

Deploy Spring boot application using jenkins with github integration



Shubhangi Thakur · [Follow](#)

6 min read · May 29, 2023

Listen

Share

This blog will take you through how to deploy spring boot application using jenkins with github integration →

Let's see

What we will cover in this artical

Step-1. Setup Jenkins in the vm

Step-2. Integrate Git with Jenkins

Step-3. Integrate Maven with Jenkins Server

Step-4. Setup Tomcat server in the vm

Step-5. Integrate tomcat with jenkins server

Step-6. Setup credentials for email notifications

Step-7. Create maven project in jenkins dashboard

Step-1. Setup Jenkins in the vm

1. Create Gcp vm using ubuntu 22.04 version boot disk

2. Install Java using this commands:-

```
sudo apt update  
sudo apt install openjdk-11-jdk -y
```

3. Install jenkins using this commands:-

```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee /u  
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jen  
  
sudo apt update  
sudo apt install jenkins -y  
sudo systemctl status jenkins
```

4. Open a web browser, and navigate to an external IP address. Use this syntax:

http://ip_address_or_domain:8080

Step-2. Integrate Git with Jenkins

1. Ssh into vm

2. Install git in the vm using this command:

```
sudo apt update  
sudo apt install git  
git --version
```

3. Go to inside Jenkins dashboard: Manage Plugins → Available Plugins → Github → Install without restart

[GitHub plugin 1.37.0](#)

This plugin integrates [GitHub](#) to Jenkins.
[Report an issue with this plugin](#)

Github plugin

4. Go to Manage Plugins → Global Tool Configuration → Git installations → Save → Apply

Git installations

≡ Git

Name

Path to Git executable ?

Install automatically ?

Add Git ▾

Save Apply

Step-3. Integrate Maven with Jenkins Server

1. SSh into vm
2. Run following commands:

```
cd /opt
wget https://dlcdn.apache.org/maven/maven-3/3.9.1/binaries/apache-maven-3.9..
```

Installed zip file → Unzip file

```
tar -xvzf tar.gz file
mv apache-maven-3.8.3 maven
cd maven
ll
cd ~
vi .profile
```

Add java and maven path in this file:

```
# ~/.profile: executed by Bourne-compatible login shells.

if [ "$BASH" ]; then
    if [ -f ~/.bashrc ]; then
        . ~/.bashrc
    fi
fi
M2_HOME=/opt/maven
M2=/opt/maven/bin
JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
PATH=$PATH:$HOME/bin:$JAVA_HOME:$M2_HOME:$M2
export PATH
mesg n 2> /dev/null || true
```

Add java and maven home path

3. Go to jenkins dashboard and install maven plugin: **Manage Plugins → Available Plugins → Maven Integration → Install without restart**

Name ↓

Maven Integration plugin 3.21

This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTs as well as the automated configuration of various Jenkins publishers such as Junit.

[Report an issue with this plugin](#)

Maven plugin

4. Go to **Manage Plugins → Global Tool Configuration → JDK installations → Save → Apply**

JDK installations ^

 Edited

JDK installations

List of JDK installations on this system

[Add JDK](#)

JDK

Name

java-11

JAVA_HOME

/usr/lib/jvm/java-11-openjdk-amd64



Install automatically 

[Add JDK](#)

Git installations

[Save](#)

[Apply](#)

5 . Go to Manage Plugins → Global Tool Configuration → Maven installations → Save → Apply

Maven

Maven installations ^  Edited

Maven installations

List of Maven installations on this system

[Add Maven](#)

Maven

Name

maven-3.9.1

MAVEN_HOME

/opt/maven



Install automatically



[Add Maven](#)

[Save](#)

[Apply](#)

Step-4. Setup Tomcat server in the vm

1. Create Gcp vm using ubuntu 22.04 version boot disk

2. Install Java using this commands:-

```
sudo apt update  
sudo apt install openjdk-11-jdk -y  
SSh into vm
```

Run following commands:

```
cd /opt  
wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.74/bin/apache-tomcat-9.0.  
tar -xvzf tar.gz file  
mv apache-tomcat-9.0.74 tomcat  
cd tomcat  
ll  
cd bin
```

Start tomcat server: ./startup.sh

3. Access tomcat server: Open a web browser, and navigate to an external IP address. Use this syntax:

http://ip_address_or_domain:8080

4. Setup tomcat credentials:

login into vm using ssh

Update context.xml file

```
cd /opt/tomcat  
find / -name context.xml
```

Output:

```
root@shubhi-test:/opt/tomcat# find / -name context.xml
/opt/tomcat/conf/context.xml
/opt/tomcat/webapps/examples/META-INF/context.xml
/opt/tomcat/webapps/manager/META-INF/context.xml
/opt/tomcat/webapps/host-manager/META-INF/context.xml
/opt/tomcat/webapps/docs/META-INF/context.xml
```

we need to update those files

Update those files:

```
nano /opt/tomcat/webapps/examples/META-INF/context.xml
```

```
GNU nano 6.2                               /opt/tomcat/webapps/examples/META-INF/context.xml
<?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.
-->
<Context>
    <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
        sameSiteCookies="strict" />
    <!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
        allow="127\\.\\d+\\.\\d+\\.\\d+|::1|0:0:0:0:0:0:1" /> -->
</Context>
```

```
nano /opt/tomcat/webapps/manager/META-INF/context.xml
```

```
GNU nano 6.2                               /opt/tomcat/webapps/manager/META-INF/context.xml
<?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.
-->
<Context antiResourceLocking="false" privileged="true" >
  <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
    sameSiteCookies="strict" />
  <!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\\.\\d+\\.\\d+\\.\\d+|::1|0:0:0:0:0:0:1" /> -->
  <Manager sessionAttributeValueClassNameFilter="java\\.lang\\.\\(?:Boolean|Integer|Long|Number|String)|org\\.apach>
</Context>
```

nano /opt/tomcat/webapps/host-manager/META-INF/context.xml

```
GNU nano 6.2                               /opt/tomcat/webapps/host-manager/META-INF/context.xml
<?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.
-->
<Context antiResourceLocking="false" privileged="true" >
  <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
    sameSiteCookies="strict" />
  <!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\\.\\d+\\.\\d+\\.\\d+|::1|0:0:0:0:0:0:1" /> -->
  <Manager sessionAttributeValueClassNameFilter="java\\.lang\\.\\(?:Boolean|Integer|Long|Number|String)|org\\.apach>
</Context>
```

nano /opt/tomcat/webapps/docs/META-INF/context.xml

```
GNU nano 6.2                               /opt/tomcat/webapps/docs/META-INF/context.xml
<?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

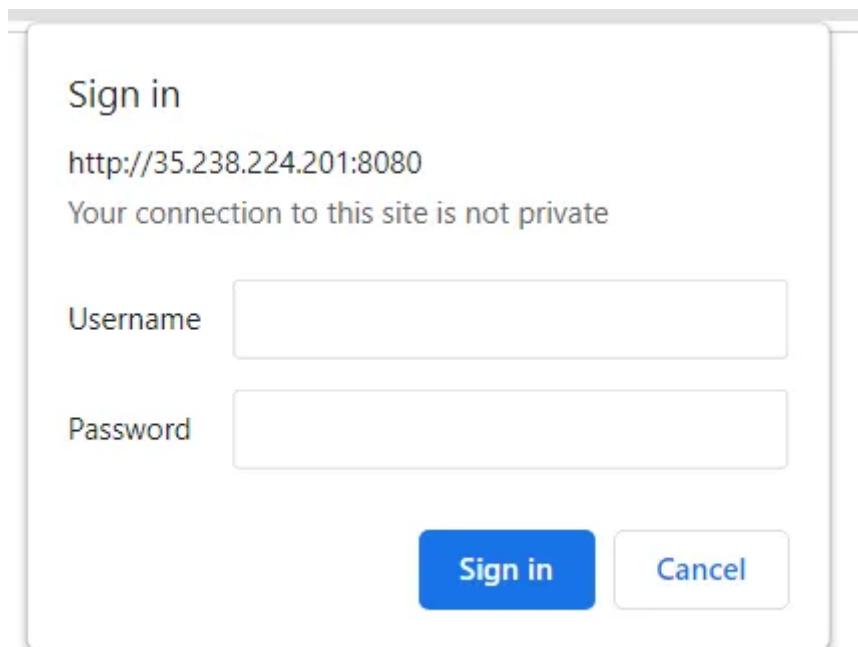
    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    See the License for the specific language governing permissions and
    limitations under the License.
-->
<Context antiResourceLocking="false" >
    <!-- I<Valve className="org.apache.catalina.valves.RemoteAddrValve"
        allow="127\\.\\d+\\.\\d+\\.\\d+|::1|0:0:0:0:0:0:0:1" /> -->
</Context>
```

Stop tomcat: ./shutdown.sh

Start again: ./startup.sh

Go to tomcat web page and click manager app ->

Asking for credentials →



```
cd /opt/tomcat/config
```

Update this file:

tomcat-users.xml

Give your credentials in this file: ADD this lines

```
<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<role rolename="manager-jmx"/>

<role rolename="manager-status"/>

<user username="admin" password="admin" roles="manager-gui, manager-script,
manager-jmx, manager-status"/>

<user username="deployer" password="deployer" roles="manager-script"/>

<user username="tomcat" password="s3cret" roles="manager-gui"/>

</tomcat-users>
```

```

GNU nano 6.2                               tomcat-users.xml
you must define such a user - the username and password are arbitrary.

Built-in Tomcat manager roles:
- manager-gui      - allows access to the HTML GUI and the status pages
- manager-script   - allows access to the HTTP API and the status pages
- manager-jmx       - allows access to the JMX proxy and the status pages
- manager-status    - allows access to the status pages only

The users below are wrapped in a comment and are therefore ignored. If you
wish to configure one or more of these users for use with the manager web
application, do not forget to remove the <!... ...> that surrounds them. You
will also need to set the passwords to something appropriate.

-->
<!--
<user username="admin" password="" roles="manager-gui"/>
<user username="robot" password="" roles="manager-script"/>
-->
<!--
The sample user and role entries below are intended for use with the
examples web application. They are wrapped in a comment and thus are ignored
when reading this file. If you wish to configure these users for use with the
examples web application, do not forget to remove the <!... ...> that surrounds
them. You will also need to set the passwords to something appropriate.

-->
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<role rolename="manager-jmx"/>
<role rolename="manager-status"/>
<user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>
<user username="deployer" password="deployer" roles="manager-script"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
</tomcat-users>

```

[]

^G Help **^O Write Out** **^W Where Is** **^K Cut** **^T Execute** **^C Location** **M-U Undo**
^X Exit **^R Read File** **^V Replace** **^U Paste** **^J Justify** **^/ Go To Line** **M-E Redo**

Save this file and login tomcat manager app

Step-5. Integrate tomcat with jenkins server

1. Go to jenkins dashboard and install maven plugin: **Manage Plugins** → **Available Plugins** → **Deploy to container** → **Install without restart**
2. Go to: **Manage credentials** → **System** → **global credentials** → **add credentials** → give username and password → **create**

Scope [?](#)

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

Username [?](#)

deployer

Treat username as secret [?](#)

Password [?](#)



Concealed

ID [?](#)

tomcat_deployer

Description [?](#)

Save

3. Username and password stored in this file tomcat-users.xml

Step-6. Setup credentials for email notifications

1. Go to: Manage Jenkins → configure system → at the last email notification option → SMTP server → smtp.gmail.com → Default user e-mail suffix → @gmail.com → Click advanced → Use SMTP Authentication → give email in username → give password → Use SSL → SMTP Port → 465

E-mail Notification

SMTP server

smtp.gmail.com

Default user e-mail suffix ?

@gmail.com

Advanced ^

 Edited

Use SMTP Authentication ?

User Name

shubhangi.thakur9494@gmail.com

Password



Concealed

Use SSL ?

Use TLS

Save

Apply

>Password

Concealed

Use SSL ?

Use TLS

SMTP Port ?

465

Reply-To Address

Charset

UTF-8

Test configuration by sending test e-mail

Save **Apply**

NOTE: You have to create app Password

2. Save → Apply

Step-7. Create maven project in jenkins

1. Click → New item → Enter name → Select maven → Ok

Enter an item name

spring-boot-app

» Required field



Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, commonly used for something other than software build.



Maven project

Build a maven project. Jenkins takes advantage of your POM files and



Pipeline

Orchestrates long-running activities that can span multiple build agents and/or organizing complex activities that do not easily fit in free-style.



Multi-configuration project

Suitable for projects that need a large number of different configurations builds, etc.



Folder

OK

Is a container that stores nested items in it. Useful for grouping things under a separate namespace, so you can have multiple things of the same name.

Github Repository: <https://github.com/thecloudside/hello-world-spring-boot-app>

2. Give description → source code management → Select git → give repository url → select branch → Build triggers → Select Build whenever a snapshot dependency is built → Select poll SCM → ****

General

Description

deploy spring boot app on the tomcat server using maven

[Plain text] [Preview](#)

- Commit agent's Docker container [?](#)
- Define a Docker template
- Discard old builds [?](#)
- GitHub project
- This project is parameterized [?](#)
- Throttle builds [?](#)
- Execute concurrent builds if necessary [?](#)

Source Code Management

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#)

`https://github.com/thecloudside/hello-world-testing.git`

Credentials [?](#)

- none -

Add ▾

Advanced ▾

Add Repository

Branches to build [?](#)

Save

Apply

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/master

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add ▾

Save

Apply

Build Triggers

- Build whenever a SNAPSHOT dependency is built [?](#)
- Schedule build when some upstream has no successful builds [?](#)
- Trigger builds remotely (e.g., from scripts) [?](#)
- Build after other projects are built [?](#)
- Build periodically [?](#)
- GitHub hook trigger for GITScm polling [?](#)

- Poll SCM [?](#)

Schedule [?](#)

```
* * * * *
```

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * * " to poll every hour.
Would last have run at Thursday, April 27, 2023 at 8:07:34 AM Coordinated Universal Time; would next run at Friday, April 28, 2023 at 8:07:34 AM Coordinated Universal Time.

Ignore post-commit hooks [?](#)

Save

Apply

3. Build → Root pom → give pom file path →

Build

Root POM ?

springapp/pom.xml

Goals and options ?

Advanced ▾

Post Steps

Run only if build succeeds

Run only if build succeeds or is unstable

Run regardless of build result

Should the post-build steps run only for successful builds, etc.

Save

Apply

4. springapp/pom.xml → Goals and options → clean install

5. Build Settings → Select email notification → Recipients → give email address → check all options

Build Settings

E-mail Notification

Recipients

shubhangithakur9494@gmail.com

Send e-mail for every unstable build

Send separate e-mails to individuals who broke the build

Send e-mail for each failed module ?

Post-build Actions

≡ Deploy war/ear to a container

WAR/EAR files ?

**/*.war

Save

Apply

6. Add Post-build actions → Select Deploy war/ear to a container → war/ear files → **/*.war → containers → select tomcat 8.x remote → select tomcat deployer credentials → give tomcat server url → <http://ipaddress:8080/>

Containers

≡ Tomcat 8.x Remote

Credentials

deployer/********

Add ▾

Tomcat URL ?

http://35.238.224.201:8080/

Advanced ^

Manager context path ?

Add Container ▾

Save

Apply

7. Save → Apply → Click on the build now

Console Output

```
Started by user shubhi thakur
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/maven-tomcat
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/maven-tomcat/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/thecloudside/hello-world-testing.git # timeout=10
Fetching upstream changes from https://github.com/thecloudside/hello-world-testing.git
> git --version # timeout=10
> git --version # 'git version 2.25.1'
> git fetch --tags --force --progress -- https://github.com/thecloudside/hello-world-testing.git
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 87a46e9222e0e29e955e4999f67f93c490b45566 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 87a46e9222e0e29e955e4999f67f93c490b45566 # timeout=10
Commit message: "Update application.properties"
First time build. Skipping changelog.
Parsing POMs
Established TCP socket on 40571
[springapp] $ /usr/lib/jvm/java-11-openjdk-amd64/bin/java -cp /var/lib/jenkins/plugins/maven-plugin-1.14.jar:/opt/maven/boot/plexus-classworlds-2.6.0.jar:/opt/maven/conf/logging jenkins.maven3.agent /var/cache/jenkins/war/WEB-INF/lib/remoting-3107.v665000b_51092.jar /var/lib/jenkins/plugins/maven-
```

```
. / \ \ _ _ _ ( ) _ _ _ \ \ \ \
( ( ) \ _ | ' _ | ' _ | ' _ \ \ _ | \ \ \ \
\ \ \ _ ) | ( _ ) | | | | | ( _ | | ) ) ) )
' | _ | . _ | | | | | \ _ , | / / / /
=====|_|=====|_|/_=/_/_/_/
:: Spring Boot ::      (v2.3.4.RELEASE)

2023-04-26 11:10:29.500  INFO 9194 --- [           main] c.e.s.Springapp1Application
testing-jk with PID 9194 (started by jenkins in /var/lib/jenkins/workspace/maven-tom
2023-04-26 11:10:29.503  INFO 9194 --- [           main] c.e.s.Springapp1Application
profiles: default
2023-04-26 11:10:31.817  INFO 9194 --- [           main] o.s.s.concurrent.ThreadPool
'applicationTaskExecutor'
2023-04-26 11:10:32.244  INFO 9194 --- [           main] c.e.s.Springapp1Application
seconds (JVM running for 4.994)
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 4.429 s - in c
2023-04-26 11:10:32.865  INFO 9194 --- [extShutdownHook] o.s.s.concurrent.ThreadPool
'applicationTaskExecutor'
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[JENKINS] Recording test results
[INFO]
[INFO] --- war:3.2.3:war (default-war) @ springapp1 ---
[INFO] Packaging webapp
[INFO] Assembling webapp [springapp1] in [/var/lib/jenkins/workspace/maven-tomcat/sprin
[INFO] Processing war project
[INFO] Webapp assembled in [118 msecs]
[INFO] Building war: /var/lib/jenkins/workspace/maven-tomcat/springapp/target/springap
```

8. See your Application successfully run

```
 .--_/\_--'_--_(_)_--_--_\\_--_\\_\\_\\_
(( )\__| '_| '_| | '_\`| \\_\\_\\_
\\_/_|_|_|_|_|_|_|_|(|_|_|_))_)
' |__| .__|_|_|_|_|_\_, | / / / /
=====|_|=====|_|/_=/_/_/_/
:: Spring Boot ::      (v2.3.4.RELEASE)

2023-04-26 11:10:29.500  INFO 9194 --- [           main] c.e.s.Springapp1Application
testing-jk with PID 9194 (started by jenkins in /var/lib/jenkins/workspace/maven-tom
2023-04-26 11:10:29.503  INFO 9194 --- [           main] c.e.s.Springapp1Application
profiles: default
2023-04-26 11:10:31.817  INFO 9194 --- [           main] o.s.s.concurrent.ThreadPool
'applicationTaskExecutor'
2023-04-26 11:10:32.244  INFO 9194 --- [           main] c.e.s.Springapp1Application
seconds (JVM running for 4.994)
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 4.429 s - in c
2023-04-26 11:10:32.865  INFO 9194 --- [extShutdownHook] o.s.s.concurrent.ThreadPool
'applicationTaskExecutor'
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[JENKINS] Recording test results
[INFO]
[INFO] --- war:3.2.3:war (default-war) @ springapp1 ---
[INFO] Packaging webapp
[INFO] Assembling webapp [springapp1] in [/var/lib/jenkins/workspace/maven-tomcat/sprin
[INFO] Processing war project
[INFO] Webapp assembled in [118 msecs]
[INFO] Building war: /var/lib/jenkins/workspace/maven-tomcat/springapp/target/springap
```

9. Go to tomcat web page → manager app →

Apache Tomcat/9.0.74



If you're seeing this, you've successfully installed Tomcat. Congratulations!



Recommended Reading:
[Security Considerations How-To](#)
[Manager Application How-To](#)
[Clustering/Session Replication How-To](#)

[Server Status](#)[Manager App](#)[Host Manager](#)

Developer Quick Start

[Tomcat Setup](#)[First Web Application](#)[Realms & AAA](#)[JDBC DataSources](#)[Examples](#)[Servlet Specifications](#)[Tomcat Versions](#)

Managing Tomcat

For security, access to the [manager webapp](#) is

Documentation

[Tomcat 9.0 Documentation](#)

Getting Help

[FAQ and Mailing Lists](#)

10. springapp1-0.0.1-SNAPSHOT → type /hello → you can see application output

Message: OK

Manager

List Applications		HTML Manager Help		Manager Help	
Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/jsp_app	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/springapp1-0.0.1-SNAPSHOT	None specified		true	0	Start Stop Reload Undeploy Activate Windows settings Expire sessions with idle ≥ 30 minutes

Hello World



Not secure

Hello World

All steps are done successfully. Hope you found it interesting!

Happy Learning Dev-Ops!