execute SonarQube Scanner' (highlighted with a black box), 'Execute Windows batch command', 'Execute shell', 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', and 'Run with timeout'. The 'Execute SonarQube Scanner' option is the one being selected.

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 'Configure' screen for the 'Tomcat_Deployment' job. On the left, there's a sidebar with links: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'Post-build Actions' link is highlighted with a light blue background. On the right, there are three main 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'. At the top, there's a header bar with the Jenkins logo and the URL 'Not secure | 65.1.132.131:8080/job/Tomcat_Deployment/'. Below the header, the page title is 'Project Tomcat_Deployment'. On the left, there's a sidebar with links: Status, Changes, Workspace, Build Now (which has a callout arrow pointing to it), Configure, Delete Project, SonarQube, and Rename. In the center, there's a 'SonarQube' section with the text 'Permalinks' and a list of build history items:

- 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.

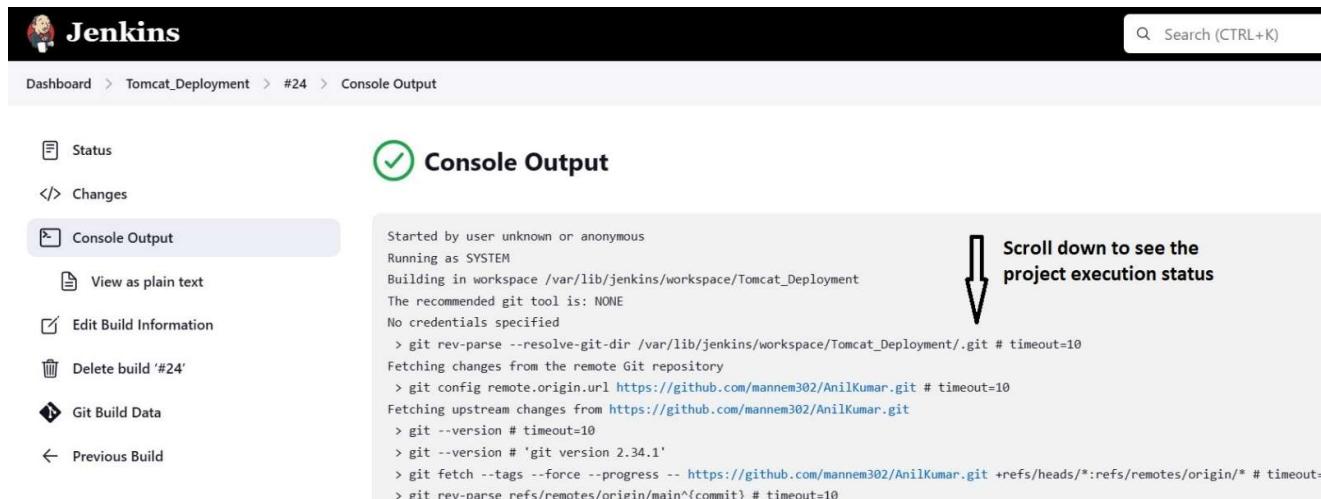
The screenshot shows the Jenkins dashboard for the project 'Tomcat_Depl'. On the left, there's a sidebar with various options: Status, Changes, Workspace, Build Now (which is highlighted with a red box), Configure, Delete Project, SonarQube, and Rename. Below this is a 'Build History' section with a 'trend' dropdown set to 'trend'. A search bar says 'Filter builds...'. Under 'Build History', a build entry for '#24 (Aug 18, 2023, 11:39 AM)' is shown, also highlighted with a red box. To the right, there's a 'SonarQube' icon and a 'Permalinks' section with a list of recent builds.

- Last build (#23), 8 min 38 sec ago
- Last stable build (#23), 8 min 38
- Last successful build (#23), 8 min
- Last failed build (#18), 3 days 18
- Last unsuccessful build (#18), 3 c
- Last completed build (#23), 8 mi

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

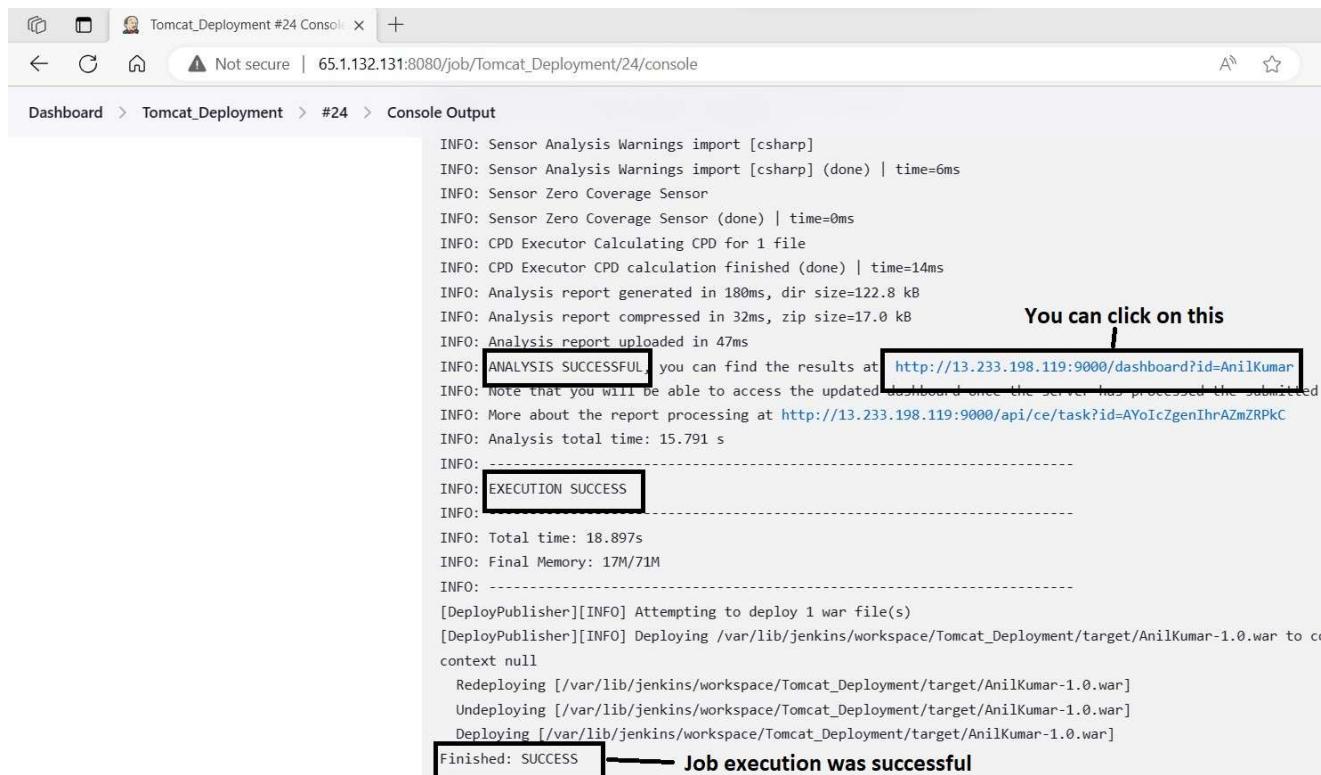
The screenshot shows the Jenkins build details for '#24 (Aug 18, 2023, 11:39:10 AM)'. The top navigation bar shows 'Dashboard > Tomcat_Depl > #24'. The left sidebar includes 'Status', 'Changes', 'Console Output' (which is highlighted with a red box and has an arrow pointing to it), 'Edit Build Information', 'Delete build #24', 'Git Build Data', and 'Previous Build'. The main content area displays the build status as green with a checkmark, the start time ('Aug 18, 2023, 11:39:10 AM'), and the commit information ('Revision: ef10ed3d341cde14ef26e8d7ebe1c9a6525de2eb' and 'Repository: https://github.com/mannem302/AnilKumar.git'). It also notes 'No changes.' and 'Started by anonymous user'.

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



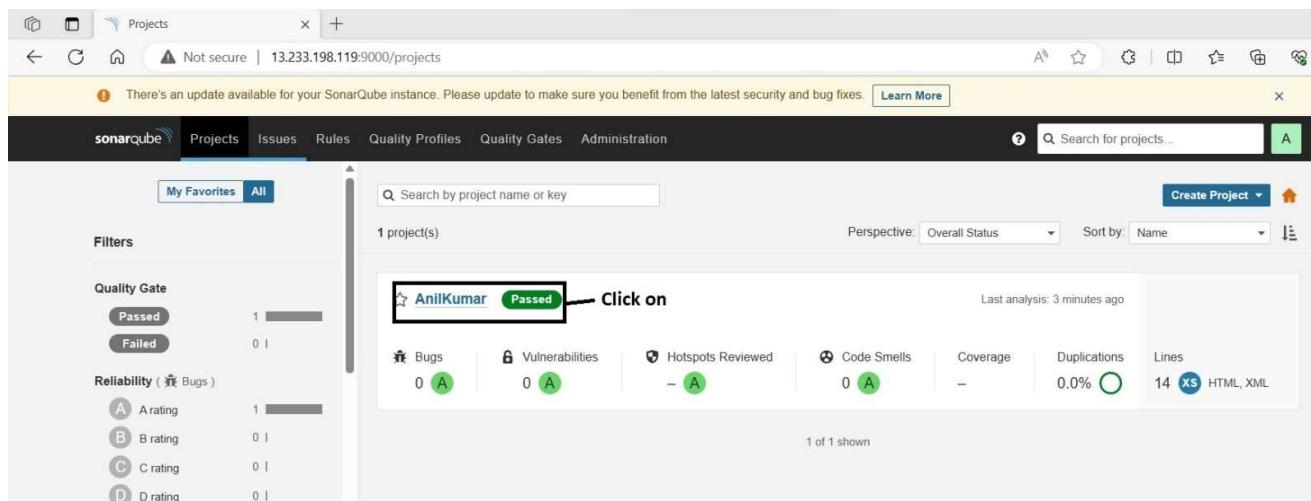
The screenshot shows the Jenkins interface for a build named 'Tomcat_Deployment #24'. The left sidebar has links for Status, Changes, Console Output (which is selected and highlighted in grey), View as plain text, Edit Build Information, Delete build '#24', Git Build Data, and Previous Build. The main 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, it says 'Not secure | 65.1.132.131:8080/job/Tomcat_Deployment/24/console'. The left sidebar is identical to the previous screenshot. The main content area shows the build log. 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 dashboard. At the top, there's a header with 'Projects' and a search bar. Below the header, a message says 'There's an update available for your SonarQube instance. Please update to make sure you benefit from the latest security and bug fixes.' A 'Learn More' button is next to it. The main area shows '1 project(s)' named 'AnilKumar' with a 'Passed' status. A callout box points to the project name with the text 'Click on'. To the right, it says 'Last analysis: 3 minutes ago'. Below the project name, there are several metrics: Bugs (0 A), Vulnerabilities (0 A), Hotspots Reviewed (- A), Code Smells (0 A), Coverage (-), Duplications (0.0%), and Lines (14 XS HTML, XML). On the left, there are filters for Quality Gate (Passed, Failed) and Reliability (A rating, B rating, C rating, D rating).

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

Jfrog Installation

For delivering build artifacts into Jfrog server. Initially we require Jfrog server and later we have to integrate it with jenkins.

Follow step by steps for Jfrog installation. Initially launch a t2.medium instance in AWS and then execute the following commands.

sudo apt-get update

After executing the above commands, the following screen will be displayed.

```
ubuntu@ip-10-0-7-39:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f Metadata [265 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [144 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/multiverse Translation-en [104 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 c-n-f Metadata [9136 B]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2888 kB]
```

Now we have to download the Jfrog installation files, follow the instructions shown below.

Execute the following commands

cd /opt

After executing above command, the following screen will display.

```
ubuntu@ip-10-0-7-39:/opt$ You have entered in opt folder
```

Now open your browser and in google search for '**Jfrog oss download**'

The page seems like as shown below.

A screenshot of a Google search results page. The search query is "jfrog oss download". The results show approximately 1,44,000 results. The top result is highlighted with a red box and an arrow pointing to it from the text "open this link". The result is from JFrog and links to "Download Artifactory OSS". Below this, another result from JFrog links to "Downloads & Open Source Solutions for Developers and ...".

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

A screenshot of the JFrog Artifactory OSS landing page. The URL is https://jfrog.com/community/download-artifactory-oss/. The page features a dark blue header with the JFrog logo and navigation links for Community, Downloads, Giving Back, and SuperFrogs. A green banner at the top says "SEE OUR LATEST AI/ML, DEVOPS, AND SECU...". The main content area has a dark blue background and displays the heading "JFROG ARTIFACTORY OPEN SOURCE FOR ARTIFACT LIFE-CYCLE MANAGEMENT". Below this, there is a paragraph of text and a "Get code source >" button.

[Get code source >](#) Click on this link

After clicking on ‘Get code Source’ link, the following page will appear.

Index of bintray-artifactory/org/artifactory/oss/jfrog-artifactory-oss

Name	Last modified	Size
..../		
6.17.4/	21-May-2021 15:08	-
6.18.5/	21-May-2021 15:22	-
6.19.3/	23-May-2021 10:29	-
6.20.4/	23-May-2021 16:57	-
6.21.1/	23-May-2021 10:24	-
6.22.2/	23-May-2021 10:32	-
6.23.13/	23-Feb-2021 08:04	-
6.23.15/	04-Apr-2021 17:29	-

Click on this link

Scroll down the page for required version

After clicking on the required version, the following page will be appeared.

Index of bintray-artifactory/org/artifactory/oss/jfrog-artifactory-oss/7.68.14

Name	Last modified	Size
..../		
jfrog-artifactory-oss-7.68.14-base.tar.gz	09-Oct-2023 15:27	109.80 KB
jfrog-artifactory-oss-7.68.14-darwin.tar.gz	09-Oct-2023 15:26	1.08 GB
jfrog-artifactory-oss-7.68.14-linux.tar.gz	09-Oct-2023 15:26	1.09 GB
jfrog-artifactory-oss-7.68.14-windows.zip	09-Oct-2023 15:26	1.09 GB

Artifactory Online Server

Now based on the operating system, we have to select the link. As shown below and copy the link and paste in Linux terminal.

Index of bintray-artifactory/org/artifactory/oss/jfrog-artifactory-oss/7.68.14

Name	Last modified	Size
..../		
jfrog-artifactory-oss-7.68.14-base.tar.gz	09-Oct-2023 15:27	109.80 KB
jfrog-artifactory-oss-7.68.14-darwin.tar.gz	09-Oct-2023 15:26	1.08 GB
jfrog-artifactory-oss-7.68.14-linux.tar.gz	09-Oct-2023 15:26	1.09 GB
jfrog-artifactory-oss-7.68.		

Right click on

Open link in new tab

Open link in new window

Open link in InPrivate window

Open link in split screen window

Open link as Profile 2

Save link as

Copy link

After copying the link, paste the link in Linux terminal followed by ‘**sudo wget**’ as shown below.

```
ubuntu@ip-10-0-7-39:/opt$ sudo wget https://releases.jfrog.io/artifactory/bintray-artifactory/org/artifactory/oss/jfrog-artifactory-oss/7.68.14/jfrog-artifactory-oss-7.68.14-linux.tar.gz
```

After pasting the link, press Enter button. The following page will appear.

```
ubuntu@ip-10-0-7-39:/opt$ sudo wget https://releases.jfrog.io/artifactory/bintray-artifactory/org/artifactory/oss/jfrog-artifactory-oss/7.68.14/jfrog-artifactory-oss-7.68.14-linux.tar.gz
--2023-10-13 07:27:51-- https://releases.jfrog.io/artifactory/bintray-artifactory/org/artifactory/oss/jfrog-artifactory-oss/7.68.14/jfrog-artifactory-oss-7.68.14-linux.tar.gz
Resolving releases.jfrog.io (releases.jfrog.io) ... 54.221.82.220, 50.17.28.229, 3.221.179.99
Connecting to releases.jfrog.io (releases.jfrog.io) |54.221.82.220|:443 ... connected.
HTTP request sent, awaiting response ... 302
Location: https://releases-cdn.jfrog.io/filestore/3c/3c101e4aefb09991acc435801b42d0fa02b7a4e0?response-content-type=application/x-gzip&response-content-disposition=attachment%3Bfilename=%3D%22jfrog-artifactory-oss-7.68.14-linux.tar.gz%22&x-jf-traceId=e7e2c4db63aa69816X-Artifactory-repositoryKey=bintray-artifactory-Artifactory-projectKey=default&x-Artifactory-artifactPath=org%2Fartifactory%2Foss%2Fjfrog-artifactory-oss%2F7.68.14%2Fjfrog-artifactory-oss-7.68.14-linux.tz_gz&x-Artifactory-username=anonymous&x-Artifactory-repoType=local&x-Artifactory-packageType=generic&Expires=16971821326SSignature=HKm9rn8-HTG5w9sgWc-OH42aDXrMCQDPKvG6Bh-NLpjidG8dXFtIw39dI0n9rvby3P9-TzmJ9x9itRXRDshyRvKuYzL7cnPm8kZ7i1103-9cpvhRev3AFTEGVrvV2f13XomIV-W-GcUtszFKOWzVbt0cpPaP8IqmHcNc70bZlwaeZdV4IGks8TJvOSkJ4RZLCVKnPnfVgpbPxoIDRhMq8n9yg4Nv0-y-uTBeo0f6QyRFydpF0hXL2gt9tByfIKjTycqZTeutRUteDjlHem09s74ZTOhs1jg8j1l2KNFBs10ZftkGbrOINh1Gmk-pi1A-Key-Pair-Id=APKAJ6NHFMVU3M6DPBA [following]
--2023-10-13 07:27:52-- https://releases-cdn.jfrog.io/filestore/3c/3c101e4aefb09991acc435801b42d0fa02b7a4e0?response-content-type=application/x-gzip&response-content-disposition=attachment%3Bfilename=%3D%22jfrog-artifactory-oss-7.68.14-linux.tar.gz%22&x-jf-traceId=e7e2c4db63aa69816X-Artifactory-repositoryKey=bintray-artifactory-Artifactory-projectKey=default&x-Artifactory-artifactPath=org%2Fartifactory%2Foss%2Fjfrog-artifactory-oss%2F7.68.14%2Fjfrog-artifactory-oss-7.68.14-linux.tar.gz&x-Artifactory-username=anonymous&x-Artifactory-repoType=local&x-Artifactory-packageType=generic&Expires=16971821326SSignature=HKm9rn8-HTG59sgWc-OH42aDXrR01yh-XWBMbipwK7lwaeZdV4IGks8TJvOSkJ4RZLCVKnPnfVgpbPxoIDRhMq8n9yg4Nv0-y-uTBeo0f6QyRFydpF0hXL2gt9tByfIKjTycqZTeutRUteDjlHem09s74ZTOhs1jg8j1l2KNFBs10ZftkGbrOINh1Gmk-pi1A-Key-Pair-Id=APKAJ6NHFMVU3M6DPBA
Resolving releases-cdn.jfrog.io (releases-cdn.jfrog.io) ... 18.66.53.5, 18.66.53.83, 18.66.53.18, ...
Connecting to releases-cdn.jfrog.io (releases-cdn.jfrog.io) |18.66.53.5|:443 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 1168062416 (1.1G) [application/x-gzip]
Saving to: 'jfrog-artifactory-oss-7.68.14-linux.tar.gz'

jfrog-artifactory-oss-7.68.14-linux.tar 100%[=====] 1.09G 116MB/s in 9.5s
2023-10-13 07:28:02 (117 MB/s) - 'jfrog-artifactory-oss-7.68.14-linux.tar.gz' saved [1168062416/1168062416]
```

File downloaded successfully

To un-tar the downloaded file, execute the following command.

sudo tar -xvf jfrog-artifactory-oss-7.68.14-linux.tar.gz and press enter.

After clicking on enter the following page will be displayed.

```
ubuntu@ip-10-0-7-39:/opt$ sudo tar -xvf jfrog-artifactory-oss-7.68.14-linux.tar.gz
artifactory-oss-7.68.14/app/bin/diagnostics/diagnosticsUtil
artifactory-oss-7.68.14/var/
artifactory-oss-7.68.14/var/etc/
artifactory-oss-7.68.14/app/
artifactory-oss-7.68.14/app/doc/
artifactory-oss-7.68.14/app/metadata/
artifactory-oss-7.68.14/app/metadata/bin/
artifactory-oss-7.68.14/app/frontend/
artifactory-oss-7.68.14/app/frontend/bin/
artifactory-oss-7.68.14/app/frontend/bin/client/
artifactory-oss-7.68.14/app/frontend/bin/client/dist/
artifactory-oss-7.68.14/app/frontend/bin/client/dist/flags/
artifactory-oss-7.68.14/app/frontend/bin/client/dist/fonts/
artifactory-oss-7.68.14/app/frontend/bin/client/dist/css/
artifactory-oss-7.68.14/app/frontend/bin/client/dist/workers/
artifactory-oss-7.68.14/app/frontend/bin/client/dist/dependencies/
artifactory-oss-7.68.14/app/frontend/bin/client/dist/js/
artifactory-oss-7.68.14/app/frontend/bin/client/dist/img/
artifactory-oss-7.68.14/app/frontend/bin/client/legacy_styles/
artifactory-oss-7.68.14/app/frontend/bin/client/legacy_styles/fonts/
artifactory-oss-7.68.14/app/frontend/bin/client/legacy_styles/fonts/OpenSans/
artifactory-oss-7.68.14/app/frontend/bin/client/legacy_styles/fonts/OpenSans/OpenSans-Italic/
artifactory-oss-7.68.14/app/frontend/bin/client/legacy_styles/fonts/OpenSans/OpenSans-BoldItalic/
artifactory-oss-7.68.14/app/frontend/bin/client/legacy_styles/fonts/OpenSans/OpenSans-ExtraBoldItalic/
artifactory-oss-7.68.14/app/frontend/bin/client/legacy_styles/fonts/OpenSans/OpenSans-Semibold/
artifactory-oss-7.68.14/app/frontend/bin/client/legacy_styles/fonts/OpenSans/OpenSans-SemiboldItalic/
artifactory-oss-7.68.14/app/frontend/bin/client/legacy_styles/fonts/OpenSans/OpenSans-Bold/
artifactory-oss-7.68.14/app/frontend/bin/client/legacy_styles/fonts/OpenSans/OpenSans-ExtraBold/
artifactory-oss-7.68.14/app/frontend/bin/client/legacy_styles/fonts/OpenSans/OpenSans-Light/
artifactory-oss-7.68.14/app/frontend/bin/client/legacy_styles/fonts/OpenSans/OpenSans-Regular/
artifactory-oss-7.68.14/app/frontend/bin/client/legacy_styles/fonts/OpenSans/OpenSans-LightItalic/
```

File Extraction u/p

After completion of extraction process, we can check it by using ‘ls’ command and for extracted file we have to change the ownership permissions. By executing the following commands.

```
ubuntu@ip-10-0-7-39:/opt$ ls  
artifactory-oss-7.68.14 jfrog-artifactory-oss-7.68.14-linux.tar.gz
```

sudo chown -R ubuntu:ubuntu artifactory-oss-7.68.14

After changing the ownership, execute the following command.

```
cd artifactory-oss-7.68.14/app/bin/
```

Now, you will enter into bin folder and now execute the following commands to start the jfrog server. (Running the shell script file)

```
./artifactory.sh
```

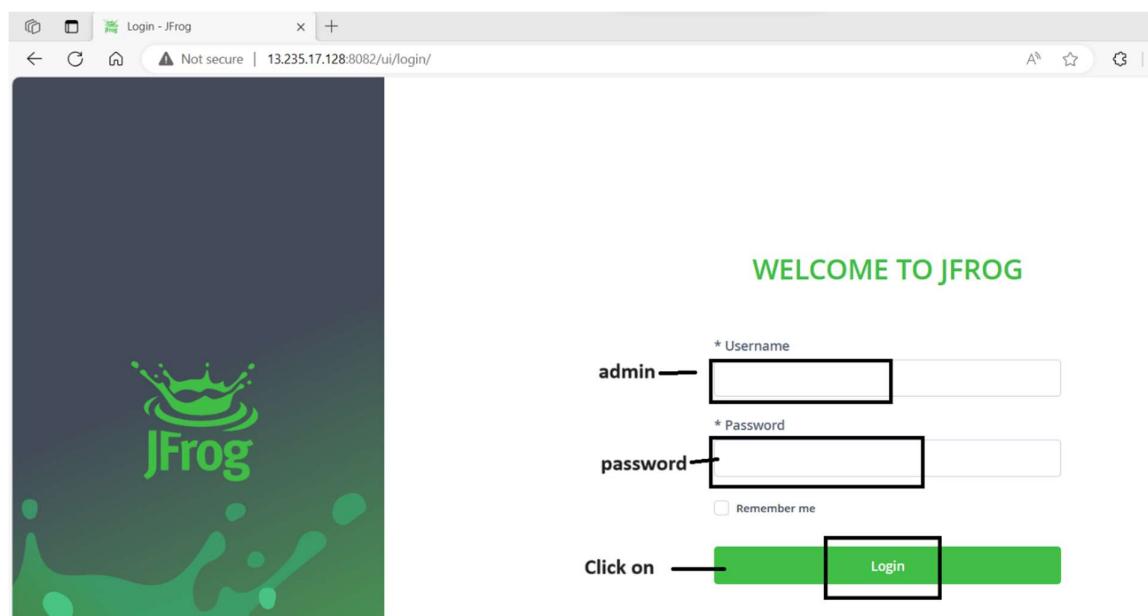
After executing the above shell script file, the following page will display.

```
ate at: /opt/artifactory-oss-7.68.14/var/etc/keys/ca.crt  
2023-10-13T07:53:29.958Z [jfrc] [INFO] [6e2a6bbfa0e1ca9e] [CertificateFileHandlerBase:176] [jf-access-task4] - Loading root certificate from database.  
2023-10-13T07:53:30.057Z [jfrc] [INFO] [6e2a6bbfa0e1ca9e] [CertificateFileHandlerBase:202] [jf-access-task4] - [ACCESS BOOTSTRAP] Saved new root certificate at: /opt/artifactory-oss-7.68.14/var/etc/keys/root.crt  
2023-10-13T07:53:32.247Z [jfrc] [INFO] [6e2a6bbfa0e1ca9e] [o.j.c.ConfigWrapperImpl:342] [e-watcher-notifier-1] - [Node ID: ip-10-0-7-39] detected local modify for config 'artifactory/config/artifactory.repository.config.latest.json'  
2023-10-13T07:53:43.315Z [jfrc] [INFO] [6efa8a6c35d88e4c] [a.e.EventsLogCleanUpService:79] [c35d88e4c|art-exec-5] - Starting cleanup of old events from event log  
2023-10-13T07:53:43.320Z [jfrc] [INFO] [6efa8a6c35d88e4c] [a.e.EventsLogCleanUpService:85] [c35d88e4c|art-exec-5] - deleteFrom=1697183592374  
2023-10-13T07:53:43.321Z [jfrc] [INFO] [6efa8a6c35d88e4c] [a.e.EventsLogCleanUpService:92] [c35d88e4c|art-exec-5] - eventsDeleteInterval=14400000  
2023-10-13T07:53:43.340Z [jfrc] [INFO] [6efa8a6c35d88e4c] [a.e.EventsLogCleanUpService:95] [c35d88e4c|art-exec-5] - maxBoundary.timestamp=1697183592374  
2023-10-13T07:53:43.341Z [jfrc] [INFO] [6efa8a6c35d88e4c] [a.e.EventsLogCleanUpService:96] [c35d88e4c|art-exec-5] - maxBoundary.deleteErrors=false  
2023-10-13T07:53:43.341Z [jfrc] [INFO] [6efa8a6c35d88e4c] [a.e.EventsLogCleanUpService:123] [c35d88e4c|art-exec-5] - Cleanup of old events from event log finished  
2023-10-13T07:53:43.372Z [jfrc] [INFO] [6efa8a6c35d88e4c] [onStatusStorageServiceImpl:124] [onitor-migration-job] - Inserting new migration status record 'MigrationStatus(identifier=event-table-repo-key-migration, started=1697183623344, finished=0, migrationInfoBlob=MarkerMigrationInfoBlob(serverId))'  
2023-10-13T07:53:43.386Z [jfrc] [INFO] [6efa8a6c35d88e4c] [entTableRepoKeyMigrationDao:84] [onitor-migration-job] - start execute events table repo key migration in the first time  
2023-10-13T07:53:48.408Z [jfrc] [INFO] [6efa8a6c35d88e4c] [onStatusStorageServiceImpl:137] [onitor-migration-job] - Migration for 'event-table-repo-key-migration' has finished.  
2023-10-13T07:53:48.414Z [jfrc] [INFO] [6efa8a6c35d88e4c] [ntTableRepoKeyMigrationDao:165] [onitor-migration-job] - successfully finished execute events table repo key migration
```

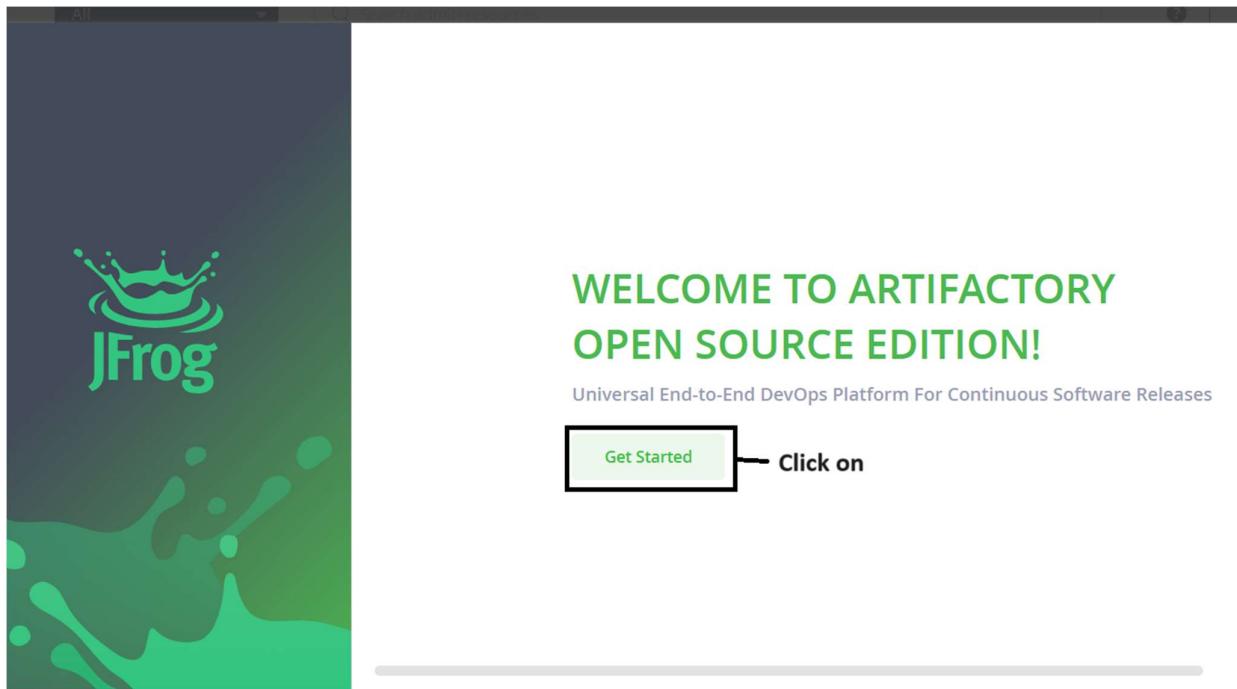
After successful installation, open the browser and enter the following URL.

http://Your_IP:8082

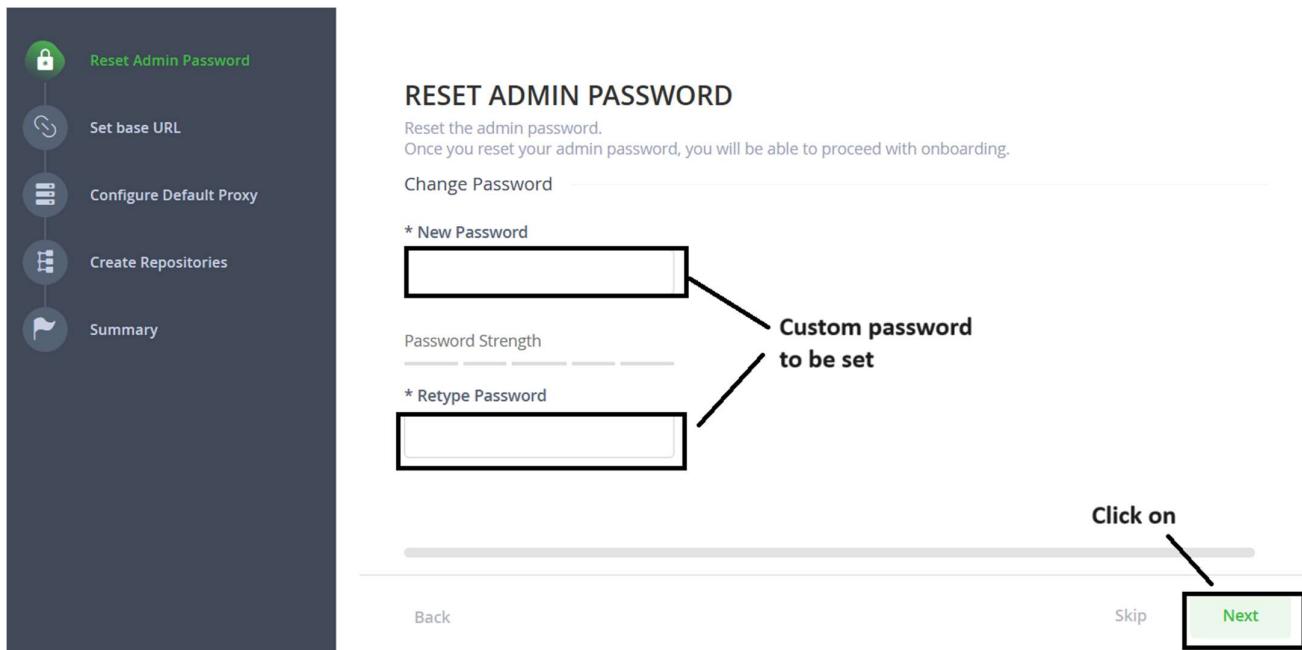
The following page will appear. Default username-**admin**, password- **password**



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



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



After clicking on ‘Next’ button, close the above shown screen.

After closing that screen, the screen looks like as shown below.

The screenshot shows the JFrog Artifactory Open Source administration interface. On the left, a sidebar menu includes 'Administration' (selected), 'Repositories', 'User Management', 'Authentication Providers', 'General', 'Proxies', 'Monitoring', and 'Artifactory'. The main content area features a 'Welcome To JFrog Platform' banner with the tagline 'Universal End-to-End DevOps Platform For Continuous Software Releases'. Below the banner is a 3D bar chart graphic and a section titled 'CREATE AND MANAGE YOUR REPOSITORY' with the sub-instruction 'Create a repository and manage access to your repository'. A green button labeled 'Create a Repository' is highlighted with a black arrow pointing to it from the text 'Click on'.

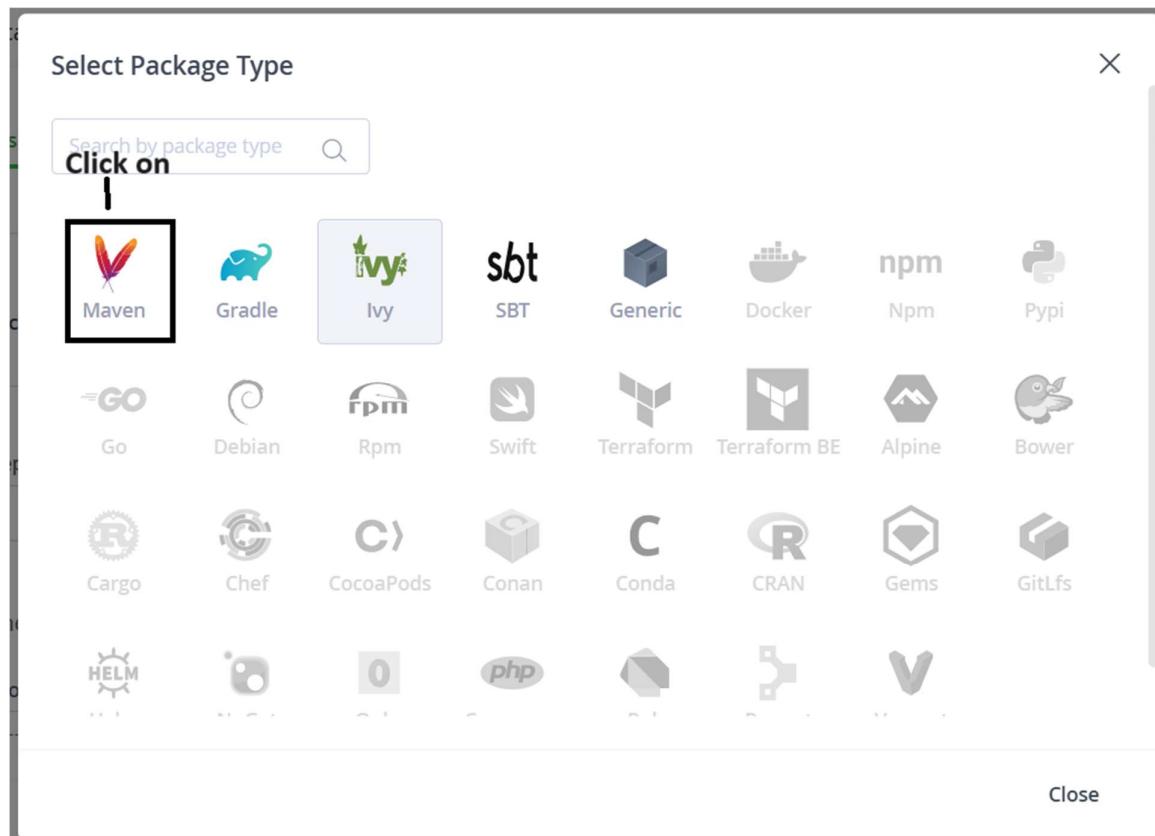
After clicking on ‘Create Repository’ button. The following page will appear.

The screenshot shows the 'Repositories' list page. The sidebar menu is identical to the previous one. The main content area displays a table titled 'Repositories' with three tabs: 'Local' (selected), 'Remote', and 'Virtual'. The table shows 1 Repository: 'example-repo-local' (Type: Generic). A green button labeled '+ Add Repositories' is highlighted with a black arrow pointing to it from the text 'Click on'.

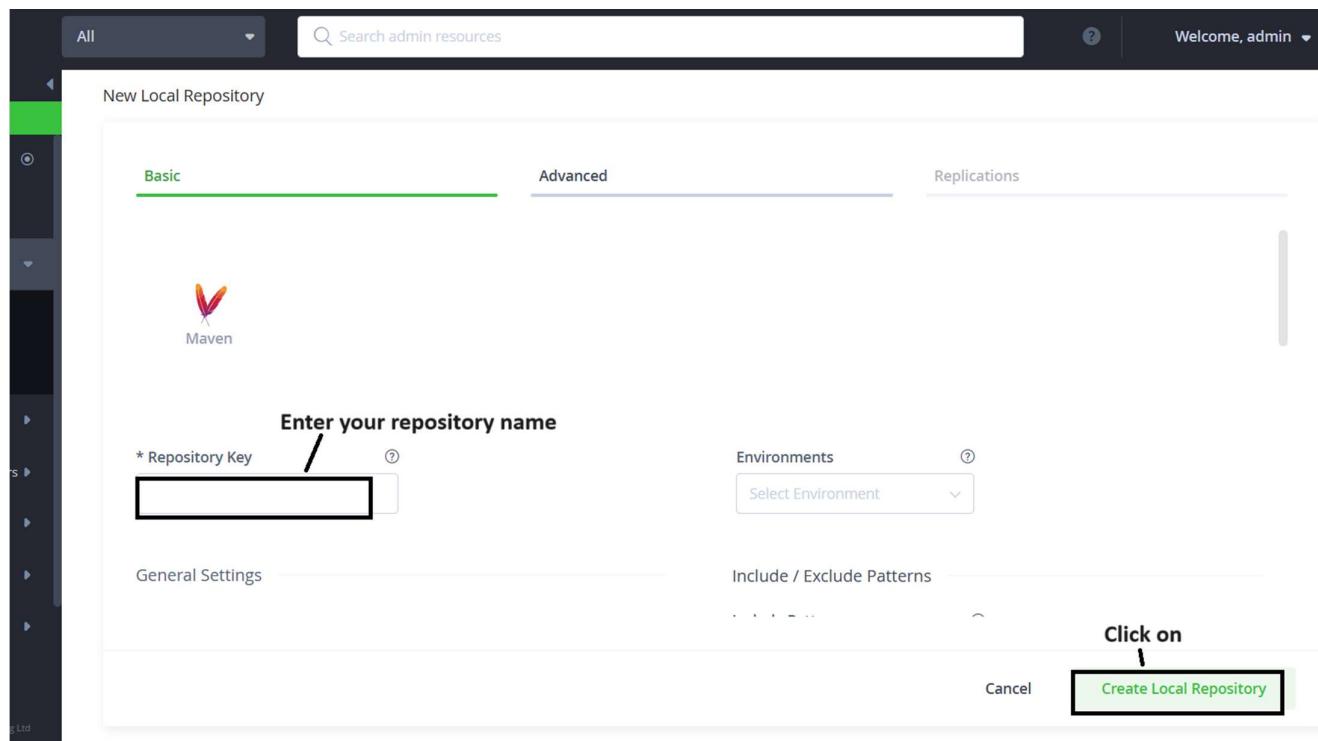
After clicking on ‘Add Repositories’ button, the following page will appear.

The screenshot shows the 'Repositories' list page after adding a new repository. The sidebar menu is identical. The main content area displays the same table as before, but now it shows 2 Repositories: 'example-repo-local' (Type: Generic) and a newly added 'Local Repository'. A green button labeled '+ Add Repositories' is highlighted with a black arrow pointing to it from the text 'Click on'. A sub-menu for 'Local Repository' is also visible.

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



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



After clicking on ‘Create Local Repository’ button, your own repository will gets created successfully as shown in below screenshot.

The screenshot shows the Jfrog Artifactory 'Repositories' page. At the top, there are tabs for 'Local', 'Remote', and 'Virtual', with 'Local' being the active tab. A search bar is at the top right. Below the tabs, it says '2 Repositories'. There is a table with columns: 'Repository Key', 'Type', 'Project', 'Environment', 'Replications', and a settings gear icon. Two rows are listed: 'Anilkumar' (Maven type) and 'example-repo-local' (Generic type). A callout box points to the 'Anilkumar' row with the text 'Your own repository Created'. The 'example-repo-local' row has a replication count of 0.

Finally, Jfrog server installation and repository creation was successful. Setup was ready.

Now we will integrate the Jfrog and Jenkins server for delivering artifacts.

For integrating both, follow the step-by-step process.

First open the jenkins dashboard and install the ‘artifactory’ plugin as shown below.

The screenshot shows the Jenkins 'Manage Jenkins > Plugins' page. The left sidebar has options for 'Updates', 'Available plugins' (which is selected), 'Installed plugins', and 'Advanced settings'. The main area is titled 'Plugins' with a 'Search this' field containing 'artifactory'. A callout box points to the search field. Below the search is a table with columns 'Install' and 'Name ↓'. The 'Artifactory' plugin is listed with a checked checkbox in the 'Install' column. A callout box points to this checkbox with the text 'Select this'. The 'Artifactory' entry includes a description: 'This plugin allows your build jobs to deploy artifacts and resolve dependencies to and from Artifactory, and then have them linked to the build job that created them. The plugin includes a vast collection of features, including a rich pipeline API library and release management for Maven and Gradle builds with Staging and Promotion.' Other plugins listed are 'lambdatest-automation' and 'zdevops'. At the bottom, there are two buttons: 'Install without restart' and 'Download now and install after restart'. A note says 'Update information obtained: 1 hr 48 min ago' and 'Check now'.

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

The screenshot shows the Jenkins 'Manage Jenkins > Plugins' section. On the left, there's a sidebar with links: 'Updates', 'Available plugins', 'Installed plugins', 'Advanced settings', and 'Download progress'. The 'Download progress' link is highlighted with a light gray background. The main content area is titled 'Download progress' and includes a 'Preparation' section with a bulleted list: '• Checking internet connectivity', '• Checking update center connectivity', and '• Success'. Below this is an 'Artifactory' section with two items: 'Loading plugin extensions' and 'Plugin installation u/p'. The 'Plugin installation u/p' item has a progress bar that is mostly blue, indicating it is currently installing. A tooltip 'Installing' is shown above the progress bar. To the right of the progress bar, there's a status message 'Plugin installation u/p' with a small circular icon. At the bottom, there are two links: 'Go back to the top page' and 'Restart Jenkins when installation is complete and no jobs are running' with a checkbox.

After completion of plugin installation, the page seems, as shown below.

This screenshot is similar to the previous one but shows the completed state of the plugin installation. The 'Manage Jenkins > Plugins' link in the sidebar is highlighted with a black box and a callout arrow pointing to it from the text 'Click on'. The main content area shows the 'Download progress' section with the 'Preparation' and 'Artifactory' sections. In the 'Artifactory' section, both 'Loading plugin extensions' and 'Plugin installation u/p' now have green checkmarks next to the word 'Success', indicating they have completed successfully. A bold text 'Installation completed' is displayed to the right of the success messages.

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

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 these are sections for 'Build Queue' (empty) and 'Build Executor Status' (2 idle). The main area is titled 'Manage Jenkins' and displays a message about a new Jenkins version available for download. It also shows a warning about multiple security vulnerabilities in Jenkins 2.401.3 core and libraries, with a link to update. A callout box labeled 'Click on' points to the 'System' button under 'System Configuration', which is described as 'Configure global settings and paths'. Other options shown include 'Tools' (configure tools), 'Plugins' (add, remove, disable or enable plugins), 'Nodes and Clouds' (add, remove, control and monitor nodes), and 'Managed files' (e.g. settings.xml for maven, central managed scripts, custom files).

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' and includes a 'Home directory' section where the default directory is set to '/var/lib/jenkins'. Below this is a 'System Message' section with a placeholder for notifications. A callout arrow points down the page with the instruction 'Scroll down the page, to view Jfrog settings'.

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

JFrog

Use the Credentials Plugin ?

JFrog Platform Instances

List of JFrog platform instances that projects will want to deploy artifacts and build info

Add JFrog Platform Instance

Click on

After clicking on ‘Add Jfrog Platform Instance’, the following page will appear.

JFrog instance details

Instance ID ? You can enter any name
Myjfrog

JFrog Platform URL ? Jfrog URL to be pasted
http://your_IP:8082

Advanced Configuration ▾

Default Deployer Credentials

Username ?
Password ?

Second click on Save First click on Apply Click on Test Connection

After clicking on ‘Test connection’, if you seen the following screen. Connection established successfully.

Found JFrog Artifactory 7.6.14 at http://13.235.17.128:8082/artifactory
JFrog Distribution not found at http://13.235.17.128:8082/distribution

Connection established successfully

Test Connection

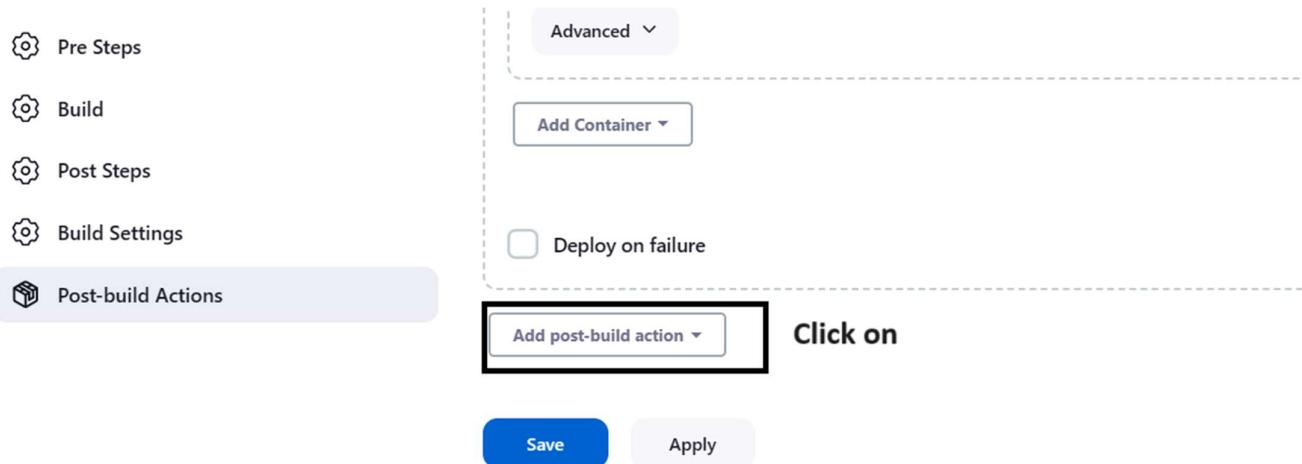
After clicking on ‘Apply and Save’ button, the following screen will appear.

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', 'Build Queue' (which says 'No builds in the queue.'), and 'Build Executor Status' (which shows 1 Idle and 2 Idle). The main area has tabs 'All' and '+' at the top. Below them is a table with columns 'S', 'W', 'Name', 'Last Success', and 'Last Failure'. A project named 'Tomcat Deployment' is listed with a green checkmark icon, a yellow cloud icon, and the name 'Tomcat Deployment'. To its right are 'Last Success' (1 day 4 hr #5) and 'Last Failure' (1 day 5 hr #1). Below the table are 'Icon legend', 'Atom feed for all', and 'Atom feed for failures'. A context menu is open over the 'Tomcat Deployment' row. It has a heading 'Click on drop down icon'. Inside the menu, there's a 'Configure' option which is highlighted with a red box and a black arrow pointing to it from the text 'Click on'. Other options in the menu include '>/ Changes', 'Workspace', 'Build Now', 'Delete Maven project', 'Modules', 'SonarQube', and 'Rename'. The entire context menu is enclosed in a white box with a black border.

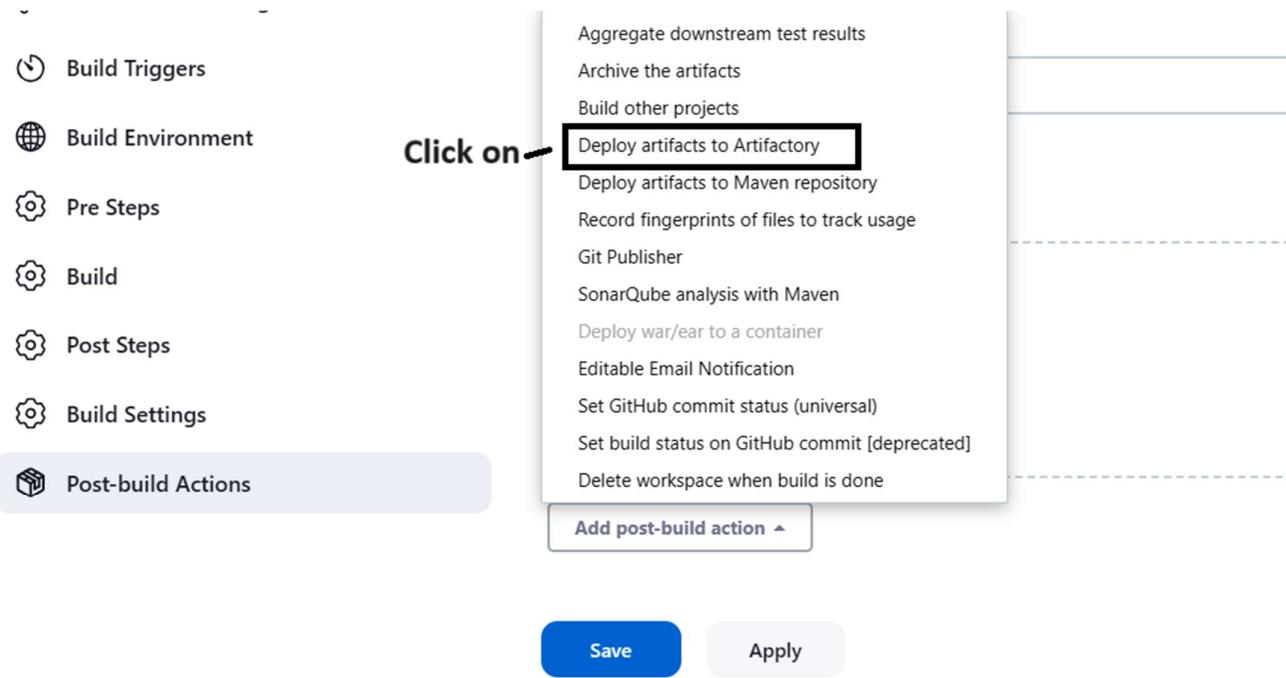
After clicking on ‘Configure’ button, the following page will appear.

The screenshot shows the 'Configuration' page for the 'Tomcat Deployment' project. At the top, there's a breadcrumb navigation: 'Dashboard > Tomcat_Deployment > Configuration'. The page title is 'Configure' and there are two tabs: 'General' (which is selected) and 'Advanced'. On the left, there's a sidebar with sections: 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Pre Steps', 'Build', 'Post Steps', 'Build Settings' (which is highlighted with a red box and a black arrow pointing to it from the text 'Click on'), and 'Post-build Actions'. The 'General' tab contains a 'Description' field with a placeholder '(Plain text) Preview'. Below it are several checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterized', 'Throttle builds', and 'Execute concurrent builds if necessary'. At the bottom of the 'General' tab, there's a 'Advanced' dropdown menu.

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



After clicking on ‘Add post-build action’ button, the following screen will appear.



After clicking on ‘Deploy artifacts to Artifactory’ button, the following page will appear.

The screenshot shows a Jenkins configuration page for a job. On the left, there is a sidebar with the following options: Source Code Management, Build Triggers, Build Environment, Pre Steps, Build, Post Steps, Build Settings, and Post-build Actions. The 'Post-build Actions' option is highlighted with a light blue background. The main panel has a dashed border and contains the following fields:

- Tomcat URL: http://52.66.211.153:8080/
- Advanced dropdown menu
- Add Container dropdown menu
- Deploy on failure checkbox
- Add post-build action dropdown menu

At the bottom are two buttons: Save (blue) and Apply (grey).

An arrow points upwards from the 'Deploy on failure' checkbox area towards the top of the page, with the text "Scroll up the page, to see the Jfrog related settings".

After scrolling up the page, the following screen will appear.

The screenshot shows the 'Deploy artifacts to Artifactory' configuration screen. On the left, there is a sidebar with the same set of options as the previous screenshot. The 'Post-build Actions' option is highlighted. The main panel has a dashed border and contains the following fields:

- Artifactory server: http://13.235.17.128:8082/artifactory
- Target releases repository
- Target snapshot repository
- Custom staging configuration

A red arrow points downwards from the 'Custom staging configuration' field towards the bottom of the page, with the text "Scroll down for refresh button".

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

The screenshot shows a confirmation message: "Items refreshed successfully". There is a "Refresh Repositories" button with a black border and a dropdown menu icon next to it. A red box highlights the "Refresh Repositories" button. To the left, there is a "Target snapshot repository" dropdown menu with "Anilkumar" selected. Below it is a "Custom staging configuration" dropdown menu with "None" selected. At the bottom left, there is an "Override default credentials" checkbox. A red box highlights the "Override default credentials" checkbox.

Text annotations on the right side of the screen:

- "Click on to select our own repository" with a red box highlighting the dropdown menu icon next to the "Target snapshot repository" field.
- "Click on" with a red box highlighting the "Refresh Repositories" button.

Finally click on ‘Apply and Save’ buttons as shown in below screenshot.

The screenshot shows the Jenkins 'Configure' screen for a project. On the left, a sidebar lists various build configurations: General, Source Code Management, Build Triggers, Build Environment, Pre Steps, Build, Post Steps, Build Settings, and Post-build Actions. 'Post-build Actions' is currently selected and highlighted with a grey background. The main panel displays the 'Deploy artifacts to Artifactory' configuration. It includes fields for 'Artifactory server' (set to http://13.235.17.128:8082/artifactory), 'Target releases repository' (set to Anilkumar), 'Target snapshot repository' (set to Anilkumar), and 'Custom staging configuration' (set to None). At the bottom of the panel, there are two buttons: 'Save' (highlighted in blue) and 'Apply'. Below the panel, a message says 'Items refreshed successfully' and a 'Refresh Repositories' button is available. A note at the bottom indicates to 'Second click on Save' and 'First Click on Apply'.

clicking on ‘Apply and Save’ buttons, the following page will appear. And now project is ready to build.

The screenshot shows the Jenkins dashboard for the 'Maven project Tomcat_Deployment'. The top navigation bar shows 'Dashboard > Tomcat_Deployment >'. On the left, a sidebar provides quick access to Status, Changes, Workspace, Build Now (highlighted with a red box and an arrow pointing to it, with the text 'Click on' next to it), Configure, Delete Maven project, and Modules. The main content area is titled 'Maven project Tomcat_Deployment'. It includes links for 'Artifactory Build Info' (with a green circular icon), 'SonarQube' (with a blue square icon), and 'Permalinks' (with a blue link icon).

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

The screenshot shows a sidebar with options: Workspace, Build Now (highlighted with a yellow arrow), Configure, Delete Maven project, Modules, Artifactory Build Info, SonarQube, and Rename. To the right, there's a section titled 'Artifactory Build Info' with a green icon, another for 'SonarQube' with a blue icon, and a 'Permalinks' section. Below these are lists of build history items:

- Last build (#6), 7.3 sec ago
- Last stable build (#5), 1 day 5 hr ago
- Last successful build (#5), 1 day 5 hr ago
- Last failed build (#1), 1 day 6 hr ago
- Last unsuccessful build (#1), 1 day 6 hr ago
- Last completed build (#5), 1 day 5 hr ago

A 'Build History' card is shown with a 'trend' dropdown set to '▼'. It displays build #6 from Oct 13, 2023, at 1:32 PM. A black arrow points from the text 'click on it' to the build number 'it' in the card.

After clicking on ‘#6’ button, the following screen will appear.

The screenshot shows the Jenkins build #6 details page. The top navigation bar shows 'Dashboard > Tomcat_Deployment > #6'. The left sidebar has links: Status (highlighted with a yellow arrow), Changes, Console Output (highlighted with a yellow arrow and labeled 'Click on'), Edit Build Information, Delete build '#6', Git Build Data, Redeploy Artifacts, Artifactory Build Info, See Fingerprints, and Previous Build. The main content area shows a green checkmark icon next to 'Build #6 (Oct 13, 2023, 1:32:27 PM)'. To the right, it says 'No changes.' and 'Started by anonymous user'. It also shows 'Revision: 13f4fd861346fa4f5a85dc4a19d5279500e2e853' and 'Repository: <https://github.com/mannem302/AnilKumar.git>'. Below this is a 'Module Builds' section with an Artifactory Build Info icon and a link to 'Artifactory Build Info'.

After clicking on ‘Console output’ button, we can see the following page and in that we can find build related logs.

The screenshot shows the Jenkins interface for a build named 'Tomcat_Deployment' (Build #6). The left sidebar has links for Status, Changes, Console Output (which is selected), View as plain text, Edit Build Information, Delete build '#6', Git Build Data, Redeploy Artifacts, Artifactory Build Info, See Fingerprints, and Previous Build. The main area is titled 'Console Output' with a green checkmark icon. It displays a log of command-line activity:

```
13:32:27 Started by user unknown or anonymous
13:32:27 Running as SYSTEM
13:32:27 Building in workspace /var/lib/jenkins/workspace/Tomcat_Deployment
13:32:27 The recommended git tool is: NONE
13:32:27 No credentials specified
13:32:27 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Tomcat_Deployment/.git # timeout=10
13:32:27 Fetching changes from the remote Git repository
13:32:27 > git config remote.origin.url https://github.com/mannem302/AnilKumar.git # timeout=10
13:32:27 Fetching upstream changes from https://github.com/mannem302/AnilKumar.git
13:32:27 > git --version # timeout=10
13:32:27 > git --version # 'git version 2.34.1'
13:32:27 > git fetch --tags --force --progress -- https://github.com/mannem302/AnilKumar.git +refs/heads/*:refs/remotes/origin/*
13:32:28 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
13:32:28 Checking out Revision 13f4fd861346fa4f5a85dc4a19d5279500e2e853 (refs/remotes/origin/main)
13:32:28 > git config core.sparsecheckout # timeout=10
13:32:28 > git checkout -f 13f4fd861346fa4f5a85dc4a19d5279500e2e853 # timeout=10
13:32:28 Commit message: "Update index.html"
```

A large downward-pointing arrow on the right side of the log area is labeled 'Scroll down the page'.

After scrolling down the page, you will find the logs related to Jfrog as shown below.

```
13:32:59 [INFO] Artifactory Build Info Recorder: Saving Build Info to '/var/lib/jenkins/workspace/Tomcat_Deployment/target/build-info.json'
13:32:59 [INFO] Deploying artifact: http://13.235.17.128:8082/artifactory/Anilkumar/com/Mannem/AnilKumar/1.0/AnilKumar-1.0.war
13:33:00 [INFO] Deploying artifact: http://13.235.17.128:8082/artifactory/Anilkumar/com/Mannem/AnilKumar/1.0/AnilKumar-1.0.pom
13:33:00 [INFO] Artifactory Build Info Recorder: Deploying build info ...
13:33:00 [INFO] Deploying build info...
13:33:01 [INFO] Build-info successfully deployed. Browse it in Artifactory under
http://13.235.17.128:8082/artifactory/webapp/builds/Tomcat_Deployment/6
```

Successfully, you have deployed artifacts into Jfrog server.

We can check this in Jfrog server, as shown below.

The screenshot shows the JFrog Artifactory Open Source interface. The left sidebar includes Application, Dashboard, Artifactory (with Release Bundles, Packages, Builds, Artifacts selected), and Xray. The main area shows a tree view of artifacts under 'Anilkumar': com/Mannem/AnilKumar/1.0 (containing AnilKumar-1.0.pom, AnilKumar-1.0.war, and maven-metadata.xml). A search bar at the top right says 'Search Artifacts'. To the right of the tree, there is a detailed view for 'Anilkumar' under 'General' tab:

Info
Name: Anilkumar
Package Type: Maven
Repository Path: Anilkumar/