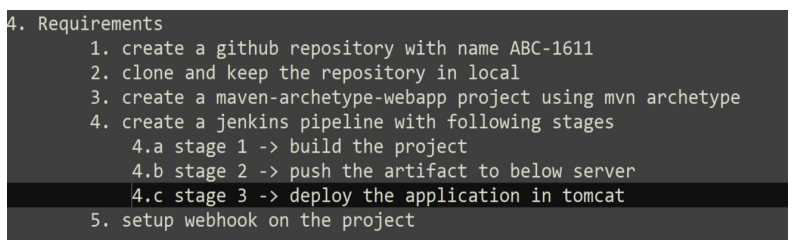
**Name: Manjunath N Evaluation-4 Date-19/04/2025**

****

****

**1. Create a GitHub Repository with Name ABC-1611**

**Go to GitHub**: [GitHub](https://github.com/).

**Create New Repository**:

Click the **+** button at the top right, then select **New repository**.

Set the **Repository name** to **ABC-11**.

Choose **public/private** as per your needs.

You can initialize with a README or keep it empty for now.

**2. Clone and Keep the Repository Locally**

git clone https://github.com/ManjunathNBadri/ABC-11.git

cd ABC-1611

* **Add token**

git remote set-url origin https://ManjunathNBadri: ghp\_Kc0jNSEFGeiYwWp2Yf98XLUPVt2GPH1apjph @ https://github.com/ManjunathNBadri/ABC-1611.git

* **Add file and push to remote repo**

git add .

git commit –m “adding file”

Git push –u origin master

**3. Create a maven-archetype-webapp project using mvn archetype**

RUN THIS CODE on WSL in cloned folder i.e ABC-1611

mvn archetype:generate \

-DgroupId=com.example.webapp \

-DartifactId=my-webapp \

-DarchetypeArtifactId=maven-archetype-webapp \

-DinteractiveMode=false

* **artifactId**: This is the name of your project, in this case, ABC-1611-my-webapp.

**4. Create a jenkins pipeline with following stages**

**4.a stage 1 -> build the project**

**4.b stage 2 -> push the artifact to below server**

**4.c stage 3 -> deploy the application in tomcat**

**#pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>my-webapp</artifactId>

<version>1.0-SNAPSHOT</version>

<packaging>war</packaging>

<name>my-webapp Maven Webapp</name>

<url>http://maven.apache.org</url>

<properties>

<maven.compiler.source>17</maven.compiler.source>

<maven.compiler.target>17</maven.compiler.target>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<failOnMissingWebXml>false</failOnMissingWebXml>

</properties>

<dependencies>

<!-- JUnit for testing -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<!-- Servlet API (provided by server like Tomcat) -->

<dependency>

<groupId>jakarta.servlet</groupId>

<artifactId>jakarta.servlet-api</artifactId>

<version>5.0.0</version>

<scope>provided</scope>

</dependency>

<!-- JSTL support (optional) -->

<dependency>

<groupId>jakarta.servlet.jsp.jstl</groupId>

<artifactId>jakarta.servlet.jsp.jstl-api</artifactId>

<version>3.0.0</version>

</dependency>

<dependency>

<groupId>org.glassfish.web</groupId>

<artifactId>jakarta.servlet.jsp.jstl</artifactId>

<version>3.0.1</version>

</dependency>

</dependencies>

<build>

<finalName>my-webapp</finalName>

<plugins>

<!-- Modern WAR plugin -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-war-plugin</artifactId>

<version>3.4.0</version>

</plugin>

<!-- Compiler plugin for Java 17 -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.10.1</version>

<configuration>

<release>17</release>

</configuration>

</plugin>

</plugins>

</build>

</project>

**4 #Jenkinsfile**

pipeline {

agent any

environment {

REPO\_URL = 'https://github.com/ManjunathNBadri/ABC-1611.git'

REPO\_ID = 'adikarthikgupta'

ARTIFACT\_ID = 'my-webapp'

ARTIFACT\_VERSION = '1.0-SNAPSHOT'

WAR\_PATH = '/var/lib/jenkins/workspace/n1/my-webapp/target/my-webapp.war'

REPO\_URL\_MAVEN = 'https://pkgs.dev.azure.com/adikarthikgupta/\_packaging/adikarthikgupta/maven/v1'

TOMCAT\_DIR = '/home/azureuser/tomcat/webapps'

TOMCAT\_SHUTDOWN = '/home/azureuser/tomcat/bin/shutdown.sh'

TOMCAT\_STARTUP = '/home/azureuser/tomcat/bin/startup.sh'

}

stages {

stage('Clone Repository') {

steps {

echo "🔁 Cloning source code from GitHub..."

checkout([

$class: 'GitSCM',

branches: [[name: "refs/heads/main"]],

userRemoteConfigs: [[

url: "${REPO\_URL}",

credentialsId: "github-credentials" // Ensure this credential is stored in Jenkins

]]

])

}

}

stage('Build WAR with Maven') {

steps {

dir('my-webapp') {

echo "Building WAR file..."

sh 'mvn clean package -DskipTests'

sh 'ls -lh target/'

}

}

}

stage('Upload WAR to Azure Artifacts') {

steps {

withCredentials([string(credentialsId: 'azure-pat', variable: 'AZURE\_PAT')]) {

dir('my-webapp') {

echo "Creating temp settings.xml for Azure Artifacts..."

script {

writeFile file: 'settings-temp.xml', text: """

<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 https://maven.apache.org/xsd/settings-1.0.0.xsd">

<servers>

<server>

<id>${REPO\_ID}</id>

<username>${REPO\_ID}</username>

<password>${AZURE\_PAT}</password>

</server>

</servers>

</settings>

"""

}

echo "📦 Uploading WAR to Azure Artifacts..."

sh """

mvn deploy:deploy-file -s settings-temp.xml \

-Dfile=target/my-webapp.war \

-DgroupId=com.example \

-DartifactId=${ARTIFACT\_ID} \

-Dversion=${ARTIFACT\_VERSION} \

-Dpackaging=war \

-DrepositoryId=${REPO\_ID} \

-Durl=${REPO\_URL\_MAVEN}

"""

}

}

}

}

stage('Deploy to Tomcat') {

steps {

echo "🚀 Deploying WAR to Tomcat..."

sh """

if pgrep -f tomcat; then

echo "Stopping Tomcat..."

${TOMCAT\_SHUTDOWN}

sleep 5

else

echo "Tomcat is already stopped."

fi

echo "Copying WAR file..."

cp ${WAR\_PATH} ${TOMCAT\_DIR}/

echo "Starting Tomcat..."

${TOMCAT\_STARTUP}

"""

}

}

}

post {

success {

echo '✅ Deployment completed successfully!'

}

failure {

echo '❌ Pipeline failed. Check logs for details.'

}

always {

echo '🧹 Cleanup (if needed) complete.'

}

}

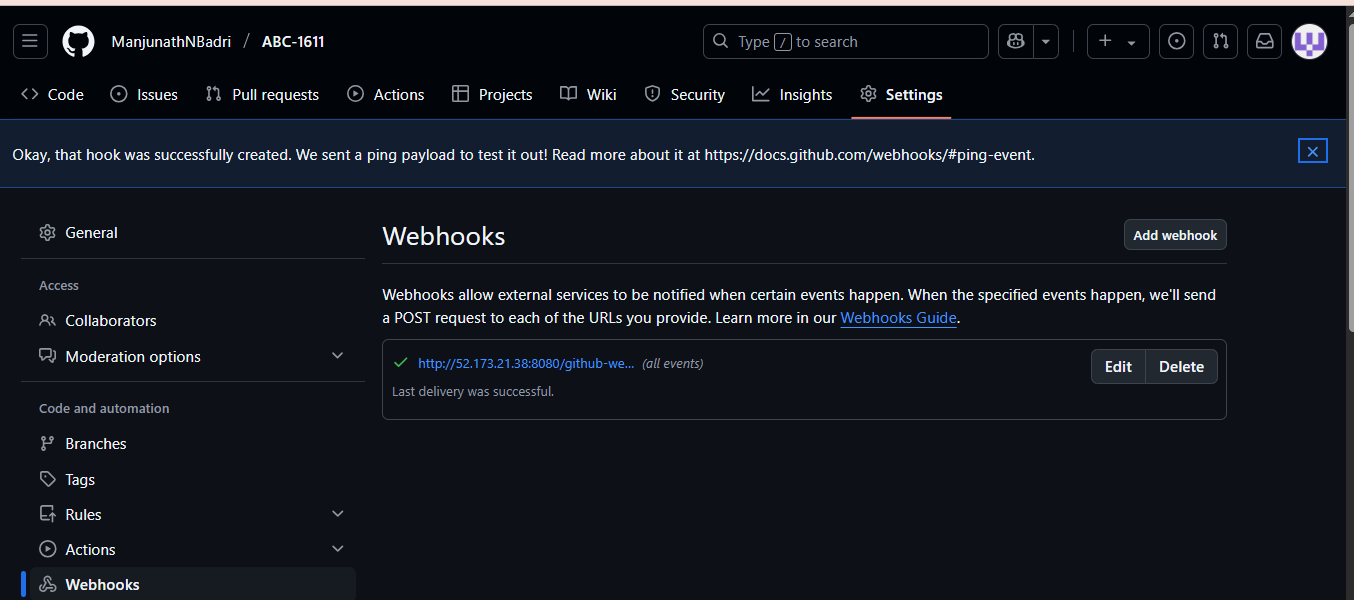
}

**5. Set up Web hook on GitHub**

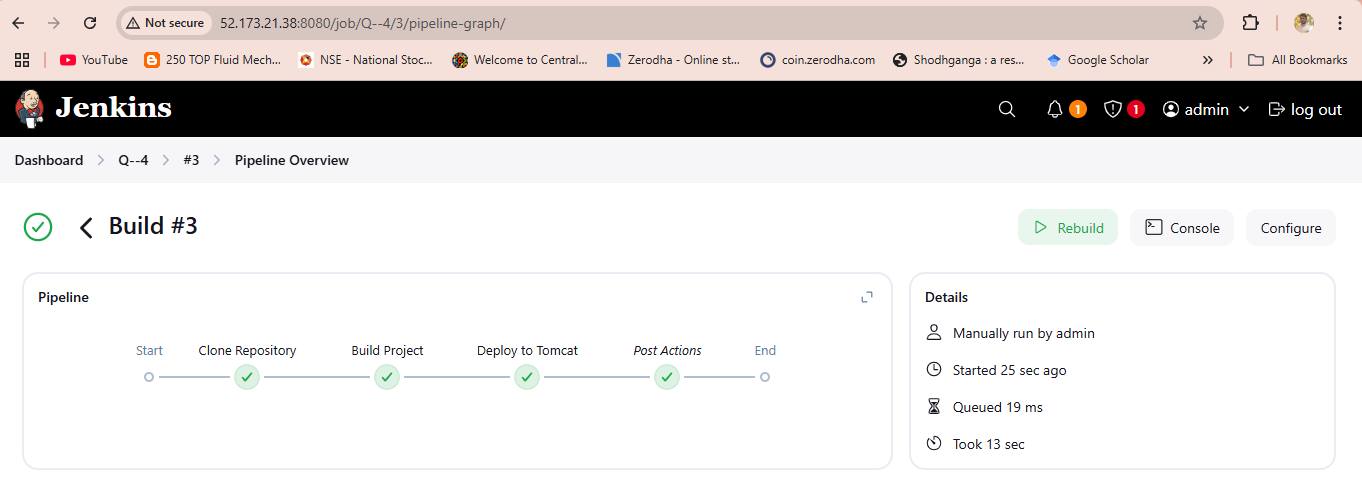
Finally, set up a **GitHub webhook** to trigger the Jenkins pipeline when code is pushed to the repository:

* **Go to your GitHub Repository**: <https://github.com/ManjunathNBadri/ABC-1611>
* **Navigate to Settings** > **Webhooks**.
* Click **Add webhook**.
* For the **Payload URL**, enter your Jenkins server's webhook URL, like:
* <http://52.173.21.38:8080/github-webhook/>
* Set **Content type** to application/json.
* Select **Just the push event** (or other events you want to trigger the pipeline).
* Click **Add webhook**.

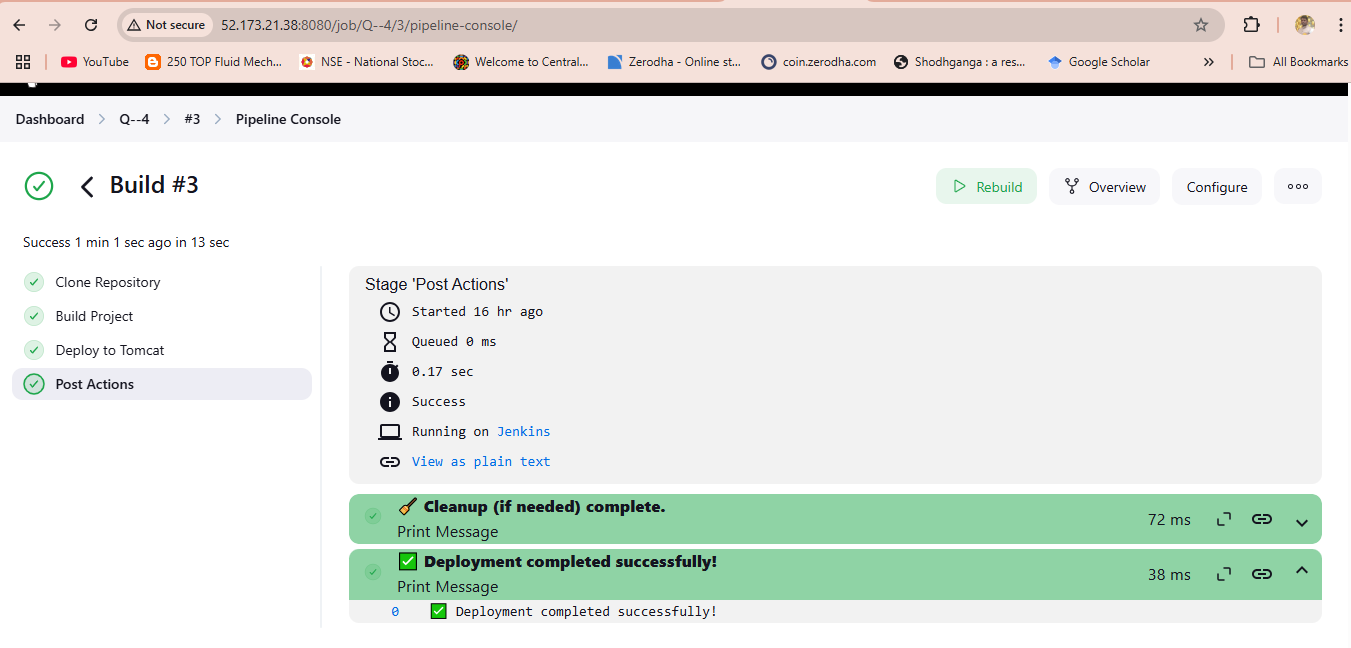
[**Webhook**](http://52.173.21.38:8080/job/Q--4/3/pipeline-graph/)  **Setting**



[**Pipeline Overview**](http://52.173.21.38:8080/job/Q--4/3/pipeline-graph/)



**Deployment completed successfully!**



* <http://52.173.21.38:8060/my-webapp/>

**Tomcat output**

