

## **Project-1**

**Install the jenkins in the linux server by using the commands:**

```
sudo wget -O /etc/yum.repos.d/jenkins.repo \
    https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
sudo yum upgrade
# Add required dependencies for the jenkins package
sudo yum install fontconfig java-17-openjdk
sudo yum install jenkins
sudo systemctl daemon-reload
```

Enable the jenkins so that we run jenkins directly by running the server.

```
sudo systemctl enable jenkins
```

```
sudo systemctl start jenkins
```

To check the status of the jenkins use the command.

```
sudo systemctl start jenkins
```

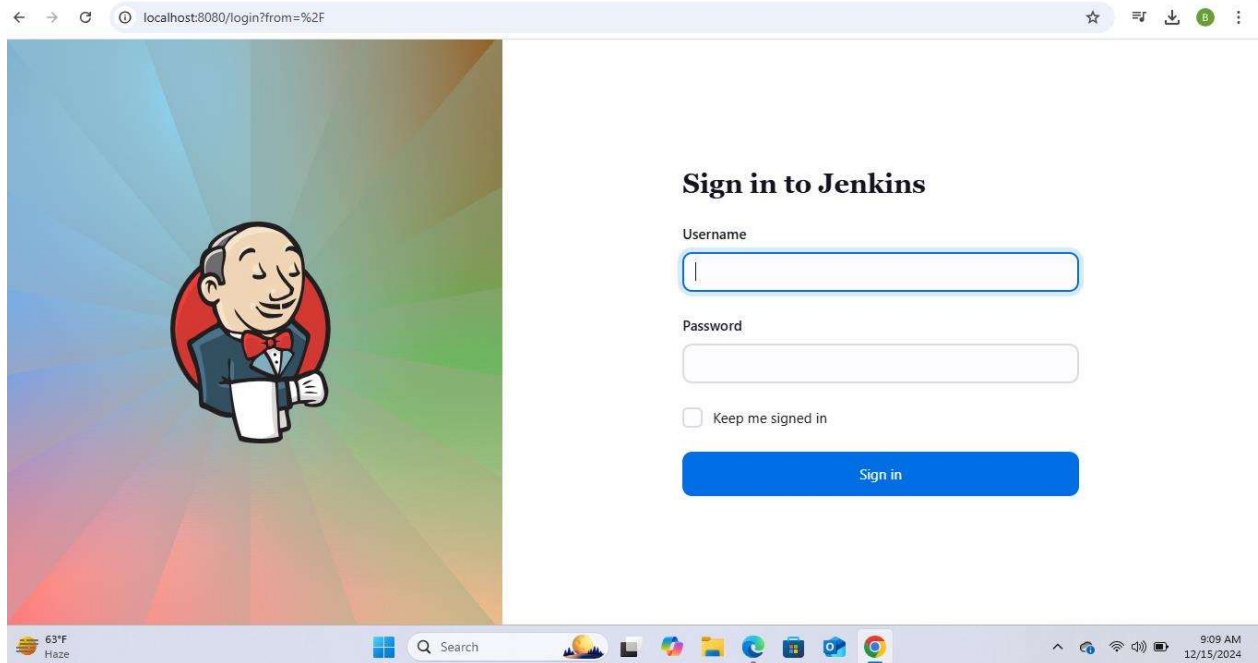
**To install jenkins in the local machine use the following link**

<https://www.jenkins.io/download/thank-you-downloading-windows-installer/>

It will download .msi file and install it, while installing you need to have java 17 or java 21 in your system.

After installing the jenkins open it using the 8080 port number. If you have installed in the local system use the <http://localhost:8080> or if you have installed in the remote then use <http://<publicIpOfServer>:8080> .Make sure that your 8080 port is opened in the Inbound rules.

**After opening the interface will look like as follows.**



**Create a Another server which can be used as a node for the jenkins.**

Install the java 17 and maven in the node server.

Use the command yum install maven for installing maven. It will install the maven.

Go to tomcat website and copy the link address of tomcat 9 tar.gz file.

Download it in server using wget link\_address it will download the tar.gz file.

Use the command tar -zxvf tomcat-----tar file to extract the files.

Go to bin folder and start the tomcat server.

```

aws
Search [Alt+S]
Mumbai Bharath

A newer version of Amazon Linux is available!
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/

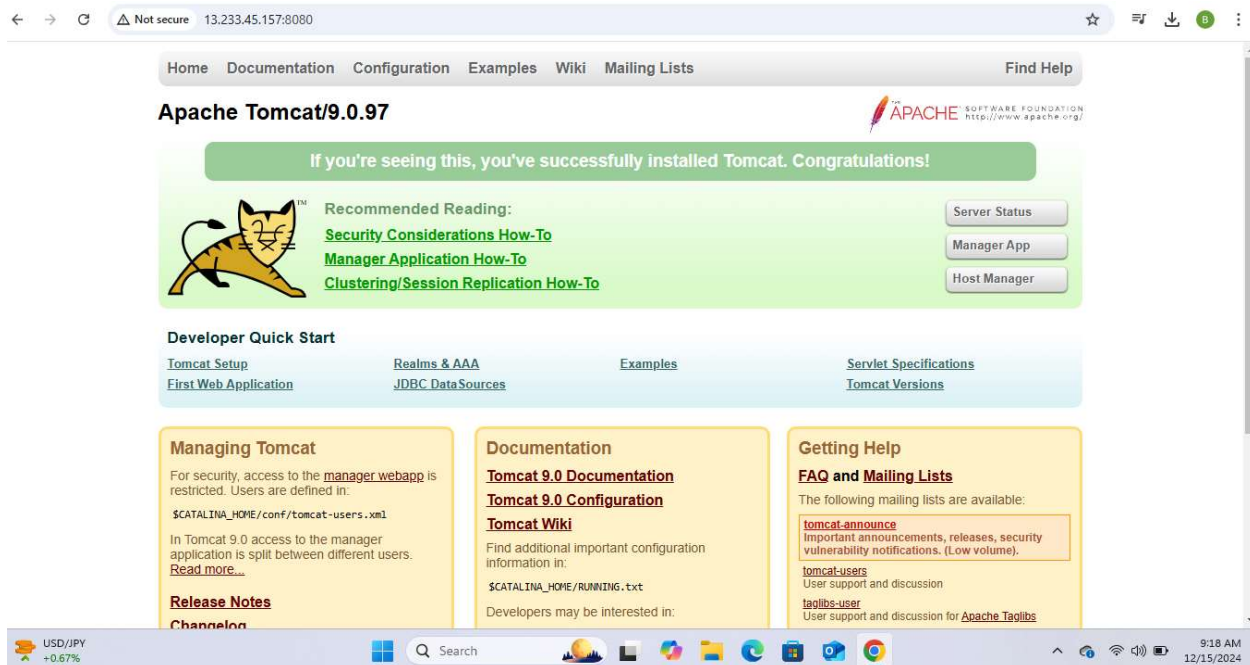
[root@ip-172-31-43-48 ~]# ls
apache-tomcat-9.0.97  apache-tomcat-9.0.97.tar.gz  nexus-3.75.1-01-unix.tar.gz  sonarqube-9.9.0.100196.zip  sonatype-work
[root@ip-172-31-43-48 ~]# cd apache-tomcat-9.0.97/
[root@ip-172-31-43-48 apache-tomcat-9.0.97]# cd bin
[root@ip-172-31-43-48 bin]# ls
bootstrap.jar  catalina.bat  catalina.sh  catalina-tasks.xml  commons-daemon.jar  commons-daemon-native.tar.gz  configtest.bat  configtest.sh  daemon.sh  digest.bat  digest.sh  makebase.bat  makebase.sh  setclasspath.bat  setclasspath.sh  shutdown.bat  shutdown.sh  startup.bat  startup.sh  tomcat-juli.jar  tomcat-native.tar.gz  tool-wrapper.sh  version.bat  version.sh
[root@ip-172-31-43-48 bin]# ./startup.sh start
Using CATALINA_BASE:   /root/apache-tomcat-9.0.97
Using CATALINA_HOME:   /root/apache-tomcat-9.0.97
Using CATALINA_TMPDIR: /root/apache-tomcat-9.0.97/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /root/apache-tomcat-9.0.97/bin/bootstrap.jar:/root/apache-tomcat-9.0.97/bin/tomcat-juli.jar
Tomcat started.
[root@ip-172-31-43-48 bin]#

i-008d6b98afd4e8f8a (node1)
PublicIPs: 13.233.45.157 PrivateIPs: 172.31.43.48

CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
64°F Haze 9:16 AM 12/15/2024

```

Open it using the 8080 port in the browser using public ip address.



## Download and install sonarqube in the server.

Copy the download link from the sonarqube website and download it using wget command.

Extract the files by using unzip command.

Move the folder from root directory to /opt/.

Create a user by using useradd command.

Change the owner and group of sonar from root to user by using chown command.

Give the permissions by using the chmod command.

After successfully starting the server open it using the 9000 port.

<http://ipaddress:9000>.

Note: Please make sure that port 9000 is opened in the inbound rules

```
aws [Alt+S]

bootstrap.jar  ciphers.bat  configtest.bat  digest.sh  setclasspath.sh  startup.sh  tool-wrapper.sh
catalina.bat  ciphers.sh  configtest.sh  makebase.bat  shutdown.bat  tomcat-juli.jar  version.bat
catalina.sh  commons-daemon.jar  daemon.sh  makebase.sh  shutdown.sh  tomcat-native.tar.gz  version.sh
catalina-tasks.xml  commons-daemon-native.tar.gz  digest.bat  setclasspath.bat  startup.bat  tool-wrapper.bat

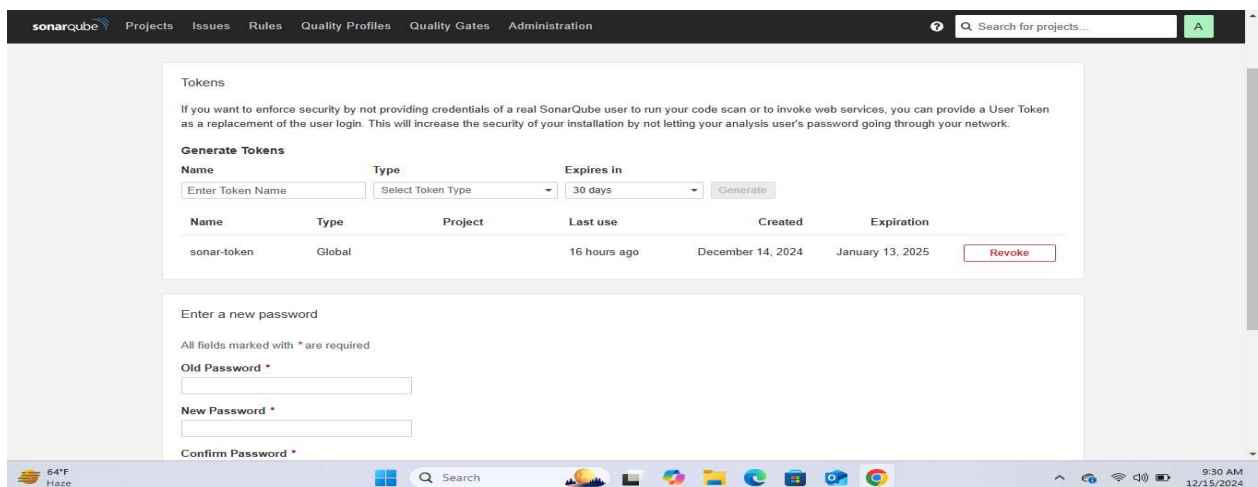
root@ip-172-31-43-48 bin]# ./startup.sh start
Using CATALINA_BASE:   /root/apache-tomcat-9.0.97
Using CATALINA_HOME:   /root/apache-tomcat-9.0.97
Using CATALINA_TMPDIR: /root/apache-tomcat-9.0.97/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /root/apache-tomcat-9.0.97/bin/bootstrap.jar:/root/apache-tomcat-9.0.97/bin/tomcat-juli.jar
tomcat started.
root@ip-172-31-43-48 bin]# su bharath
[bharath@ip-172-31-43-48 bin]$ cd /opt
[bharath@ip-172-31-43-48 opt]$ cd sonarqube-9.9.8.100196/
[bharath@ip-172-31-43-48 sonarqube-9.9.8.100196]$ ls
bin conf COPYING data dependency-license.json elasticsearch extensions lib logs temp web
[bharath@ip-172-31-43-48 sonarqube-9.9.8.100196]$ cd bin
[bharath@ip-172-31-43-48 bin]$ ls
linux-x86-64 macosx-universal-64 windows-x86-64 winsw-license
[bharath@ip-172-31-43-48 bin]$ cd linux-x86-64/
[bharath@ip-172-31-43-48 linux-x86-64]$ ls
SonarQube.pid sonar.sh
[bharath@ip-172-31-43-48 linux-x86-64]$ ./sonar.sh start
```

i-008d6b98afd4e8f8a (node1)  
PublicIPs: 13.233.45.157 PrivateIPs: 172.31.43.48

After starting the server provide username and password as admin and admin and login.  
Then change your password.



Go to administrator after login and click on security and create a token.



## Download nexus and install in the server.

Copy the download link from the nexus website and download it using wget command.

Extract the files by using unzip command.

Move the folder from root directory to /opt/.

After successfully starting the server open it using the 8081 port.

http://ipaddress:8081.

Note: Please make sure that port 8081 is opened in the inbound rules.

```
aws [Alt+S] Mumbai Bharath

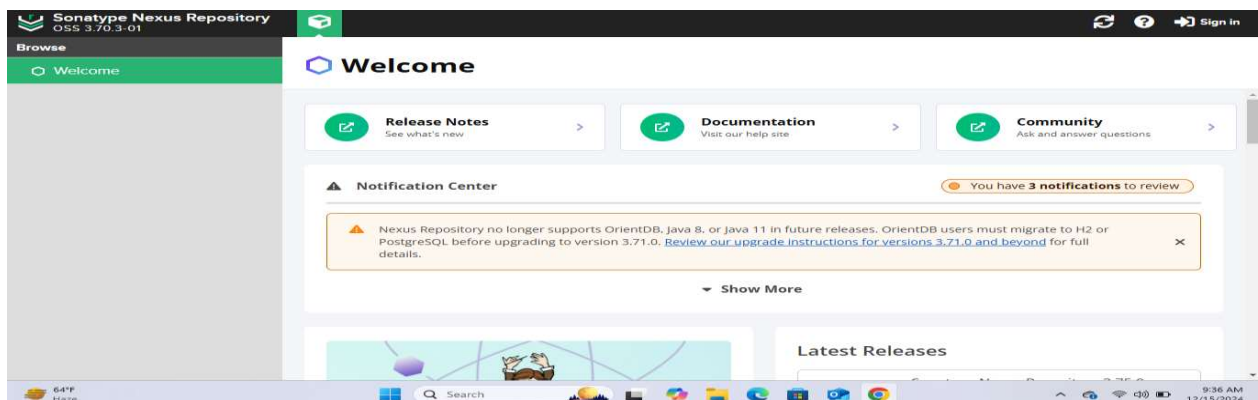
[ec2-user@ip-172-31-46-21 ~]$ sudo -i
[root@ip-172-31-46-21 ~]# cd /opt/nexus-3.70.3-01/
[root@ip-172-31-46-21 nexus-3.70.3-01]# cd bin/
[root@ip-172-31-46-21 bin]# ls
contrib  nexus  nexus.rc  nexus.vmoptions
[root@ip-172-31-46-21 bin]# ./nexus start
WARNING: *****
WARNING: Detected execution as "root" user. This is NOT recommended!
WARNING: *****
Starting nexus
[root@ip-172-31-46-21 bin]#
```

i-0eb18d015b787e259 (nexus)

PublicPis: 13.232.107.23 PrivatePis: 172.31.46.21

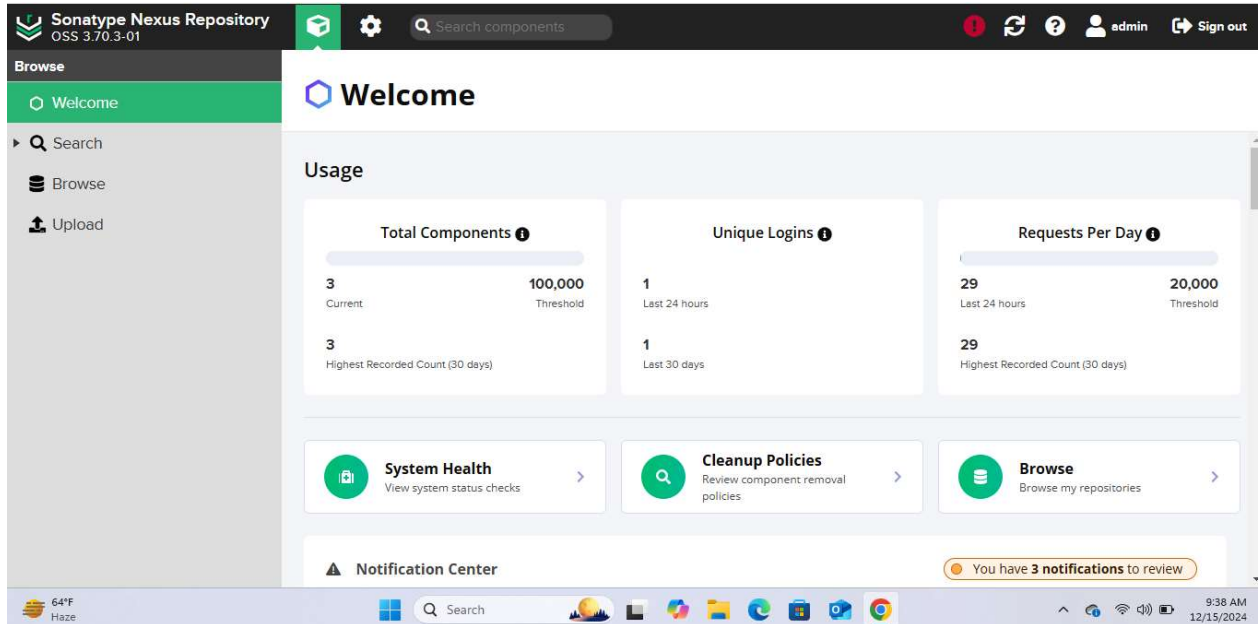


Open it in the browser with the port 8081 by using the public ip address.



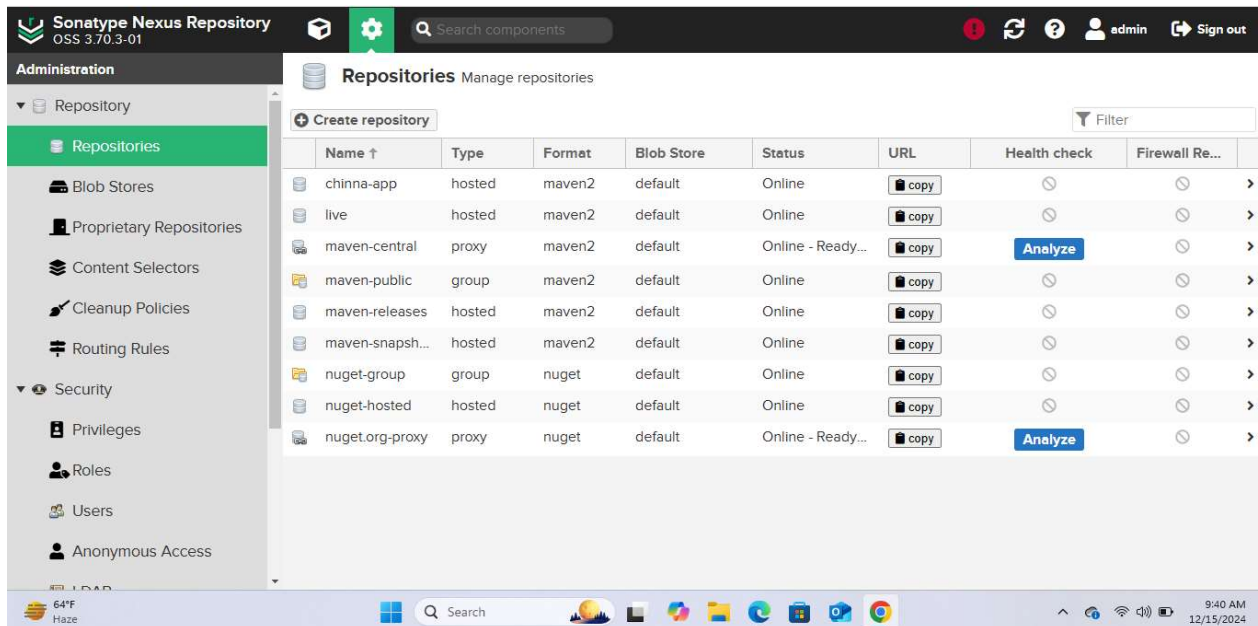
After clicking on the sign in provide username and password as admin and admin and login.

Then change your password.



The screenshot shows the Sonatype Nexus Repository OSS 3.70.3-01 Welcome screen. The interface includes a top navigation bar with the Sonatype logo, version information, a search bar, and user controls (admin, Sign out). A left sidebar contains navigation links: Browse, Welcome, Search, Browse, and Upload. The main content area is titled 'Welcome' and features a 'Usage' section with three cards: 'Total Components' (Current: 3, Threshold: 100,000, Highest Recorded Count: 3), 'Unique Logins' (Last 24 hours: 1, Last 30 days: 1), and 'Requests Per Day' (Last 24 hours: 29, Threshold: 20,000, Highest Recorded Count: 29). Below these are three action cards: 'System Health', 'Cleanup Policies', and 'Browse'. A 'Notification Center' at the bottom indicates 3 notifications to review. The Windows taskbar at the bottom shows the date and time as 9:38 AM on 12/15/2024.

Create one repository for your project.



The screenshot shows the Sonatype Nexus Repository OSS 3.70.3-01 Administration - Repositories screen. The left sidebar is expanded to 'Administration' and then 'Repositories'. The main content area is titled 'Repositories' and includes a 'Create repository' button and a table of existing repositories. The table has columns for Name, Type, Format, Blob Store, Status, URL, Health check, and Firewall Re... (Firewall Rules). The table lists several repositories, including 'chinna-app', 'live', 'maven-central', 'maven-public', 'maven-releases', 'maven-snapsh...', 'nuget-group', 'nuget-hosted', and 'nuget.org-proxy'. The 'Health check' column shows status icons and 'Analyze' buttons for some repositories. The Windows taskbar at the bottom shows the date and time as 9:40 AM on 12/15/2024.

Name ↑	Type	Format	Blob Store	Status	URL	Health check	Firewall Re...
chinna-app	hosted	maven2	default	Online	<a href="#">copy</a>		
live	hosted	maven2	default	Online	<a href="#">copy</a>		
maven-central	proxy	maven2	default	Online - Ready...	<a href="#">copy</a>	<a href="#">Analyze</a>	
maven-public	group	maven2	default	Online	<a href="#">copy</a>		
maven-releases	hosted	maven2	default	Online	<a href="#">copy</a>		
maven-snapsh...	hosted	maven2	default	Online	<a href="#">copy</a>		
nuget-group	group	nuget	default	Online	<a href="#">copy</a>		
nuget-hosted	hosted	nuget	default	Online	<a href="#">copy</a>		
nuget.org-proxy	proxy	nuget	default	Online - Ready...	<a href="#">copy</a>	<a href="#">Analyze</a>	



## Creating a node in the Jenkins

Sign in to Jenkins using username and password.

In dashboard click on manage Jenkins and then nodes and click on new node.

Give node name and click on permanent agent and then create.

Enter Remote root directory as Jenkins and enter label as dev.

Note: remember the label name we can assign job to node using label.

Select launch method as login via SSH.

Give the Host as public address of the node server.

Add the credentials as username ec2-user and secret key as the text inside the .pem file.

Set the verifying strategy as non verifying strategy and save the node.

The screenshot shows the Jenkins 'New Node' configuration page. The form is partially filled with the following values:

- Name: node2
- Description: (empty)
- Number of executors: 1
- Remote root directory: jenkins
- Labels: dev
- Usage: Use this node as much as possible
- Launch method: Launch agents via SSH
- Host: 13.233.45.157

A warning message is displayed below the 'Remote root directory' field:

⚠ Are you sure you want to use a relative path for the FS root? Note that relative paths require that you can assure that the selected launcher provides a consistent current working directory. Using an absolute path is highly recommended.

The 'Save' button is visible at the bottom of the form.

Dashboard > Manage Jenkins > Nodes >

Launch agents via JCL

Host ?

13.233.45.157

Credentials ?

ec2-user (node1)

+ Add

Host Key Verification Strategy ?

Non verifying Verification Strategy

Advanced

Availability ?

Keep this agent online as much as possible

Save

64°F Haze 9:50 AM 12/15/2024

localhost:8080/manage/computer/createItem

Dashboard > Manage Jenkins > Nodes >

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

REST API Jenkins 2.479.2

64°F Haze 9:50 AM 12/15/2024

Now the agent has been created successfully see the console of the node the agent must be online.

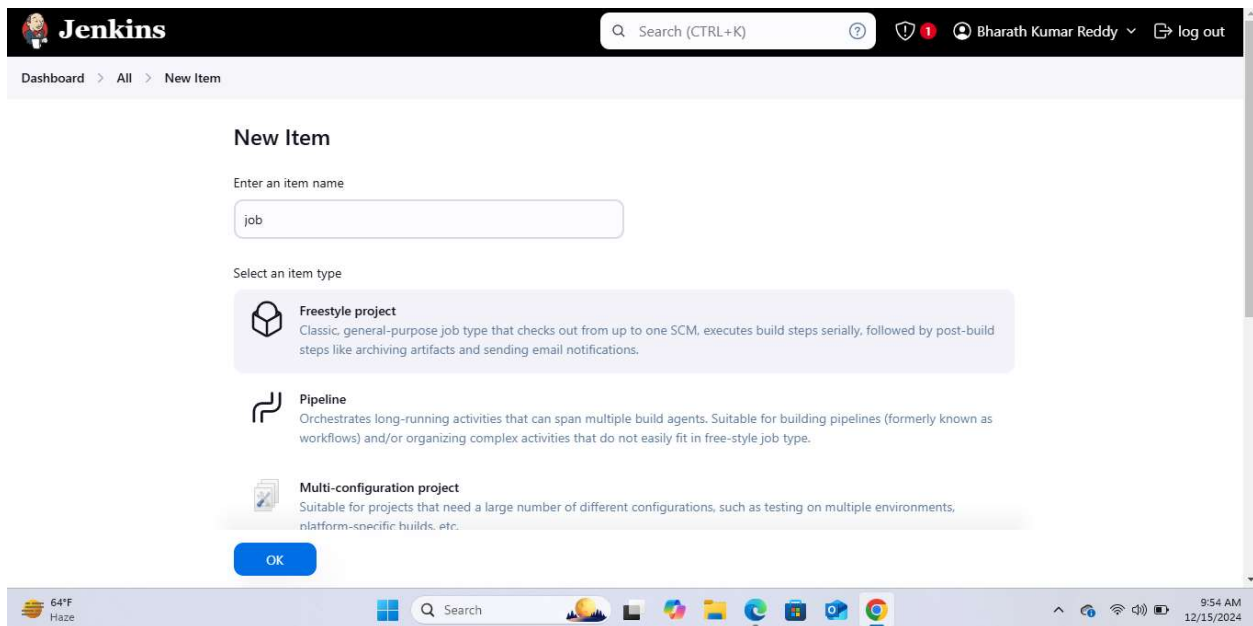
Note : install the maven and tomcat 9 from the tools.

Creation of job in the jenkins

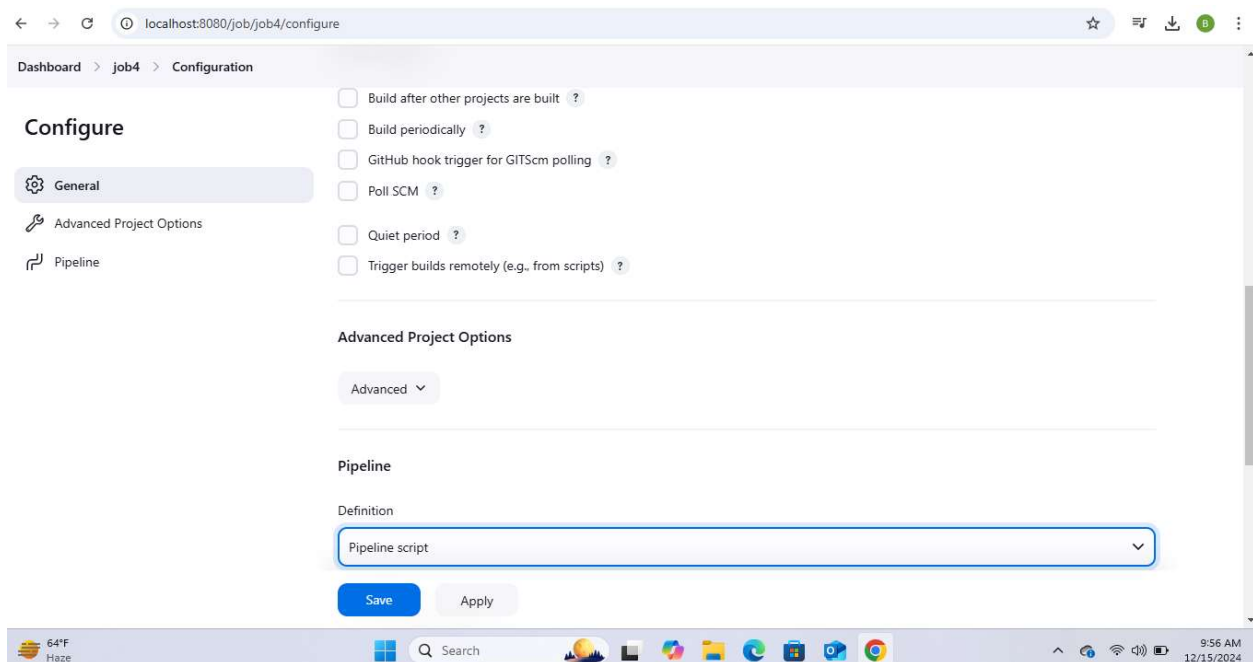


Go to the Dashboard of the Jenkins.

Click on the new item and enter the job name and select pipeline.



Select as pipeline script.



Click on the pipeline syntax and generate the code according to our requirements.

For fetching the code from git.

The screenshot shows the Jenkins Pipeline Syntax Snippet Generator interface. The top navigation bar includes the Jenkins logo, a search bar, and user information (Bharath Kumar Reddy). The breadcrumb trail is Dashboard > job4 > Pipeline Syntax. On the left, a sidebar lists various generators: Snippet Generator (selected), Declarative Directive Generator, Declarative Online Documentation, Steps Reference, Global Variables Reference, Online Documentation, Examples Reference, and IntelliJ IDEA GDSDL. The main content area is titled 'Overview' and explains the purpose of the Snippet Generator. Below this, the 'Steps' section shows a 'Sample Step' dropdown set to 'git: Git'. The configuration fields include 'Repository URL' (https://github.com/Bharath-Prajwal/live01.git), 'Branch' (main), and 'Credentials' (Bharath-Prajwal/\*\*\*\*\* (git)). There are checkboxes for 'Include in polling?' and 'Include in changelog?'. A 'Generate Pipeline Script' button is present. The generated script is displayed in a text area: `git branch: 'main', credentialsId: 'git', url: 'https://github.com/Bharath-Prajwal/live01.git'`. The bottom of the image shows a Windows taskbar with the system clock at 9:59 AM on 12/15/2024.

Note: After setting the requirements if we click on generate pipeline script it will automatically generate script.

Paste that generated scripted in the stage.

Install the plugins deploy to container, sonar qube scanner, nexus artifact uploader from the plugins.

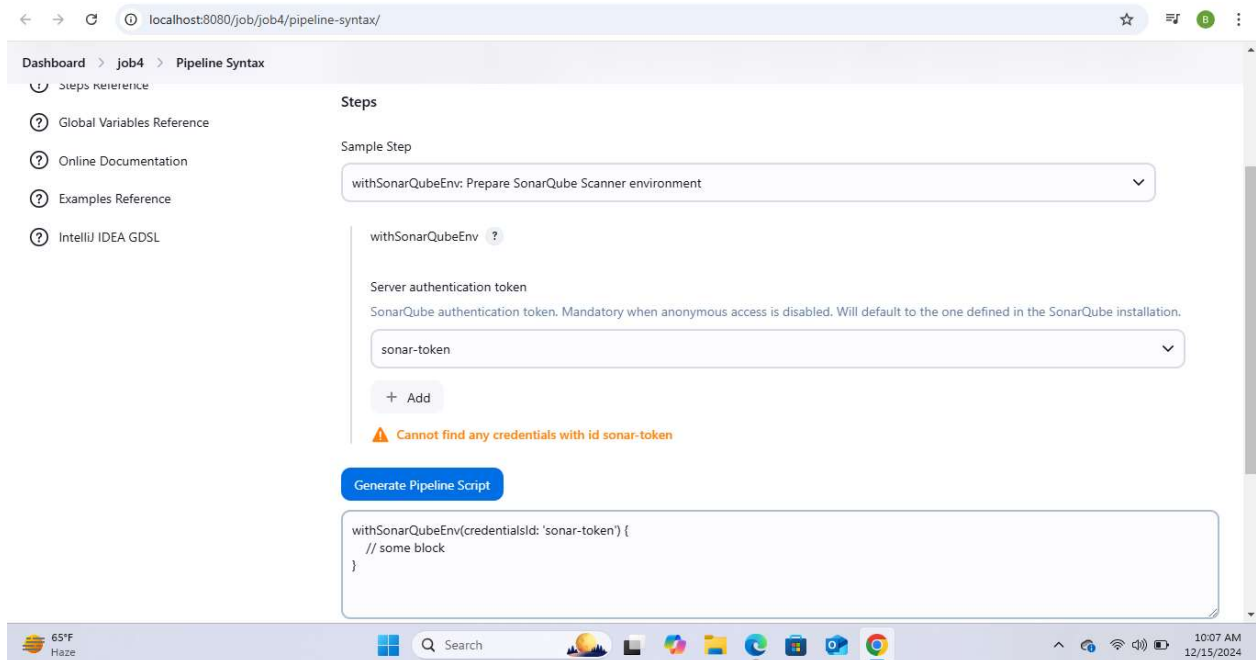
Add the tomcat and sonarqube from the system configuration and tools.

After fetching the code build it using the maven.

We use sh 'mvn clean package' command.

For testing the code in the sonarqube follow the pictures below.

For server authentication we use the key that is generated in the SonarQube.



In the block write the code as

Sh 'mvn sonar:sonar'

For uploading artifact in the nexus follow the steps.

In the nexus url paste the url of the nexus server.

Create credentials by using the username and password of the nexus.

For groupId and version refer the pom.xml file in your github repository.

For Repository enter the name of the repository that we are created in the nexus.

Click on addartifact.

Refer pom.xml for id.

Type should be war and file should be target/filename.war.

localhost:8080/job/job4/pipeline-syntax/

Dashboard > Job4 > Pipeline Syntax

Global Variables Reference

Online Documentation

Examples Reference

IntelliJ IDEA GDSDL

Sample Step

nexusArtifactUploader: Nexus Artifact Uploader

nexusArtifactUploader

**Nexus Details**

Nexus Version

NEXUS3

Protocol

HTTP

Nexus URL ?

13.232.107.23:8081

Credentials

admin/\*\*\*\*\* (nexus2)

GroupId

injavahome

Version

0.1

Repository ?

live

Artifacts

**Artifact**

ArtifactId

live

Type ?

war

Classifier ?

File ?

target/live.war

Add

Generate Pipeline Script

nexusArtifactUploader artifacts: [[artifactId: 'live', classifier: '', file: 'target/live.war', type: 'war']], credentialsId: 'nexus2', groupId: 'injavahome', nexusUrl: '13.232.107.23:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'live', version: '0.1']

For Deploying into tomcat server:

Context path is the name with the file that we can deploy into the tomcat.

Click on the add container and select tomcat 9.x

Give the tomcat username and password as credentials.

Give the host url as the tomcat url.

localhost:8080/job/job4/pipeline-syntax/

Dashboard > job4 > Pipeline Syntax

deploy: Deploy war/ear to a container

deploy

WAR/EAR files ?

\*/\*.war

Context path ?

live

Containers

**Tomcat 9.x Remote**

Credentials

tomcat/\*\*\*\*\* (tomcat)

+ Add

Tomcat URL ?

http://13.233.45.157:8080/

Advanced

Add Container

☒ Deploy on failure

Generate Pipeline Script

deploy adapters: [tomcat9(credentialsId: 'tomcat', path: '', url: 'http://13.233.45.157:8080/'), contextPath: 'live', war: '\*/\*.war']

The final script should be like:

Script ?

```
1 node ('dev') {
2   def mvnHome
3
4   stage('Preparation') { // for display purposes
5     // Get some code from a GitHub repository
6     git branch: 'main', url: 'https://github.com/Bharath-Prajwal/live01.git'
7     // Get the Maven tool.
8     // ** NOTE: This 'M3' Maven tool must be configured
9     // **      in the global configuration.
10    mvnHome = tool 'maven'
11  }
12  stage('Build') {
13    // Run the maven build
14    sh 'mvn clean package'
15  }
16  stage('sonar'){
17    withSonarQubeEnv('sonarqube', credentialsId: 'sonar-token'){
18      sh 'mvn sonar:sonar'
19    }
20  }
21
22  stage('nexus'){
23    nexusArtifactUploader artifacts: [
24      [
25        artifactId: 'live',
26        classifier: '',
27        file: 'target/live.war',
28        file: 'target/live.war',
29        type: 'war'
30      ],
31      credentialsId: 'nexus2',
32      groupId: 'in.javahome',
33      nexusUrl: '35.154.77.114:8081',
34      nexusVersion: 'nexus3',
35      protocol: 'http',
36      repository: 'live',
37      version: '0.1'
38    ]
39  }
40  stage('Deploy') {
41    deploy adapters: [tomcat9(credentialsId: 'tomcat', path: '', url: 'http://13.126.184.89:8080/']], contextPath:
42  }
43 }
```

After that click on build now.



The bulid was success and it is deployed into tomcat successfully.

## Create a job using the Freestyle.

Restict it to run in the node using the label name of the node.

Select git and give the repository url for fetching the code and credentials.

In the build steps select invoke top level maven targets.

Give the goals as clean package.

In add build steps select sonarqube scanner and give analysis properties as

sonar.projectKey=projectname

sonar.projectVersion=version

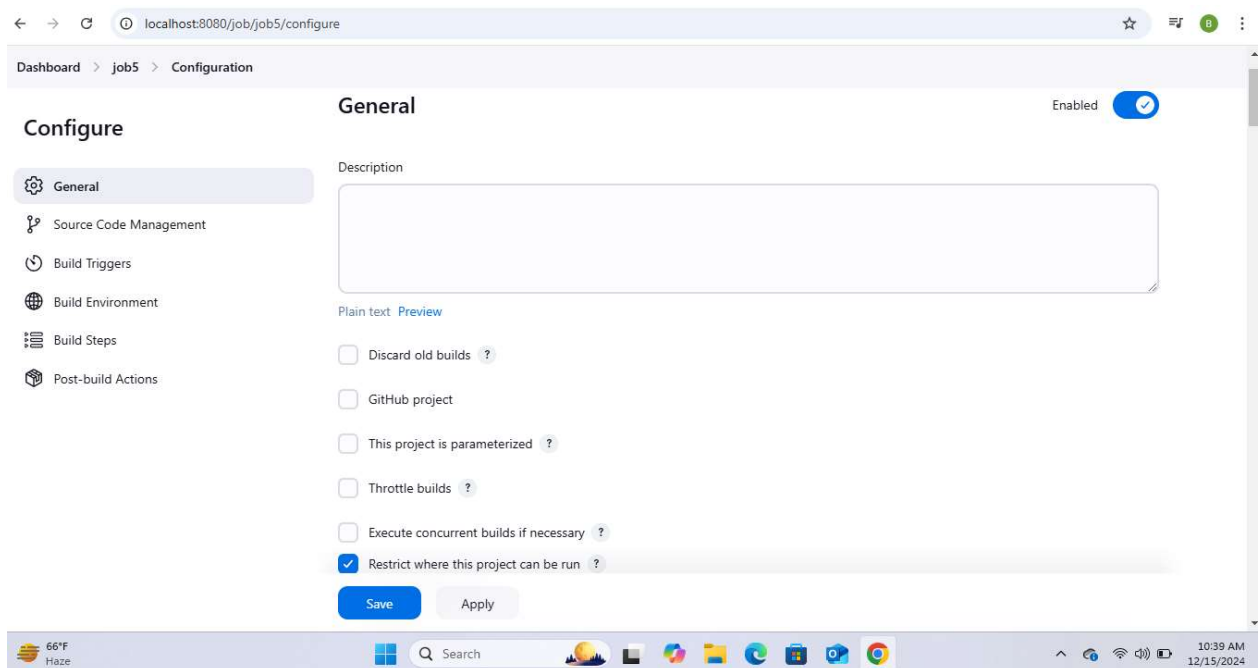
sonar.sources=src/main/java

sonar.java.binaries=target/classes

Again click on add build step and select nexus artifact uploader and update the data as we have done in pipeline project a same.

In the post build actions select deploy to war/ear to a container.

Give the context path and tomcat url and credentials.





Dashboard > job5 > Configuration

### Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

☒ Restrict where this project can be run ?

Label Expression ?

dev

Label dev matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced ▾

#### Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/Bharath-Prajwal/live01.git

Credentials ?

Save Apply

66°F Haze 10:39 AM 12/15/2024

Dashboard > job5 > Configuration

### Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Credentials ?

Bharath-Prajwal/\*\*\*\*\* (git)

+ Add

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

\*/main

Add Branch

Repository browser ?

Save Apply

66°F Haze 10:40 AM 12/15/2024

← → ↺ localhost:8080/job/job5/configure

Dashboard > job5 > Configuration

### Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Repository browser ?

(Auto)

Additional Behaviours

Add ▾

#### Build Triggers

☐ Trigger builds remotely (e.g., from scripts) ?

☐ Build after other projects are built ?

☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

#### Build Environment

Save Apply

66°F Haze 10:40 AM 12/15/2024

Dashboard > JobS > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Build Steps

Invoke top-level Maven targets ?

Maven Version

maven

Goals

clean package

Advanced

Nexus artifact uploader

Nexus Details

Nexus Version

NEXUS3

Save

Apply

66°F Haze

Search

10:41 AM 12/15/2024

Dashboard > JobS > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Nexus Version

NEXUS3

Protocol

HTTP

Nexus URL ?

35.154.77.114:8081

Credentials

admin/\*\*\*\*\* (nexus2)

+ Add

Groupid

in.javahome

Save

Apply

66°F Haze

Search

10:41 AM 12/15/2024

Dashboard > JobS > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Version

0.1

Repository ?

live

Artifacts

Artifact

ArtifactId

hiring

Type ?

war

Classifier ?

Save

Apply

66°F Haze

Search

10:42 AM 12/15/2024

localhost:8080/job/job5/configure

Dashboard > job5 > Configuration

### Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Classifier ?

File ?

Add

**Execute SonarQube Scanner**

JDK ?

JDK to be used for this SonarQube analysis

(Inherit From Job)

Save

Apply

Dashboard > job5 > Configuration

### Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

(Inherit From Job)

Path to project properties ?

Analysis properties ?

```
sonar.projectKey=live
sonar.projectVersion=0.1
sonar.sources=src/main/java
sonar.java.binaries=target/classes
```

Additional arguments ?

JVM Options ?

Save

Apply

Dashboard > job5 > Configuration

### Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

#### Post-build Actions

**Deploy war/ear to a container**

WAR/EAR files ?

Context path ?

Containers

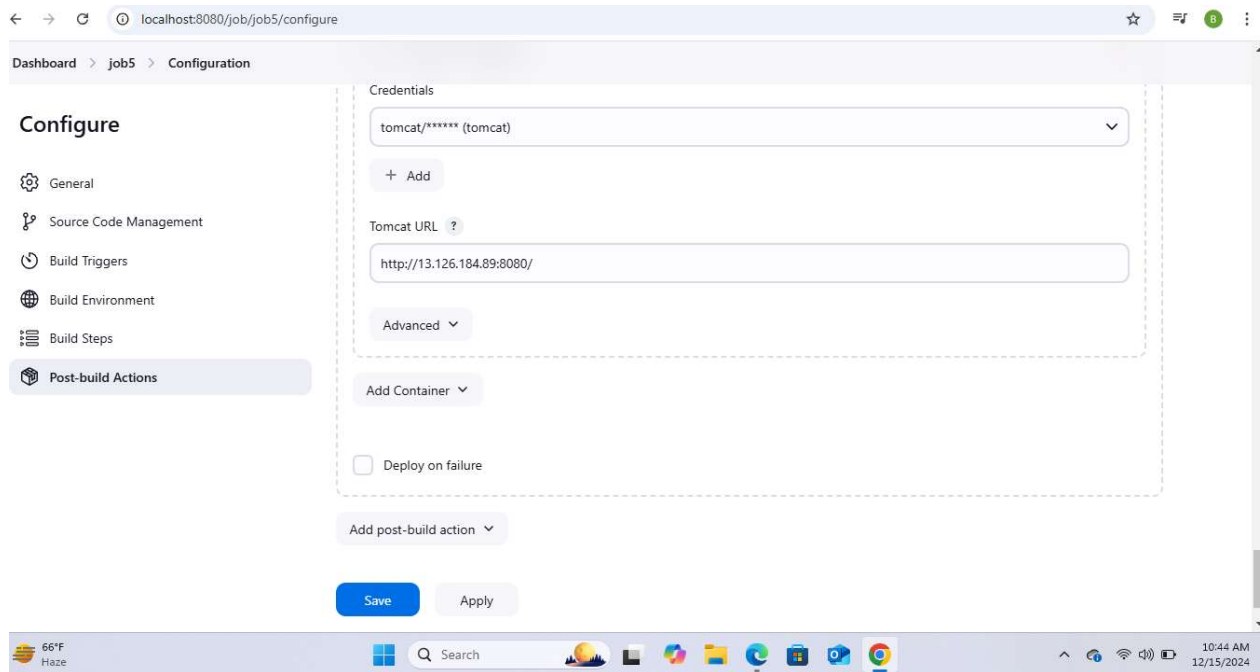
**Tomcat 9.x Remote**

Credentials

tomcat/\*\*\*\*\* (tomcat)

Save

Apply



Click on save and then click on build now