

Lesson 07 Demo 01

Configuring Deploy Plugin for Performing Automated CD

Objective: To configure a CI/CD pipeline in Jenkins for deploying a Java application to Tomcat Apache

Tools required: GitHub, Jenkins, and Tomcat Apache

Prerequisites: You need to have a Jenkins up and running.

Steps to be followed:

1. Install Tomcat Apache 9 on Ubuntu VM
2. Log in to the Jenkins CI tool and install the Deploy to container plugin
3. Configure the deployment stage in the Jenkins pipeline

Step 1: Install Tomcat Apache 9 on Ubuntu VM

- 1.1 Open the terminal in your lab and use the following command to switch to the root user:

sudo su

```
sakshiguptasimp@ip-172-31-25-100:~$ sudo su
root@ip-172-31-25-100:/home/sakshiguptasimp#
```

1.2 Install Tomcat Apache and other required packages using the following command:

apt update

apt install tomcat9 tomcat9-admin

```
root@ip-172-31-25-100:/home/sakshiguptasimp# apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Ign:6 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:8 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:9 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:7 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.28/deb InRelease [1189 B]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1638 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [307 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1864 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1421 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [316 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1073 kB]
Get:16 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages [31.5 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [245 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [42.7 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [10.4 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [67.1 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.0 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [27.2 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.2 kB]
Get:24 https://pkg.jenkins.io/debian-stable binary/ Packages [26.7 kB]
Get:25 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [246 kB]
Get:26 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.28/deb Packages [13.9 kB]
```

```
root@ip-172-31-25-100:/home/sakshiguptasimp# apt install tomcat9 tomcat9-admin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libapr1 libeclipse-jdt-core-java libtcnative-1 libtomcat9-java
  tomcat9-common
Suggested packages:
  tomcat9-docs tomcat9-examples tomcat9-user
The following NEW packages will be installed:
  libapr1 libeclipse-jdt-core-java libtcnative-1 libtomcat9-java tomcat9
  tomcat9-admin tomcat9-common
0 upgraded, 7 newly installed, 0 to remove and 160 not upgraded.
Need to get 12.7 MB of archives.
After this operation, 16.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8ubuntu0.22.04.1 [108 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 libeclipse-jdt-core-java all 3.27.0+eclipse4.21-1 [6240 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 libtomcat9-java all 9.0.58-1ubuntu0.1 [6047 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 tomcat9-common all 9.0.58-1ubuntu0.1 [60.9 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 tomcat9 all 9.0.58-1ubuntu0.1 [37.0 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 tomcat9-admin all 9.0.58-1ubuntu0.1 [68.8 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 libtcnative-1 amd64 1.2.31-1build1 [95.1 kB]
Fetched 12.7 MB in 1s (11.3 MB/s)
```

1.3 Once the installation is complete, open the **tomcat-users.xml** file using the following command:

vi /etc/tomcat9/tomcat-users.xml

```
root@ip-172-31-25-100:/home/sakshiguptasimp# vi /etc/tomcat9/tomcat-users.xml
```

1.4 Add the following content in **tomcat-users.xml** file:

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

```
<!--
<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"/>
-->
<user username="tomcat" password="password" roles="admin-gui,manager-gui,manager-script"/>
</tomcat-users>
"/etc/tomcat9/tomcat-users.xml" 58L, 2850B
```

Note: To save the file and exit, press **Esc**, then type **:wq**, and press **Enter**

1.5 Open the **server.xml** file using the following command and scroll down to change the connector port number of Tomcat to **9090**:

```
vim /etc/tomcat9/server.xml
```

```
root@ip-172-31-25-100:/home/sakshiguptasimp# vim /etc/tomcat9/server.xml
```

```
<Service name="Catalina">

  <!--The connectors can use a shared executor, you can define one or more named thread pools-->
  <!--
  <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
    maxThreads="150" minSpareThreads="4"/>
  -->

  <!-- A "Connector" represents an endpoint by which requests are received
  and responses are returned. Documentation at :
  Java HTTP Connector: /docs/config/http.html
  Java AJP  Connector: /docs/config/ajp.html
  APR (HTTP/AJP) Connector: /docs/apr.html
  Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
  -->
  <Connector port="9090" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
  <!-- A "Connector" using the shared thread pool-->
  <!--
  <Connector executor="tomcatThreadPool"
    port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
  -->
```

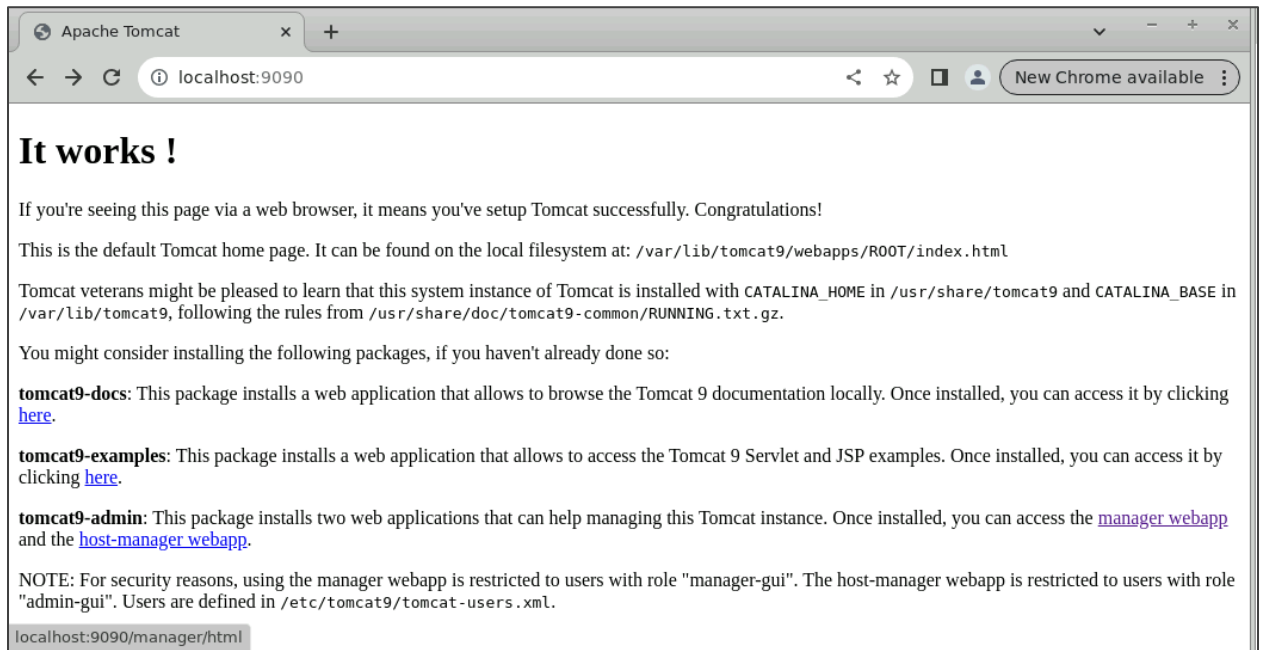
Note: To save the file and exit, press **Esc**, then type **:wq**, and press **Enter**

1.6 Restart Tomcat using the following command:

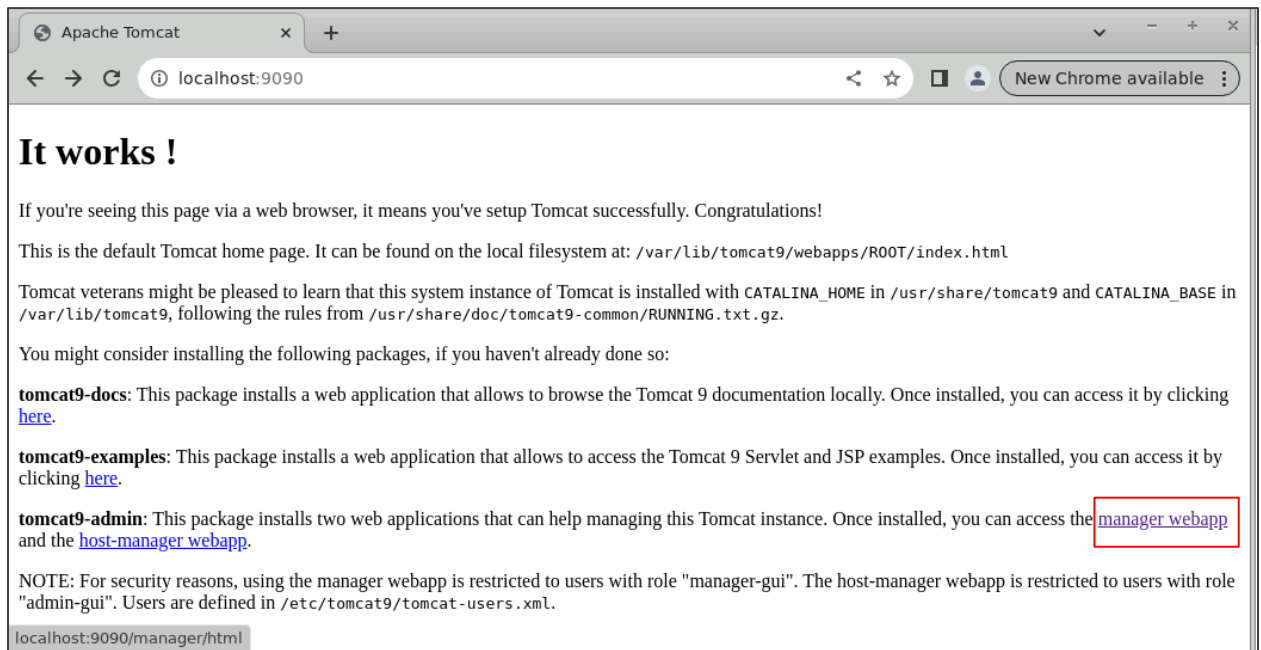
systemctl restart tomcat9

```
root@ip-172-31-25-100:/home/sakshiguptasimp# systemctl restart tomcat9
root@ip-172-31-25-100:/home/sakshiguptasimp# █
```

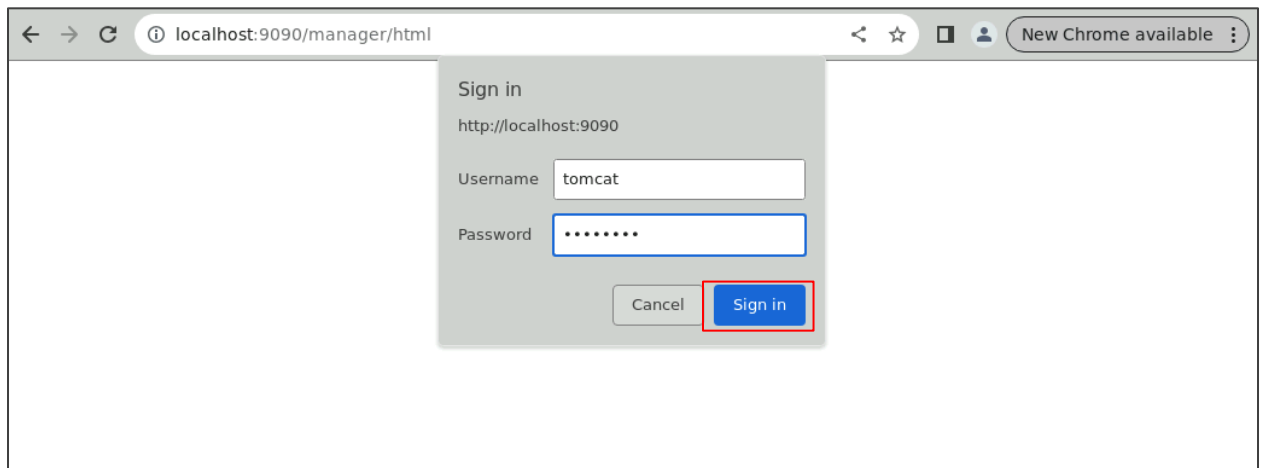
1.7 Navigate to **localhost:9090** in your web browser and access Tomcat





1.8 Click and access **manager webapp** to make sure the Tomcat setup is complete



1.9 Enter the credentials and click on **Sign in**



Note: The credentials for accessing Tomcat manager web app are
Username: **tomcat** and Password: **password**.

Tomcat Web Application Manager

Message: OK

Manager

[List Applications](#)
[HTML Manager Help](#)
[Manager Help](#)
[Server Status](#)


Applications

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

Step 2: Log in to Jenkins CI tool and install Deploy to container plugin

2.1 Navigate to **localhost:8080** in your web browser, enter your credentials, and click on **Sign in**

← → ↻ localhost:8080/login?from=%2F
🔍 ☆ 📱 👤 Finish update ⋮



Sign in to Jenkins

Username

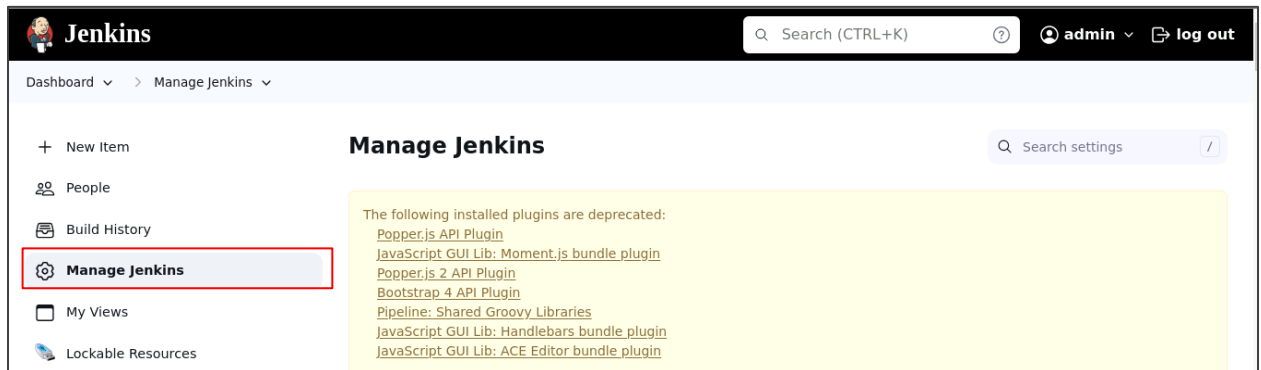
Password

☐ Keep me signed in

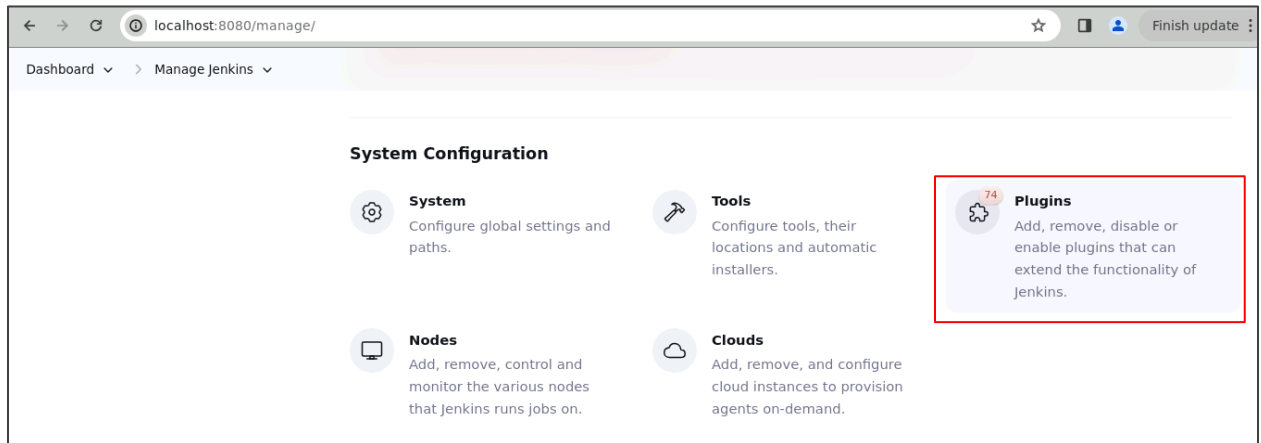
Sign in

Note: The credentials for accessing Jenkins in the lab are Username: **admin** and Password: **Root123\$**.

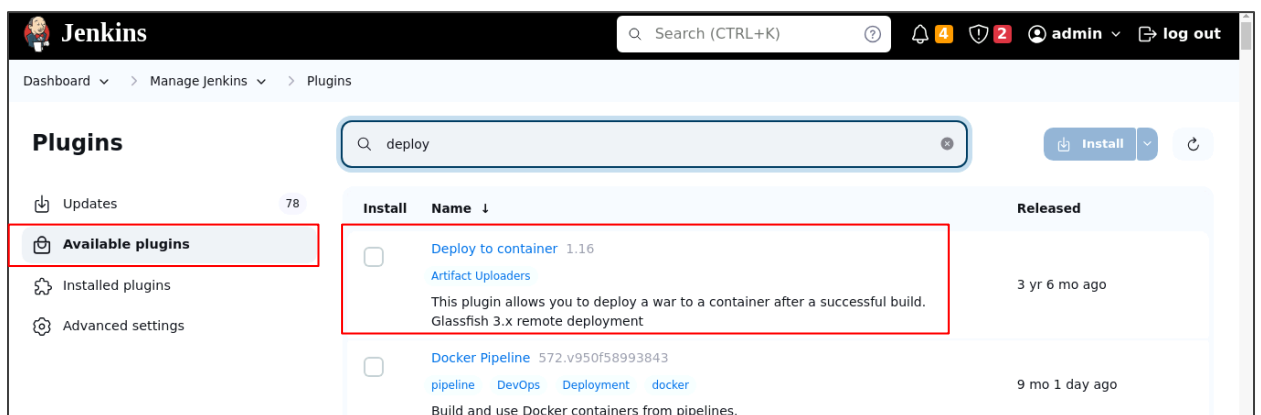
2.2 Click on **Manage Jenkins** on the Jenkins dashboard



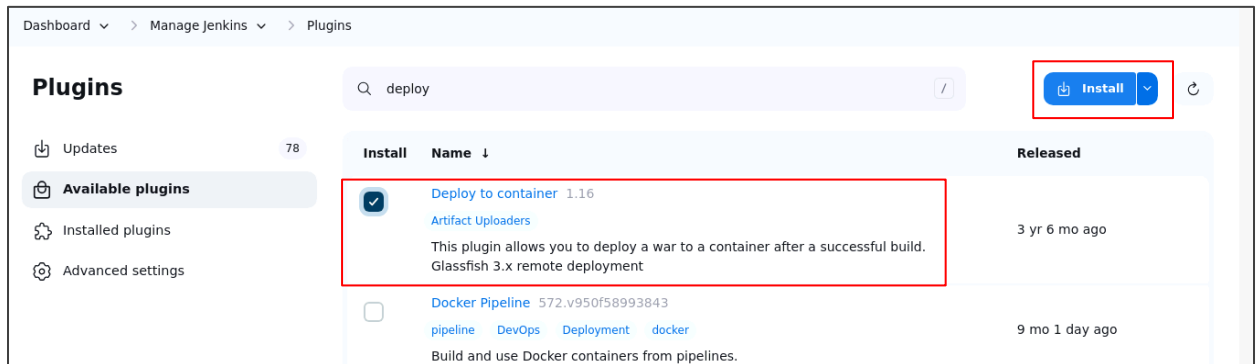
2.3 Scroll down and click on **Plugins** under **System Configuration**



2.4 Navigate to **Available plugins** and search for **Deploy to container** plugin



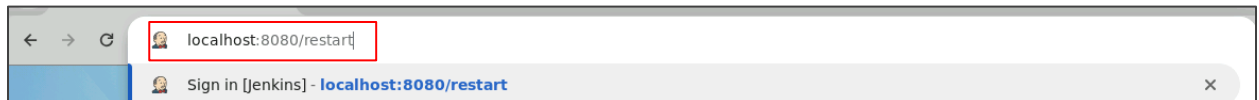
2.5 Select **Deploy to container** plugin, click on **Install**



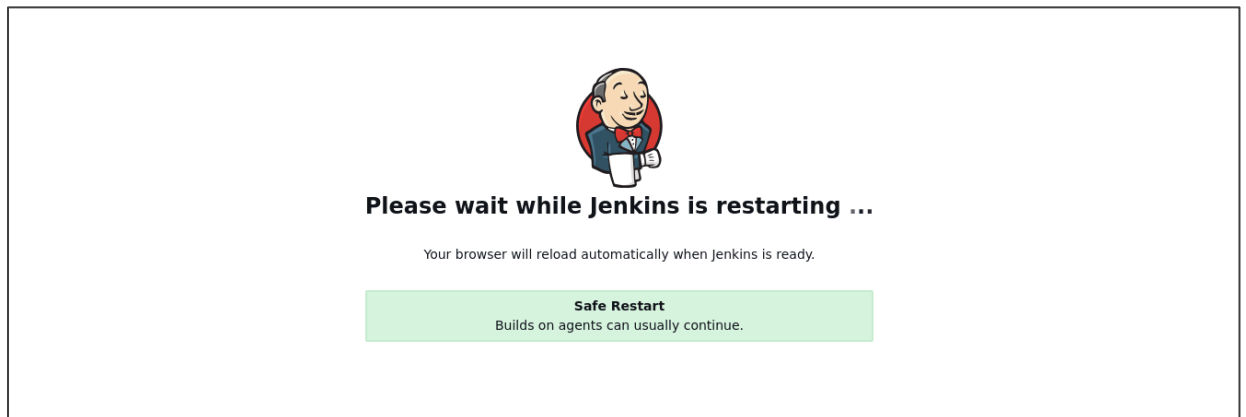
The screenshot shows the Jenkins 'Plugins' page. The breadcrumb navigation at the top reads 'Dashboard > Manage Jenkins > Plugins'. A search bar contains the text 'deploy'. On the right, an 'Install' button is highlighted with a red box. On the left, a sidebar lists 'Available plugins', 'Installed plugins', and 'Advanced settings'. The main table lists plugins with columns 'Install', 'Name', and 'Released'. The 'Deploy to container' plugin (version 1.16) is selected with a checked checkbox and is highlighted with a red box. Its description states: 'This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment'. Below it, the 'Docker Pipeline' plugin (version 572.v950f58993843) is listed with an unchecked checkbox. The 'Released' column shows '3 yr 6 mo ago' for the first plugin and '9 mo 1 day ago' for the second.

Install	Name	Released
<input checked="" type="checkbox"/>	Deploy to container 1.16 Artifact Uploaders This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment	3 yr 6 mo ago
<input type="checkbox"/>	Docker Pipeline 572.v950f58993843 pipeline DevOps Deployment docker Build and use Docker containers from pipelines.	9 mo 1 day ago

2.6 After installation, navigate to the following URL to restart Jenkins: **http://localhost:8080/restart**



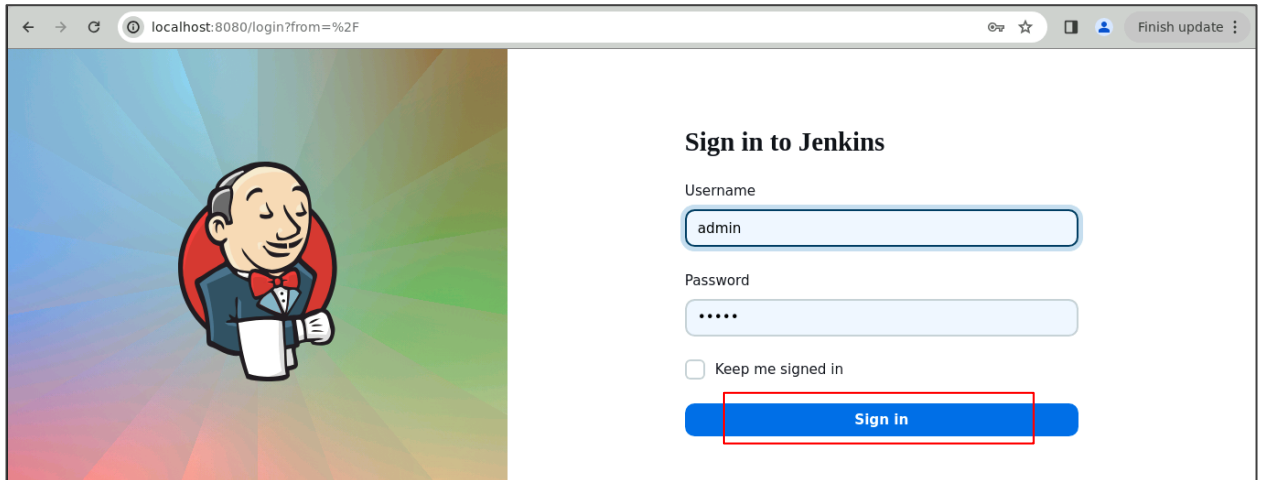
The screenshot shows a web browser window. The address bar contains the URL 'localhost:8080/restart', which is highlighted with a red box. Below the address bar, a tab is labeled 'Sign in [Jenkins] - localhost:8080/restart'.



The screenshot shows the Jenkins restart page. At the top center is the Jenkins logo (a cartoon man in a suit). Below the logo, the text reads 'Please wait while Jenkins is restarting ...'. Underneath, a smaller line of text says 'Your browser will reload automatically when Jenkins is ready.' At the bottom, there is a green button labeled 'Safe Restart' with the text 'Builds on agents can usually continue.' below it.

Step 3: Configure the deployment stage in the Jenkins pipeline

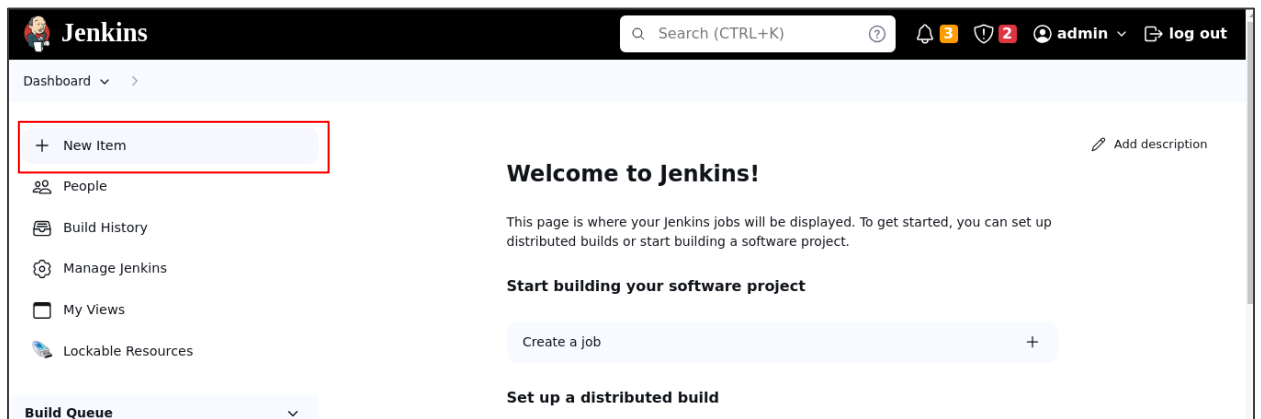
3.1 Enter your credentials and **Sign in** to Jenkins CI tool



A screenshot of a web browser showing the Jenkins login page. The address bar indicates the URL is `localhost:8080/login?from=%2F`. On the left, there is a cartoon illustration of a man in a tuxedo. On the right, the heading "Sign in to Jenkins" is followed by input fields for "Username" (containing "admin") and "Password" (masked with dots). Below these is a checkbox for "Keep me signed in" and a blue "Sign In" button, which is highlighted with a red rectangle.

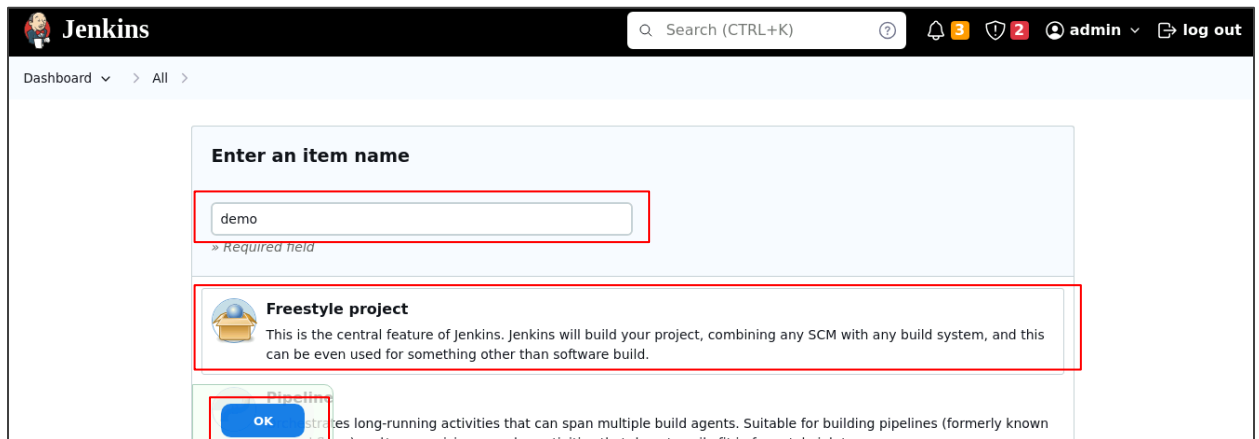
Note: The credentials for accessing Jenkins in the lab are Username: **admin** and Password: **Root123\$**.

3.2 Click on **New Item** to create new Jenkins job



A screenshot of the Jenkins dashboard. The top navigation bar includes the Jenkins logo, a search bar, and user information for "admin" with a "log out" link. The left sidebar contains a menu with "New Item" (highlighted with a red rectangle), "People", "Build History", "Manage Jenkins", "My Views", and "Lockable Resources". The main content area displays a "Welcome to Jenkins!" message and a "Start building your software project" section with a "Create a job" button. A "Build Queue" section is visible at the bottom left.

3.3 Select **Freestyle project** while creating Jenkins job and provide a custom job name, then click on **OK** to continue



The image shows the Jenkins 'Enter an item name' dialog. At the top, there's a search bar and user information. Below, a text input field contains 'demo'. A red box highlights this field. Below the input, a red box highlights the 'Freestyle project' option, which is described as the central feature of Jenkins. At the bottom, a blue 'OK' button is highlighted with a red box.

Jenkins

Search (CTRL+K)

admin log out

Dashboard > All >

Enter an item name

demo

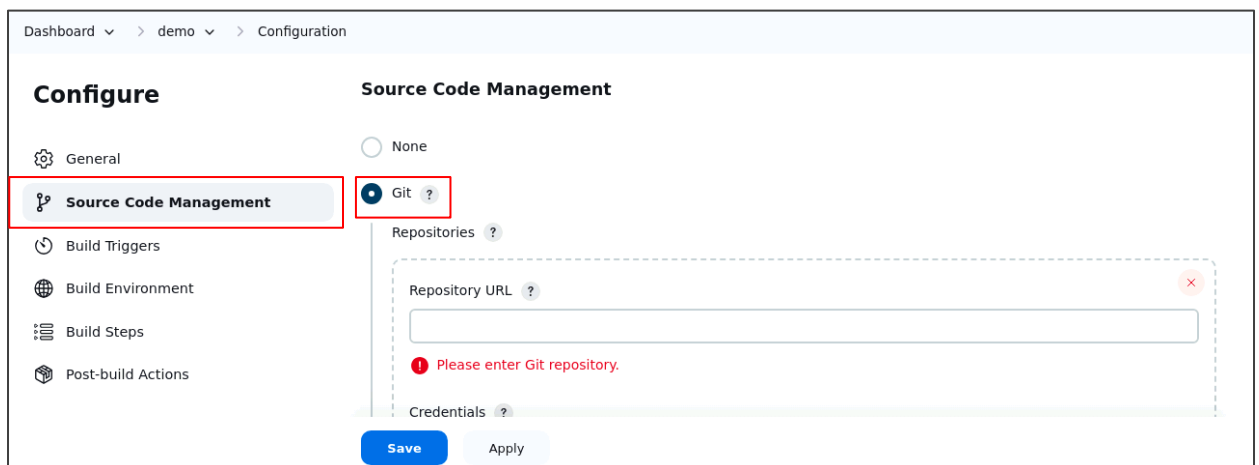
Required field

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

OK

3.4 Go to **Source Code Management** and select **Git**



The image shows the Jenkins 'Configure' page for a job named 'demo'. On the left, a sidebar lists configuration sections: General, Source Code Management (highlighted with a red box), Build Triggers, Build Environment, Build Steps, and Post-build Actions. On the right, the 'Source Code Management' section is active, showing radio buttons for 'None' and 'Git' (selected and highlighted with a red box). Below, there's a 'Repositories' section with a 'Repository URL' input field and a red error message 'Please enter Git repository.' At the bottom, there are 'Save' and 'Apply' buttons.

Dashboard > demo > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

None

Git

Repositories

Repository URL

Please enter Git repository.

Credentials

Save Apply

3.5 Enter the following **Repository URL**:

`https://github.com/github-simplilearn-net/MavenBuild`

Dashboard > demo > Configuration

Configure

- General
- Source Code Management**
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Source Code Management

☐ None

☒ Git

Repositories

Repository URL

`https://github.com/github-simplilearn-net/MavenBuild`

Credentials

3.6 Go to **Build Steps** under **Configure**:

Dashboard > demo > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**
- Post-build Actions

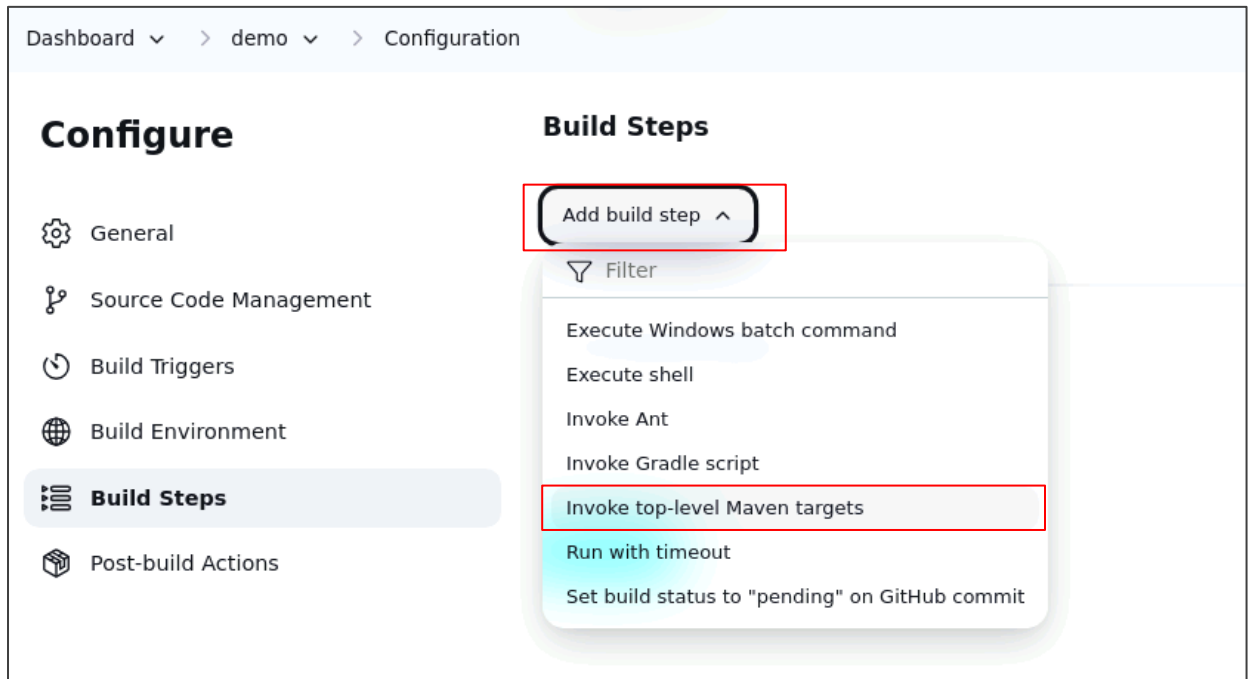
Build Steps

Add build step

Post-build Actions

Add post-build action

3.7 Click on **Add build step** and select **Invoke top-level Maven targets**



3.8 Enter the following **Maven Version**:
mymaven



3.9 Enter the following **Goals**: **clean install**

Dashboard > All > demo > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**
- Post-build Actions

Build Steps

Invoke top-level Maven targets ?

Maven Version

mymaven

Goals

clean install

Advanced

Save Apply

3.10 Go to **Post-build Actions**

Dashboard > demo > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**
- Post-build Actions**

Advanced

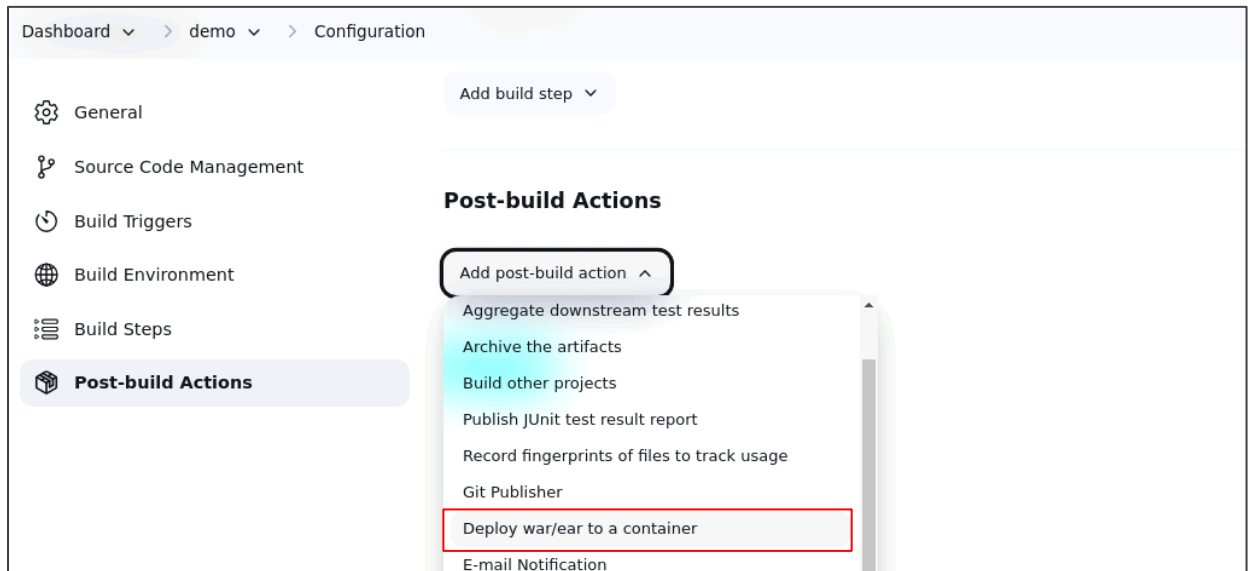
Add build step

Post-build Actions

Add post-build action

Save Apply

3.11 Click on **Add post-build action** and select **Deploy war/ear to a container**

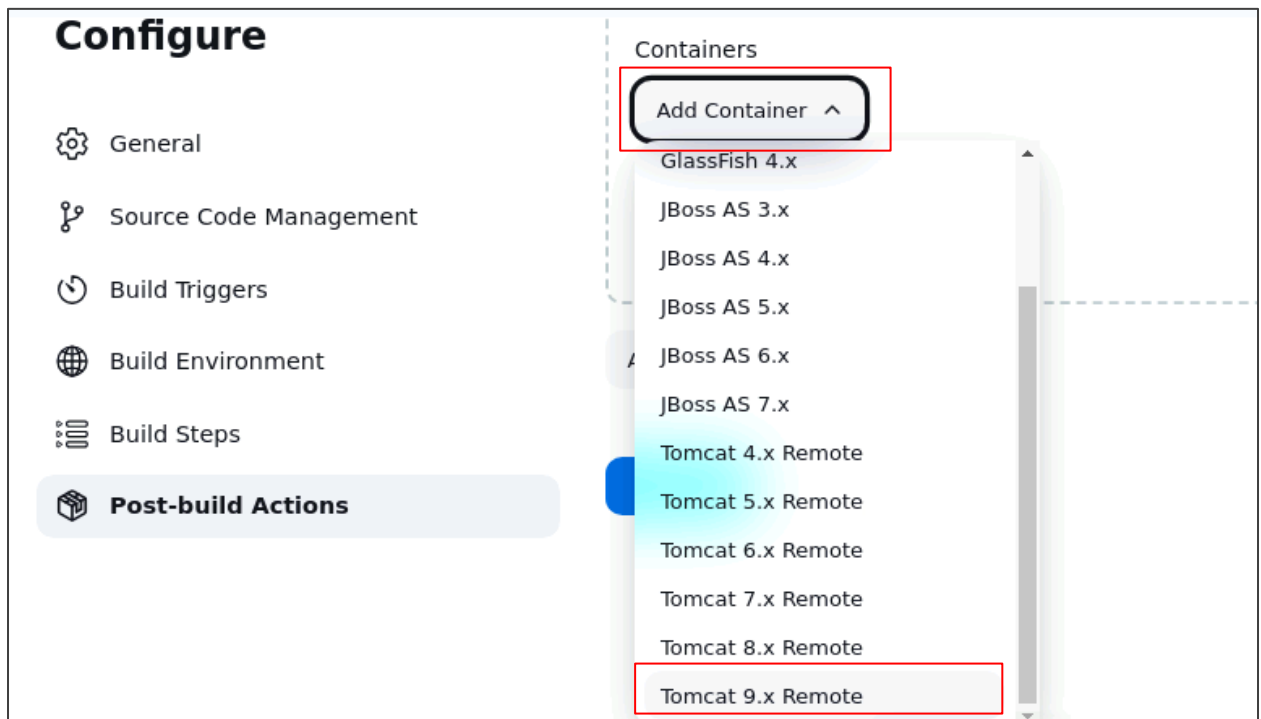


3.12 Enter the following path under **WAR/EAR files**:

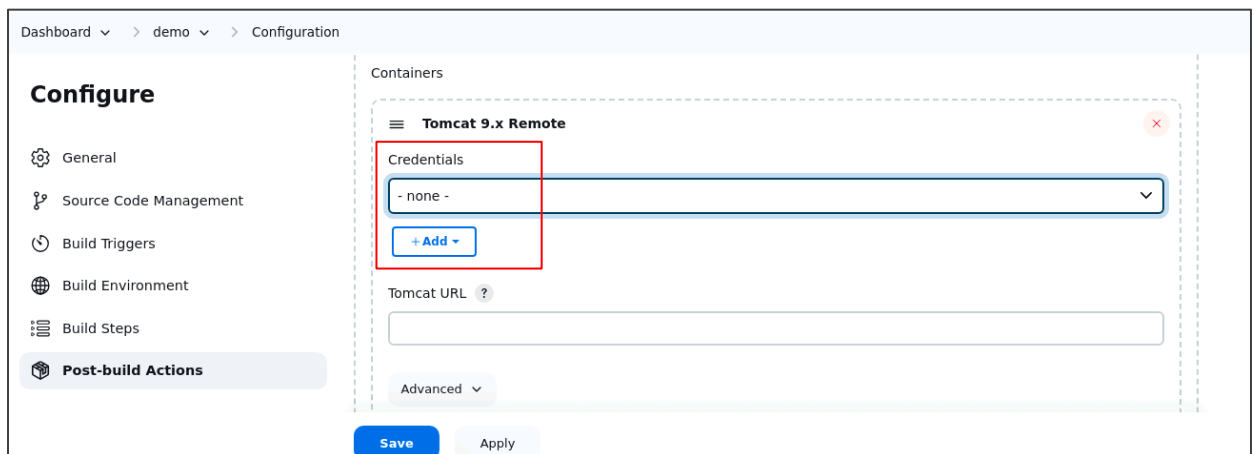
****/*.war**

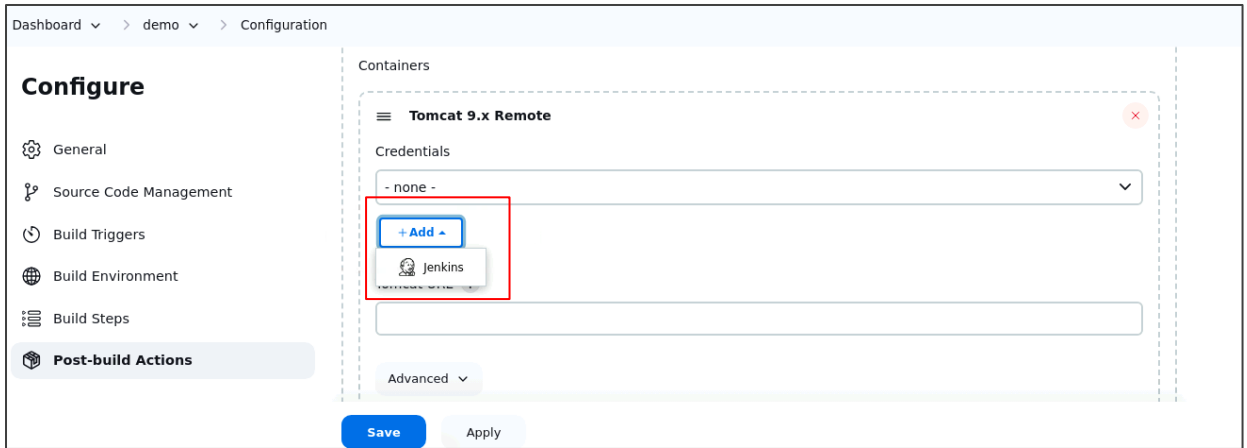


3.13 Click on **Add Container** and select **Tomcat 9.x Remote**



3.14 Click on **Add** under **Credentials** and select **Jenkins** to add the Tomcat credentials





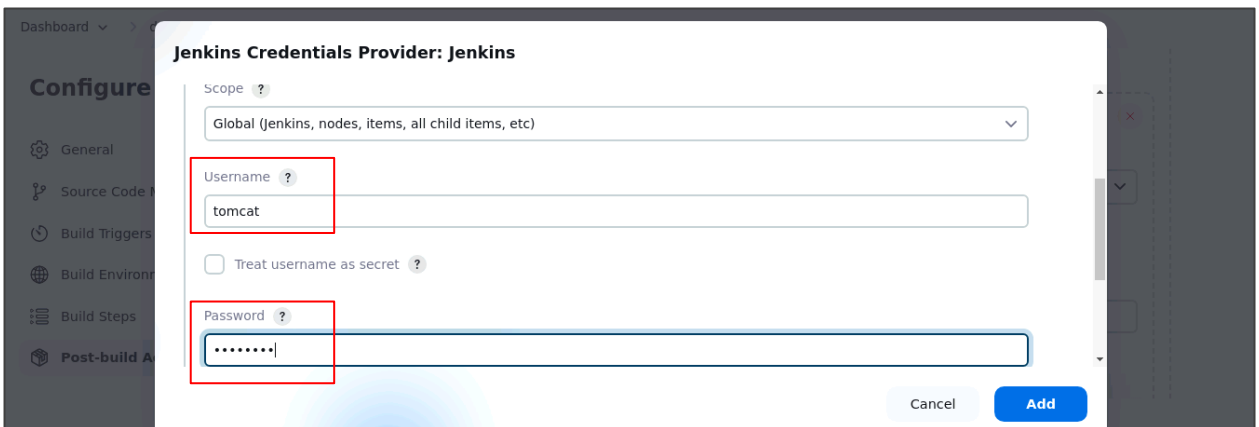
3.15 Scroll down and enter the following details:

Username: tomcat

Password: password

ID: tomcat-id

Description: tomcat-id



3.16 Click on **Add** to save the credentials

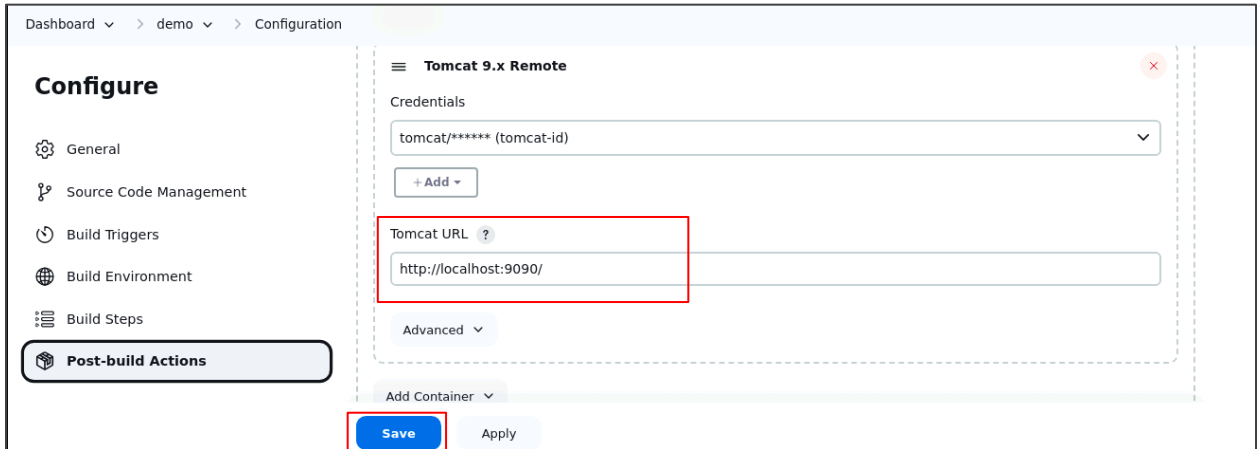
The screenshot shows the 'Jenkins Credentials Provider: Jenkins' dialog box. It has three input fields: 'Password' (masked with dots), 'ID' (containing 'tomcat-id'), and 'Description' (containing 'tomcat-id'). At the bottom right, there are two buttons: 'Cancel' and 'Add'. The 'Add' button is highlighted with a red rectangle.

3.17 In the **Credentials** dropdown menu, select the added credentials

The screenshot shows the Jenkins configuration page for 'Containers'. Under the 'Tomcat 9.x Remote' section, there is a 'Credentials' dropdown menu. The dropdown is open, showing three options: '- none -', '- none -', and 'tomcat/***** (tomcat-id)'. The 'tomcat/***** (tomcat-id)' option is selected and highlighted with a blue background. A red rectangle highlights the dropdown menu. Below the dropdown is a 'Tomcat URL' input field and an 'Advanced' dropdown menu. At the bottom, there are 'Save' and 'Apply' buttons.

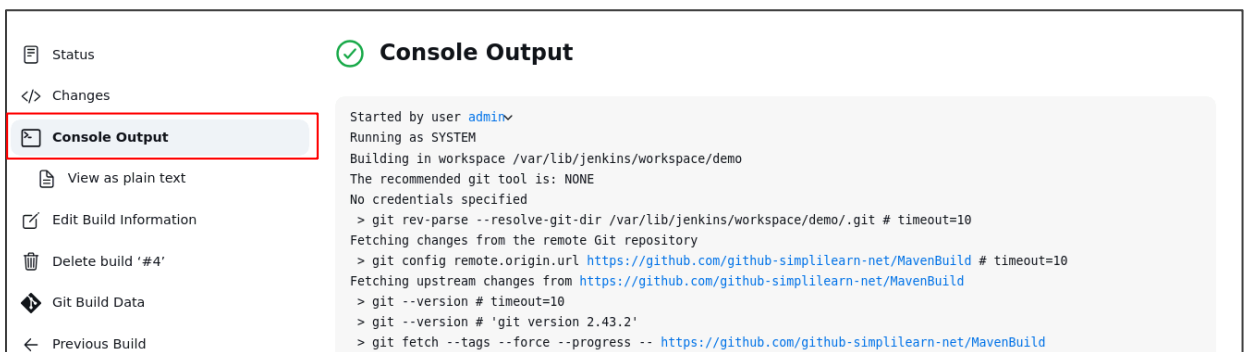
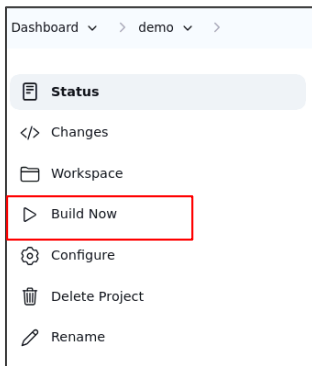
The screenshot shows the same Jenkins configuration page, but now the 'Credentials' dropdown menu is closed. The selected credential, 'tomcat/***** (tomcat-id)', is now visible in the dropdown field. A red rectangle highlights the dropdown field. Below the dropdown is a '+ Add' button. The 'Save' and 'Apply' buttons remain at the bottom.

3.18 Enter the following **Tomcat URL** and **Save** the job:
http://localhost:9090/



The screenshot shows the Jenkins 'Configure' page for a job named 'demo'. The left sidebar contains a menu with 'Post-build Actions' highlighted. The main content area is titled 'Tomcat 9.x Remote' and includes a 'Credentials' dropdown set to 'tomcat/***** (tomcat-id)', a '+ Add' button, and a 'Tomcat URL' field containing 'http://localhost:9090/'. Below this is an 'Advanced' dropdown and an 'Add Container' dropdown. At the bottom, there are 'Save' and 'Apply' buttons, with the 'Save' button highlighted by a red box.

3.19 Click on **Build Now** and then select **Console Output** to check the output

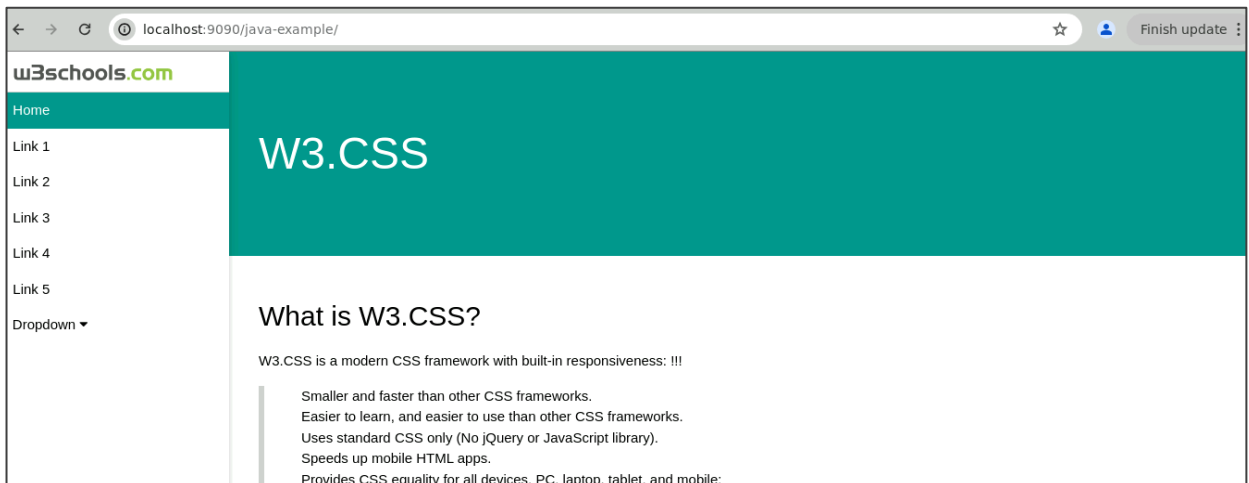


```
Dashboard > All > demo > #4 > Console Output

[INFO] Installing /var/lib/jenkins/workspace/demo/target/java-example.war to
/var/lib/jenkins/.m2/repository/com/java/example/java-example/1.0-SNAPSHOT/java-example-1.0-SNAPSHOT.war
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 4.579 s
[INFO] Finished at: 2024-05-09T11:51:21Z
[INFO] -----
[DeployPublisher][INFO] Attempting to deploy 1 war file(s)
[DeployPublisher][INFO] Deploying /var/lib/jenkins/workspace/demo/target/java-example.war to container Tomcat 9.x
Remote with context null
[/var/lib/jenkins/workspace/demo/target/java-example.war] is not deployed. Doing a fresh deployment.
Deploying [/var/lib/jenkins/workspace/demo/target/java-example.war]
Finished: SUCCESS
```

REST API Jenkins 2.426.3

3.20 Access the deployed application in your web browser with the following URL:
<http://localhost:9090/java-example/>



By following these steps, you have successfully configured a CI/CD pipeline in Jenkins for deploying a Java application to Apache Tomcat.