# Jenkins CI/CD Pipeline for WAR Deployment

This document provides a detailed guide for setting up and running a Jenkins CI/CD pipeline to build, analyze, publish, and deploy a WAR file to a Tomcat server with Nexus artifact storage and SonarQube analysis.

## Prerequisites

Ensure the following requirements are met before running the pipeline:

1. 1. Jenkins server installed and running.
2. 2. Maven installed and configured in Jenkins (Manage Jenkins → Global Tool Configuration).
3. 3. SonarQube server up and running with a project created.
4. 4. Nexus Repository Manager 3 running and configured with a 'maven-releases' repository.
5. 5. Tomcat server running on the target deployment server.
6. 6. SSH access configured between Jenkins server and Tomcat server.
7. 7. RSA private key stored on Jenkins at /var/lib/jenkins/.ssh/jenkins\_key with permissions 600.
8. 8. Nexus credentials stored in Jenkins Credentials (ID: nexus\_creds).
9. 9. SonarQube token stored in Jenkins Credentials (ID: sonar\_creds).
10. 10. Correct groupId, artifactId, and version defined in pom.xml.

## File Preparation

1. Copy the SSH private key to Jenkins server:

sudo mkdir -p /var/lib/jenkins/.ssh  
sudo cp jenkins\_key /var/lib/jenkins/.ssh/  
sudo chown jenkins:jenkins /var/lib/jenkins/.ssh/jenkins\_key  
sudo chmod 600 /var/lib/jenkins/.ssh/jenkins\_key

2. Ensure the public key is added to the ~/.ssh/authorized\_keys file on the Tomcat server.

## Pipeline Stages

### 1. Build WAR

This stage compiles the project and packages it into a WAR file using Maven:

mvn clean package -DskipTests

### 2. SonarQube Analysis

This stage runs code quality analysis with SonarQube:

mvn sonar:sonar -Dsonar.projectKey=wwp -Dsonar.host.url=http://<SONAR\_HOST>:9000 -Dsonar.login=<SONAR\_TOKEN> -Dsonar.java.binaries=target/classes

### 3. Extract Version

Extracts the version from pom.xml using Maven Help Plugin:

mvn help:evaluate -Dexpression=project.version -q -DforceStdout

### 4. Publish to Nexus

Uploads the generated WAR file to Nexus repository:

nexusArtifactUploader(  
 nexusVersion: 'nexus3',  
 protocol: 'http',  
 nexusUrl: '<NEXUS\_URL>',  
 groupId: 'koddas.web.war',  
 artifactId: 'wwp',  
 version: '<ART\_VERSION>',  
 repository: 'maven-releases',  
 credentialsId: 'nexus\_creds',  
 artifacts: [[ file: '<WAR\_FILE>', type: 'war' ]]  
)

### 5. Deploy to Tomcat

Copies the WAR file to the Tomcat server and restarts Tomcat:

scp -i /var/lib/jenkins/.ssh/jenkins\_key -o StrictHostKeyChecking=no <WAR\_FILE> ubuntu@<TOMCAT\_SERVER>:/tmp/  
ssh -i /var/lib/jenkins/.ssh/jenkins\_key -o StrictHostKeyChecking=no ubuntu@<TOMCAT\_SERVER> '  
 sudo mv /tmp/\*.war /opt/tomcat/webapps/ && sudo systemctl restart tomcat  
'

### 6. Display URLs

Displays application and Nexus artifact URLs for reference.

## Post Actions

✅ Success: Prints success message.  
❌ Failure: Prints failure message.