

# CI/CD Infrastructure with Jenkins Master and Dedicated Nodes (Nexus, SonarQube, Tomcat)

set up a CI/CD infrastructure with Jenkins as the master orchestrator and three nodes serving specific roles:

Nexus: Artifact repository (stores build outputs, dependencies)

SonarQube: Code quality and security analysis

Tomcat: Application deployment/runtime server

Instances (4) <a href="#">Info</a>		Last updated less than a minute ago		<a href="#">Instance state</a> ▾	<a href="#">Actions</a> ▾	<a href="#">Launch instances</a>	▼
		<input type="text"/> Find Instance by attribute or tag (case-sensitive)		All states ▾			
		<a href="#">Instance state = running</a>					
□	Name ⚡	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
□	sai@jenkins	i-09029c8b961b762a6	Running	t3.micro	3/3 checks passed	<a href="#">View alarms</a> +	eu-north-1
□	sai@tomcat	i-05554257b839cae8a	Running	t3.small	Initializing	<a href="#">View alarms</a> +	eu-north-1
□	sai@nexus	i-0185ebe6fb98f610c	Running	c7i-flex.large	Initializing	<a href="#">View alarms</a> +	eu-north-1
□	sai@sonar	i-0fccb2c432c405be2	Running	c7i-flex.large	3/3 checks passed	<a href="#">View alarms</a> +	eu-north-1

## Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

.....

# Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

## Install suggested plugins

Install plugins the Jenkins community finds most useful.

## Select plugins to install

Select and install plugins most suitable for your needs.

## Getting Started

---

# Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

 Jenkins

+ New Item

Build History

Build Queue  
No builds in the queue.

Build Executor Status  
0/2

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

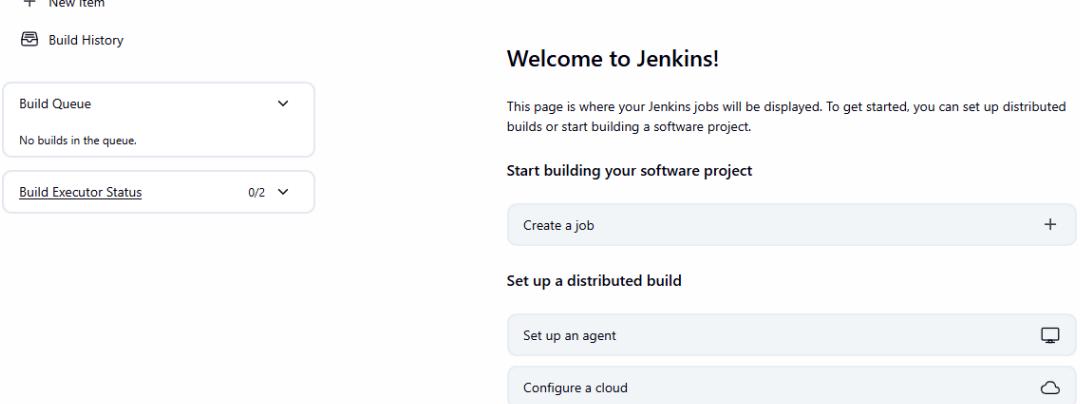
Create a job +

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds 



 Jenkins / Manage Jenkins / Plugins

Plugins

Updates

Available plugins 

Installed plugins

Advanced settings

Download progress

Search bar: g

Sonar Quality Gates 352.vdcdb\_d7994fb\_6  
Library plugins (for use by other plugins) analysis Other Post-Build Actions 8 mo 13 days ago 96

Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed")

Maven Integration 3.27  
Build Tools 2 mo 23 days ago 97

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.

Nexus Artifact Uploader 2.14  
Artifact Uploaders 2 yr 11 mo ago 77

This plugin to upload the artifact to Nexus Repository.  
This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.

GitHub Integration 0.7.2  
emailist Build Triggers 9 mo 22 days ago 67

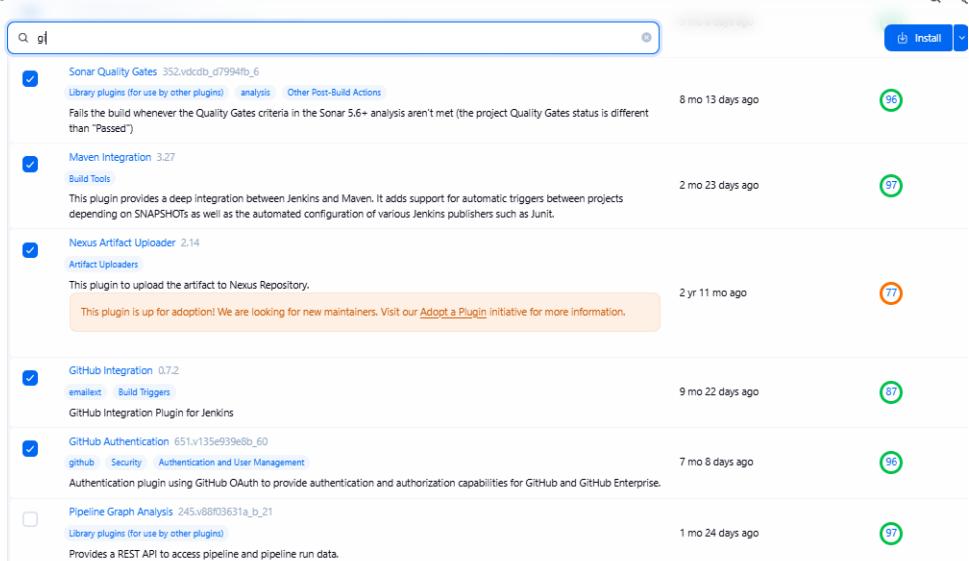
GitHub Integration Plugin for Jenkins

GitHub Authentication 651v135e939e8b\_60  
github Security Authentication and User Management 7 mo 8 days ago 96

Authentication plugin using GitHub OAuth to provide authentication and authorization capabilities for GitHub and GitHub Enterprise.

Pipeline Graph Analysis 245v8803631a\_b\_21  
Library plugins (for use by other plugins) 1 mo 24 days ago 97

Provides a REST API to access pipeline and pipeline run data.





Default global settings provider

[Use default maven global settings](#)

## JDK installations

[+ Add JDK](#)[≡ JDK](#)

Name

! Required

JAVA\_HOME

 Install automatically [?](#)[+ Add JDK](#)

## Git installations

[Save](#)[Apply](#)

## Install Java

```
sudo apt update  
sudo apt install openjdk-17-jdk -y  
java -version
```

```
cd /opt  
sudo wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.9.4.87374.zip
```

```
sudo apt install unzip -y  
sudo unzip sonarqube-9.9.4.87374.zip
```

```
sudo mv sonarqube-9.9.4.87374 sonarqube  
sudo chown -R $USER:$USER /opt/sonarqube
```

```
cd /opt/sonarqube/bin/linux-x86-64  
../sonar.sh start
```

-  Not secure 13.60.190.18:9000/sessions/new?return\_to=%2F

## Log in to SonarQube

This account should not use the default password.

### Enter a new password

All fields marked with \* are required

**Old Password \***

**New Password \***

**Confirm Password \***

The screenshot shows the SonarQube Administration interface. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and a search bar. The Administration tab is selected. Below the navigation, there are tabs for Configuration, Security (which is selected), Projects, System, and Marketplace. The main content area is titled "Users" and contains a sub-instruction "Create and administer individual users." A search bar is present. A table lists one user: "Administrator admin" (with a green 'A' icon). The table columns are SCM Accounts, Last connection, Groups, and Tokens. The user has "sonar-administrators" and "sonar-users" groups, 0 tokens, and was last connected "1 hour ago". A "Create User" button is located in the top right corner of the table area.

```
ubuntu@jenkins:~/.ssh$ ^C
ubuntu@jenkins:~/.ssh$ ssh ubuntu@13.60.190.18
The authenticity of host '13.60.190.18 (13.60.190.18)' can't be established.
ED25519 key fingerprint is SHA256:LkLU32KMZ22kWejqUWRlumTk+2Qm/kXd5gKbiBx5fms .
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.60.190.18' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Mon Nov  3 06:18:59 UTC 2025

  System load:  0.04              Temperature:          -273.1 °C
  Usage of /:   52.2% of 6.71GB  Processes:             154
  Memory usage: 6%                Users logged in:      1
  Swap usage:   0%                IPv4 address for enp39s0: 172.31.40.5

Expanded Security Maintenance for Applications is not enabled.

15 updates can be applied immediately.
5 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Mon Nov  3 06:12:49 2025 from 14.195.14.22
ubuntu@sonar:~$ exit
Logout
Connection to 13.60.190.18 closed
```

## Credentials

T	P	Store ↓	Domain	ID	Name
		System	(global)	sonar	sonar
		System	(global)	75ac2b33-3ca9-4e6e-aaf8-8ff5f10c34e6	jenkins
		System	(global)	ssh-agent	ubuntu

## Nodes

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	3.58 GB	0 B	3.58 GB	0ms
	Data obtained	16 min	16 min	16 min	16 min	16 min	16 min

Icon: S M L

Legend

## New node

Node name

sonar

Type

 Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

**Create**

**Jenkins** / Manage Jenkins / Nodes

---

Remote root directory ?  
/home/ubuntu/jenkins

---

Labels ?  
sonar

---

Usage ?  
Use this node as much as possible

---

Launch method ?  
Launch agents via SSH

---

Host ?  
13.60.190.18

---

Credentials ?  
ubuntu

---

Host Key Verification Strategy ?  
Non verifying Verification Strategy

---

Advanced ▾

---

Availability ?  
Keep this agent online as much as possible

---

**Node Properties**

- Disable deferred wipeout on this node ?
- Disk Space Monitoring Thresholds
- Environment variables
- Tool Locations

---

**Save**

**Jenkins** / Manage Jenkins / Nodes

**Nodes**

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	3.58 GiB	0 B	3.58 GiB	0ms
	sonar	Linux (amd64)	In sync	3.17 GiB	0 B	3.17 GiB	35ms
	Data obtained		3 sec	3 sec	3 sec	3 sec	3 sec

Icon: S M L

Legend

```
sudo apt install openjdk-17-jre-headless
```

```
sudo wget https://download.sonatype.com/nexus/3/nexus-3.85.0-03-linux-x86\_64.tar.gz
```

```
tar -xvf nexus-3.85.0-03-linux-x86_64.tar.gz
```

Jenkins / Manage Jenkins / Nodes

### Nodes

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time		
💻	Built-In Node	Linux (amd64)	In sync	3.58 GiB	!	0.8	3.58 GiB	0ms	⚙️
💻	nexus	Linux (amd64)	In sync	3.47 GiB	!	0.8	3.47 GiB	33ms	⚙️
💻	sonar	Linux (amd64)	In sync	3.16 GiB	!	0.8	3.16 GiB	15ms	⚙️
	Data obtained	0.19 sec	0.19 sec	0.19 sec	0.19 sec	0.19 sec	0.19 sec	0.19 sec	

Jenkins / Manage Jenkins / Nodes

### Nodes

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time		
💻	Built-In Node	Linux (amd64)	In sync	3.44 GiB	!	0.8	3.44 GiB	0ms	⚙️
💻	nexus	Linux (amd64)	In sync	3.47 GiB	!	0.8	3.47 GiB	28ms	⚙️
💻	sonar	Linux (amd64)	In sync	3.16 GiB	!	0.8	3.16 GiB	24ms	⚙️
💻	tomcat	Linux (amd64)	In sync	4.44 GiB	!	0.8	4.44 GiB	73ms	⚙️
	Data obtained	0.42 sec	0.41 sec	0.36 sec	0.35 sec	0.36 sec	0.35 sec	0.35 sec	

Icon: S M L

Legend

## New Item

Enter an item name

1st.job

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, follows steps like archiving artifacts and sending email notifications.



Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (pipelines) and/or organizing complex activities that do not easily fit in free-style job type.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments or platform-specific builds, etc.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which creates a separate namespace, so you can have multiple things of the same name as long as they're in different folders.



Multibranch Pipeline

OK

```
pipeline { agent { label 'sonar' }

tools {
    jdk 'JDK17'
    maven 'Maven'
}

environment {
    SONARQUBE_SERVER = 'sonar'          // Must match Jenkins → Configure System → SonarQube name
}
```

```

MVN_SETTINGS = '/etc/maven/settings.xml'
NEXUS_URL = 'http://13.62.99.225:8081'
NEXUS_REPO = 'maven-releases'
NEXUS_GROUP = 'com.web.cal'
NEXUS_ARTIFACT = 'webapp-add'
TOMCAT_URL = 'http://16.170.35.175:8082/manager/text'
}

stages {

/* === Stage 1: Checkout Code === */
stage('Checkout Code') {
    steps {
        echo '📦 Cloning source from GitHub...'
        checkout([
            $class: 'GitSCM',
            branches: [[name: '*/main']],
            userRemoteConfigs: [
                url: 'https://github.com/mrtechreddy/Java-Web-Calculator-App.git'
            ]
        ])
    }
}

/* === Stage 2: SonarQube Analysis === */
stage('SonarQube Analysis') {
    steps {
        echo '🔍 Running SonarQube static analysis...'
        withSonarQubeEnv("${SONARQUBE_SERVER}") {
            sh 'mvn clean verify sonar:sonar -DskipTests --settings ${MVN_SETTINGS}'
        }
    }
}

/* === Stage 3: Build Artifact === */
stage('Build Artifact') {
    steps {
        echo '⚙️ Building WAR...'
        sh 'mvn clean package -DskipTests --settings ${MVN_SETTINGS}'
        sh 'echo ✅ Build Completed!'
        sh 'ls -lh target/*.war || echo "No WAR file found."'
    }
}

/* === Stage 4: Upload Artifact to Nexus === */
stage('Upload Artifact to Nexus') {
    steps {
}

```

```

        withCredentials([usernamePassword(credentialsId: 'nexus',
usernameVariable: 'NEXUS_USR', passwordVariable: 'NEXUS_PSW')]) {
            sh '''#!/bin/bash
                set -e
                WAR_FILE=$(find target -type f -name "*.war" | head -
n1)
                if [[ ! -f "$WAR_FILE" ]]; then
                    echo "✖ No WAR file found in target/"; exit 1
                fi

                FILE_NAME=$(basename "$WAR_FILE")
                VERSION="0.0.${BUILD_NUMBER}"
                GROUP_PATH=$(echo "${NEXUS_GROUP}" | tr '.' '/')
                echo "📤 Uploading $FILE_NAME to Nexus as version
$VERSION..."
                curl -f -u "${NEXUS_USR}">${NEXUS_PSW}" --upload-file
"$WAR_FILE" \
"${NEXUS_URL}/repository/${NEXUS_REPO}/${GROUP_PATH}/${NEXUS_ARTIFACT}/${V
ERSION}/${NEXUS_ARTIFACT}-${VERSION}.war"
                echo "✅ Artifact uploaded successfully to Nexus!"
            ...
        }
    }

/* === Stage 5: Deploy to Tomcat === */
stage('Deploy to Tomcat') {
    agent { label 'tomcat' }
    steps {
        withCredentials([
            usernamePassword(credentialsId: 'nexus', usernameVariable:
'NEXUS_USR', passwordVariable: 'NEXUS_PSW'),
            usernamePassword(credentialsId: 'Tomcat',
usernameVariable: 'TOMCAT_USR', passwordVariable: 'TOMCAT_PSW')
        ]) {
            sh '''#!/bin/bash
                set -e
                cd /tmp || exit 1
                rm -f *.war

                VERSION="0.0.${BUILD_NUMBER}"
                GROUP_PATH=$(echo "${NEXUS_GROUP}" | tr '.' '/')
                WAR_URL="${NEXUS_URL}/repository/${NEXUS_REPO}/${GROUP_PATH}/${NEXUS_ARTIF
ACT}/${VERSION}/${NEXUS_ARTIFACT}-${VERSION}.war"

                echo "⬇️ Downloading WAR from Nexus: $WAR_URL"
            }
        }
    }
}

```

```

curl -u "${NEXUS_USR}:${NEXUS_PSW}" -O "$WAR_URL"

WAR_FILE=$(basename "$WAR_URL")
APP_NAME="${NEXUS_ARTIFACT}"

echo "📝 Undeploying old app (if exists)..."
curl -u "${TOMCAT_USR}:${TOMCAT_PSW}"
"${TOMCAT_URL}/undeploy?path=/${APP_NAME}" || true

echo "🚀 Deploying new WAR to Tomcat..."
curl -u "${TOMCAT_USR}:${TOMCAT_PSW}" --upload-file
"$WAR_FILE" \
"${TOMCAT_URL}/deploy?path=/${APP_NAME}&update=true"

echo "✅ Deployment successful! Application updated."
    ...
}

}

}

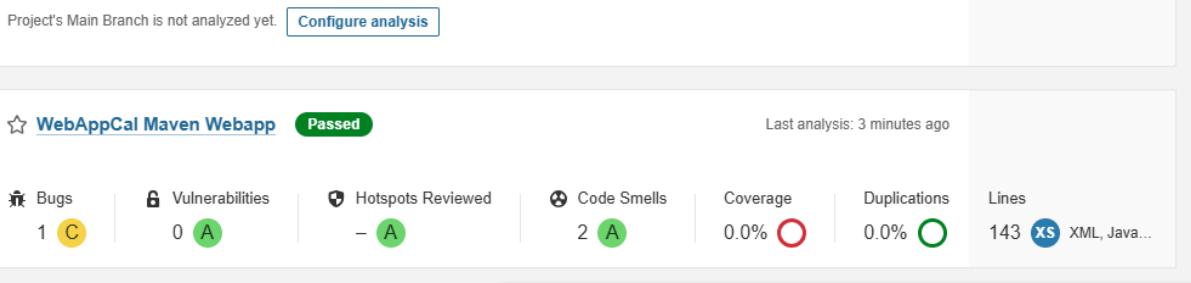
post {
    success {
        echo '🎉 Pipeline completed successfully – Application live on
Tomcat!'
    }
    failure {
        echo '❌ Pipeline failed – Check Jenkins logs.'
    }
}
}
```

Jenkins / 1st job / #1

- [Pipeline Overview](#)
- [Restart from Stage](#)
- [Replay](#)
- [Pipeline Steps](#)
- [Workspaces](#)

```
[Pipeline] script
[Pipeline] [
[Pipeline] echo
Hello from Sonar Node - Stage 1
[Pipeline] sh
+ whoami
ubuntu
[Pipeline] sh
+ hostname -i
172.31.48.5 fe80::8f2:55ff:fe89:617b
[Pipeline] echo
Cloning the code...
[Pipeline] echo
Building the code...
[Pipeline] }
[Pipeline] //
[Pipeline] [
[Pipeline] //
[Pipeline] stage
[Pipeline] {
  (nexus Node - Stage 2)
[Pipeline] node
Running on nexus in /home/ubuntu/jenkins/workspace/1st job
[Pipeline] [
[Pipeline] script
[Pipeline] [
[Pipeline] echo
Hello from Sonar Node - Stage 2
[Pipeline] sh
+ whoami
ubuntu
[Pipeline] sh
+ hostname -i
172.31.32.208 fe80::88e:5eff:fe69:1d45
[Pipeline] echo
Cloning the code...
[Pipeline] echo
Building the code...
[Pipeline] }
[Pipeline] //
[Pipeline] [
[Pipeline] //
[Pipeline] stage
[Pipeline] {
  (tomcat Node - Stage 3)
[Pipeline] node
Running on tomcat in /home/ubuntu/jenkins/workspace/1st job
[Pipeline] [
[Pipeline] script
[Pipeline] [
  restarttomcat.sh
]
```

☆ sai



## sonatype nexus repository

Community Edition

Dashboard

Search

**Browse**

Upload

Settings

**Browse** / **maven-releases**

HTML View

com

  web

    cal

      webapp-add

        0.0.6

          webapp-add-0.0.6.war

</> Changes

Console Output

Edit Build Information

Delete build '#6'

Timings

Git Build Data

Pipeline Overview

Restart from Stage

Replay

Pipeline Steps

Workspaces

← Previous Build

Started by user sai

[Pipeline] Start of Pipeline

[Pipeline] node

Running on sonar in /home/ubuntu/jenkins/workspace/endtoend

[Pipeline] {

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Tool Install)

[Pipeline] tool

[Pipeline] envVarsForTool

[Pipeline] tool

[Pipeline] envVarsForTool

[Pipeline] }

[Pipeline] // stage

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Checkout code)

[Pipeline] tool

[Pipeline] envVarsForTool

[Pipeline] tool

[Pipeline] envVarsForTool

[Pipeline] withEnv

[Pipeline] {

[Pipeline] echo

Cloning source from GitHub...

[Pipeline] checkout

The recommended git tool is: NONE

No pending builds

```
[@1;34mINFO@m] 14:45:18.150 Sensor Java CPD Block Indexer (done) | time=13ms
[@1;34mINFO@m] 14:45:18.154 SCM Publisher SCM provider for this project is: git
[@1;34mINFO@m] 14:45:18.155 SCM Publisher 5 source files to be analyzed
[@1;34mINFO@m] 14:45:18.242 SCM Publisher 5/5 source files have been analyzed (done) | time=86ms
[@1;34mINFO@m] 14:45:18.245 CPD Executor Calculating CPD for 2 files
[@1;34mINFO@m] 14:45:18.251 CPD Executor CPD calculation finished (done) | time=6ms
[@1;34mINFO@m] 14:45:18.329 Analysis report generated in 73ms, dir size=136.7 kB
[@1;34mINFO@m] 14:45:18.345 Analysis report compressed in 16ms, zip size=25.2 kB
[@1;34mINFO@m] 14:45:18.929 Analysis report uploaded in 583ms
[@1;34mINFO@m] 14:45:18.932 ANALYSIS SUCCESSFUL, you can find the results at: http://13.62.98.135:9000/dashboard?
[@1;34mINFO@m] 14:45:18.932 Note that you will be able to access the updated dashboard once the server has proces
[@1;34mINFO@m] 14:45:18.932 More about the report processing at http://13.62.98.135:9000/api/ce/task?id=AZpPVFezI
[@1;34mINFO@m] 14:45:18.956 Analysis total time: 8.853 s
[@1;34mINFO@m] @1m-----
[@1;34mINFO@m] @1;32mBUILD SUCCESS@m
[@1;34mINFO@m] @1m-----
[@1;34mINFO@m] Total time: 12.901 s
[@1;34mINFO@m] Finished at: 2025-11-04T14:45:18Z
[@1;34mINFO@m] @1m-----
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build artifact)
```

```
[@1;34mINFO@m]
[@1;34mINFO@m] @1m-- @0;32mmaven-war-plugin:3.3.2:war@m @1m(default-war)@m @36mwebapp-add@0;1m --@m
[@1;34mINFO@m] Packaging webapp
[@1;34mINFO@m] Assembling webapp [webapp-add] in [/home/ubuntu/jenkins/workspace/endtoend/target/webapp-add-0.0.2]
[@1;34mINFO@m] Processing war project
[@1;34mINFO@m] Copying webapp resources [/home/ubuntu/jenkins/workspace/endtoend/src/main/webapp]
[@1;34mINFO@m] Building war: /home/ubuntu/jenkins/workspace/endtoend/target/webapp-add-0.0.2.war
[@1;34mINFO@m] @1m-----
[@1;34mINFO@m] @1;32mBUILD SUCCESS@m
[@1;34mINFO@m] @1m-----
[@1;34mINFO@m] Total time: 4.095 s
[@1;34mINFO@m] Finished at: 2025-11-04T14:45:26Z
[@1;34mINFO@m] @1m-----
[Pipeline] sh
+ echo ✓ Build Completed!
✓ Build Completed!
[Pipeline] sh
+ ls -lh target/webapp-add-0.0.2.war
-rw-rw-r-- 1 ubuntu ubuntu 3.9K Nov 4 14:45 target/webapp-add-0.0.2.war
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
```

```
0      0      0      0      0      0      0      0 ---:---:---:---:---:---:---:--- 0
100  3984    0      0  100  3984      0  24124 ---:---:---:---:---:---:---:--- 24145
[✓] Artifact uploaded successfully to Nexus!
[Pipeline]
[Pipeline] // withCredentials
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy to Tomcat)
[Pipeline] node
Running on tomcat in /home/ubuntu/jenkins/workspace/endtoend
[Pipeline] {
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] tool
```

```
[✓] Deployment successful! Application updated.
[Pipeline]
[Pipeline] // withCredentials
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // node
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
⚠ Pipeline completed successfully - Application live or
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

---

