

Auto Deployment of Application on Apache Tomcat using Jenkins

1. Objective

To automatically deploy a Java web application (WAR file) on Apache Tomcat using Jenkins, without manual intervention, by configuring Jenkins triggers (CRON).

2. Tools & Technologies Used

- Apache Tomcat 9
- Jenkins
- Deploy to Container Jenkins Plugin
- Java WAR file
- Linux (Ubuntu)
- CRON Scheduler

3. Concept of Auto Deployment

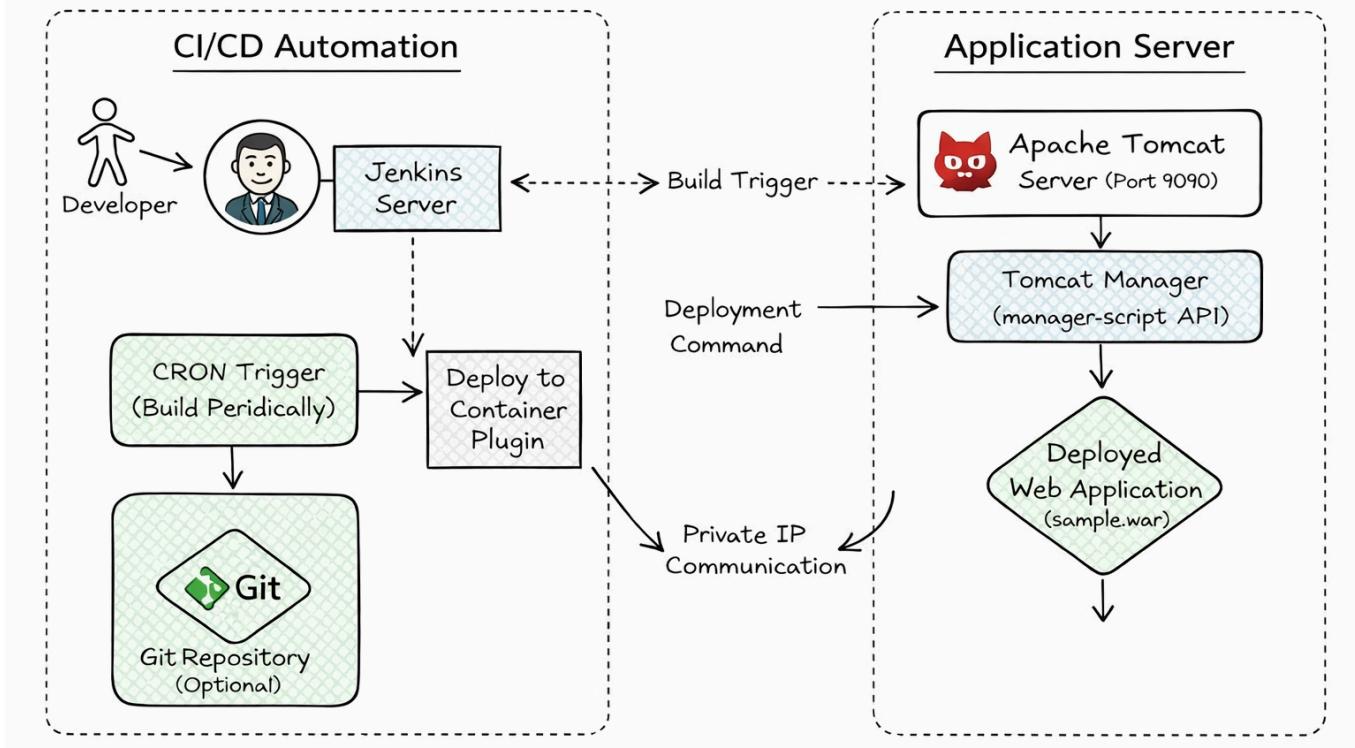
Auto deployment is the process of deploying application code to a server **automatically** whenever a job is triggered, without manually copying files or restarting services.

In this setup:

- Jenkins acts as the **automation tool**
- Tomcat Manager acts as the **deployment endpoint**
- CRON triggers Jenkins at fixed intervals
- Deployment happens automatically

4. Architecture Diagram

Auto Deployment Architecture using Jenkins and Apache Tomcat



5. Apache Tomcat Setup

5.1 Install Tomcat

download a sample war file from this [link](#)

```
sudo su
cd /opt
wget https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample
tar -xvzf apache-tomcat-9.0.113.tar.gz
```

5.2 Change Tomcat Port

Edit:

```
nano conf/server.xml
```

Change:

```
<Connector port="8080">
```

to:

```
<Connector port="9090">
```

```
<!-- 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"  
maxParameterCount="1000"  
/>  
<!-- A "Connector" using the shared thread pool-->  
<!--  
<Connector executor="tomcatThreadPool"  
port="8080" protocol="HTTP/1.1"  
connectionTimeout="20000"  
redirectPort="8443"  
maxParameterCount="1000"  
/>  
-->  
1. Define an SSL/TLS HTTP/1.1 Connector on port 8443
```

5.3 Start Tomcat

```
cd bin  
./startup.sh
```

Access:

```
http://<IP>:9090
```

Apache Tomcat/9.0.113

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](#)

Developer Quick Start

Tomcat Setup	Realms & AAA	Examples	Servlet Specifications
First Web Application	JDBC DataSources		Tomcat Versions

6. Tomcat Manager Configuration

6.1 Configure Users and Roles

Edit:

```
nano conf/tomcat-users.xml
```

Add:

```
<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"/>
```

```
GNU nano 6.2                                     conf/tomcat-users.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.
-->
<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">

  <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"/>
```

6.2 Enable Remote Deployment

Edit:

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

Use:

```
<Context privileged="true">
<CookieProcessor
className="org.apache.tomcat.util.http.Rfc6265CookieProcessor" />
<!-- Allow remote Jenkins access -->
<!--      <Valve className="org.apache.catalina.valves.RemoteCIDRValve"
allow="127.0.0.0/8,::1/128" />      -->

</Context>
```

```
GNU nano 6.2                                         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.RemoteCIDRValve"
    allow="127.0.0.0/8,::1/128" />
-->
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Numbe
</Context>
```

Restart Tomcat:

```
./shutdown.sh
./startup.sh
```

7. Jenkins Setup

7.1 Install Jenkins

(Pre-installed or installed via package manager)

Verify:

```
systemctl status jenkins
```

7.2 Install Required Plugin

- Go to **Manage Jenkins → Plugins**
 - Install **Deploy to Container Plugin**
-

8. Jenkins Job Configuration

8.1 Create Job

- New Item → **Freestyle Project**
- Job name: **Tomcat-Auto-Deployment**

8.2 Build Step

Add **Execute shell**:

```
pwd
```

run the job now the workspace will be created , in that place the sample app war file in this workspace of your job

WAR file should exist in workspace:

```
/var/lib/jenkins/workspace/Tomcat/sample.war
```

8.3 Post Build Action (Deployment)

Add **Deploy war/ear to a container**

Field	Value
WAR/EAR files	**/*.war
Context Path	sample
Container	Tomcat 9.x
Tomcat URL	http://<IP>:9090
Credentials	deployer / deployer

**Configure**[General](#)[Source Code Management](#)[Triggers](#)[Environment](#)[Build Steps](#)[Post-build Actions](#)**Post-build Actions**

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

[Deploy war/ear to a container](#)

WAR/EAR files ?

**/*.war

Context path ?

sample

Containers[Tomcat 9.x Remote](#)**Credentials**

deployer*****

+ Add

Tomcat URL ?

http://100.25.98.116:9090/

Advanced ▾

+ Add Container

Save

Apply

9. Auto Deployment Using CRON

9.1 Enable CRON Trigger

- Go to Job → Configure
- Check **Build periodically**

9.2 CRON Expression

Example (every 5 minutes):

```
* * * * *
```

This means Jenkins automatically:

- Runs the job
- Deploys the WAR file

- Refreshes the application on Tomcat

The screenshot shows the 'Triggers' section of a Jenkins job configuration. It includes the following options:

- Trigger builds remotely (e.g., from scripts) ?
- Build after other projects are built ?
- Build periodically ?

The 'Schedule' field contains the CRON expression: `*****`. A warning message below it states: "⚠ Do you really mean 'every minute' when you say '*****'? Perhaps you meant 'H * * * *' to poll once per hour". It also indicates the last run was on Sunday, December 21, 2025, at 7:46:00 PM Coordinated Universal Time, and the next run is scheduled for Sunday, December 21, 2025, at 7:47:00 PM Coordinated Universal Time.

Other trigger options shown are:

- GitHub hook trigger for GITScm polling ?
- Poll SCM ?

10. How Auto Deployment Works (Explanation)

When the CRON trigger runs:

1. Jenkins job starts automatically
2. Jenkins connects to Tomcat Manager
3. Old application is undeployed
4. New WAR file is deployed
5. Application is live without manual steps

This completes **auto deployment**.

11. Verification

Jenkins:

- Build shows **SUCCESS**

Builds >		...
<input type="text"/> Filter		/
Today		
<input checked="" type="checkbox"/>	#43 7:50 PM	▼
<input checked="" type="checkbox"/>	#42 7:49 PM	▼
<input checked="" type="checkbox"/>	#41 7:48 PM	▼
<input checked="" type="checkbox"/>	#40 7:47 PM	▼
<input checked="" type="checkbox"/>	#39 7:46 PM	▼
<input checked="" type="checkbox"/>	#38 7:45 PM	▼
<input checked="" type="checkbox"/>	#37 7:44 PM	▼
<input checked="" type="checkbox"/>	#36 7:43 PM	▼
<input checked="" type="checkbox"/>	#35 7:42 PM	▼
<input checked="" type="checkbox"/>	#34 7:41 PM	▼
<input checked="" type="checkbox"/>	#33 7:40 PM	▼

Tomcat:

<http://<IP>:9090/sample>

Application loads successfully.



Sample "Hello, World" Application

This is the home page for a sample application used to illustrate the source directory organization of a web application utilizing the prin

To prove that they work, you can execute either of the following links:

- To a [JSP page](#).
- To a [servlet](#).

Conclusion

This project demonstrates **auto deployment of applications using Jenkins and Apache Tomcat**. By configuring Jenkins triggers and Tomcat Manager, application deployment is fully automated, enabling continuous delivery and DevOps best practices.