Jenkins Pipeline for Monolithic Application

Overview:

Final Project for Monolythic Applicaiton:

==

1. Create total 4 Ec2 instances of 5.10 of amazon linux with t2.medium

On 1st server: (jenkins)

==============

1. Install jekins (with java 17 and also java 11)
2. Install Git
3. Install maven
4. configure jenkins

On 2nd Server: (sonarqube)

==============

1. install sonarqube with java 11
2. configure sonarqube

On 3rd server: (nexus)

======================

1. install nexus with java 11
2. configure nexus

On 4th Sever: (Tomcat)

=======================

1. Install tomcat with java 11
2. configure tomcat

On Github modify pom.xml for your sonar and nexus links

pipeline creation:

==================

```
 pipeline {
    agent any

    stages {
       stage('git checkout') {
          steps {
            git 'https://github.com/lakshmiprasad2019/myweb.git'
          }
       }
       stage('Sonar testing') {
          steps {
            sh 'mvn sonar:sonar'
          }
       }
       stage('maven Build') {
          steps {
            sh 'mvn clean package'
          }
       }
       stage('Nexus upload') {
          steps {
```

```
          nexusArtifactUploader artifacts: [[artifactId: 'myweb', classifier: '', file: 'target/myweb-8.2.17-SNAPSHOT.war', type: 'war']], credentialsId: 'nexus3', groupId: 'in.javahome', nexusUrl: '172.31.91.86:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'maven-snapshots', version: '8.2.17-SNAPSHOT'

        }

      }

      stage('Tomcat Deployment') {

        steps {

          sh """

            ssh jenkins@172.31.86.53 sudo rm -rf /var/tmp/*.war

            ssh jenkins@172.31.86.53 sudo rm -rf /opt/apache-tomcat-9.0.95/webapps/*.war

            scp /var/lib/jenkins/workspace/MonolythicFinalProject/target/*.war jenkins@172.31.86.53:/var/tmp

            ssh jenkins@172.31.86.53 sudo cp /var/tmp/*war /opt/apache-tomcat-9.0.95/webapps/

            ssh jenkins@172.31.86.53 sudo tomdown

            ssh jenkins@172.31.86.53 sudo tomup

            """

          }

        }

      }

    }
```

Create 4 EC2 instance in AWS

To avoid confusion, we can change the hostname of each server, it will be helpful to identify each server.

```
    # hostnamectl hostname tomcat
```

Or

```
   #hostnamectl set-hostname tomcat
```

Exit and relogin

1. Server 4 – Tomcat

Install Tomcat

Install Java 11

```
# yum install java-11
```

```
# cd /opt
```

From Apache official website download the tomcat binary distribution package

https://tomcat.apache.org/download-90.cgi

```
# wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.96/bin/apache-tomcat-9.0.96.zip
```

```
# ls -l
```

Unzip the package

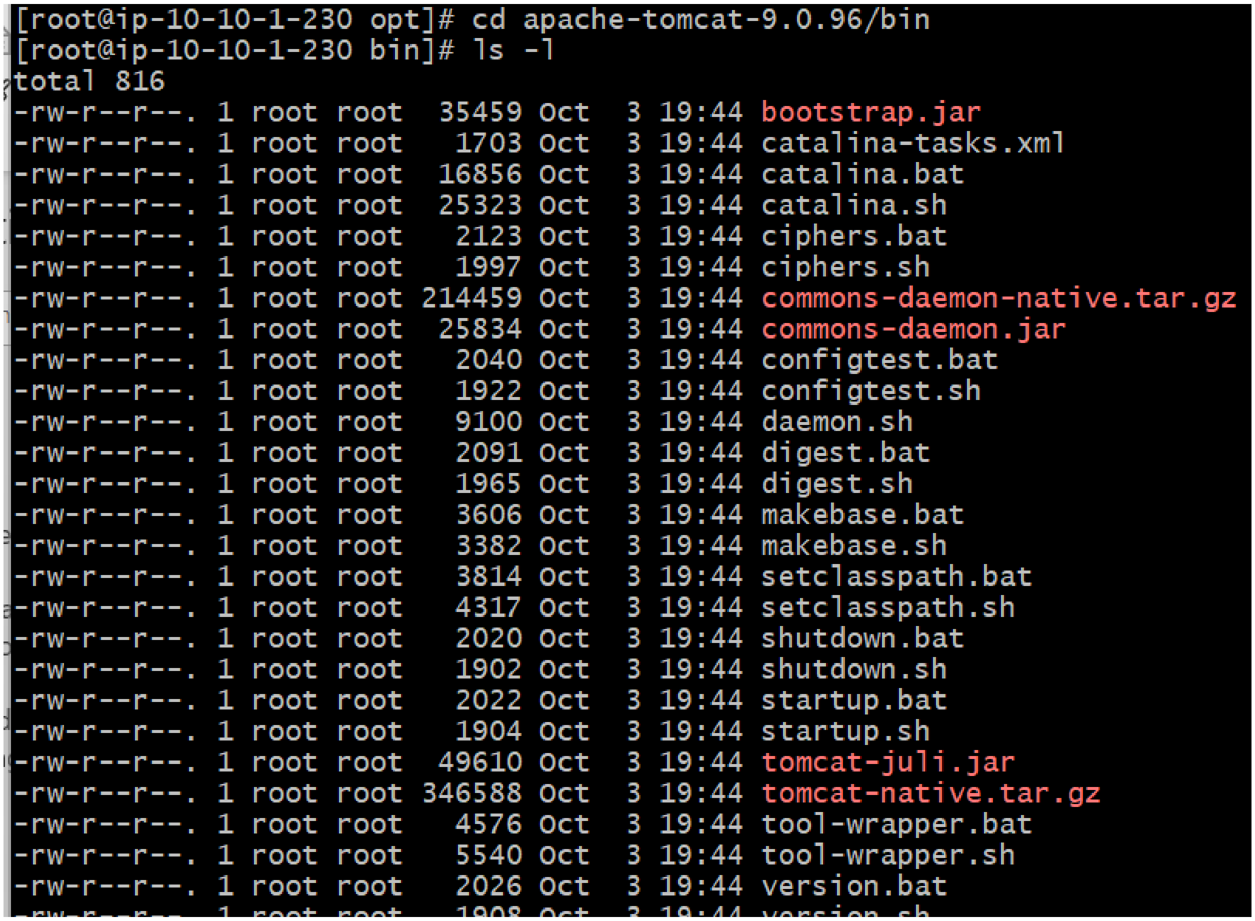
```
# unzip apache-tomcat-9.0.96.zip
```

```
# ls -l
```

To run/start and stop tomcat, we need to set executable permissions for **startup.sh** and **shutdown.sh** scripts.
Locate these files

```
# cd /opt/apache-tomcat-9.0.96/bin
```

```
# ls -l
```



```
[root@ip-10-10-1-230 opt]# cd apache-tomcat-9.0.96/bin
[root@ip-10-10-1-230 bin]# ls -l
total 816
-rw-r--r--. 1 root root  35459 Oct  3 19:44 bootstrap.jar
-rw-r--r--. 1 root root   1703 Oct  3 19:44 catalina-tasks.xml
-rw-r--r--. 1 root root  16856 Oct  3 19:44 catalina.bat
-rw-r--r--. 1 root root  25323 Oct  3 19:44 catalina.sh
-rw-r--r--. 1 root root   2123 Oct  3 19:44 ciphers.bat
-rw-r--r--. 1 root root   1997 Oct  3 19:44 ciphers.sh
-rw-r--r--. 1 root root 214459 Oct  3 19:44 commons-daemon-native.tar.gz
-rw-r--r--. 1 root root  25834 Oct  3 19:44 commons-daemon.jar
-rw-r--r--. 1 root root   2040 Oct  3 19:44 configtest.bat
-rw-r--r--. 1 root root   1922 Oct  3 19:44 configtest.sh
-rw-r--r--. 1 root root   9100 Oct  3 19:44 daemon.sh
-rw-r--r--. 1 root root   2091 Oct  3 19:44 digest.bat
-rw-r--r--. 1 root root   1965 Oct  3 19:44 digest.sh
-rw-r--r--. 1 root root   3606 Oct  3 19:44 makebase.bat
-rw-r--r--. 1 root root   3382 Oct  3 19:44 makebase.sh
-rw-r--r--. 1 root root   3814 Oct  3 19:44 setclasspath.bat
-rw-r--r--. 1 root root   4317 Oct  3 19:44 setclasspath.sh
-rw-r--r--. 1 root root   2020 Oct  3 19:44 shutdown.bat
-rw-r--r--. 1 root root   1902 Oct  3 19:44 shutdown.sh
-rw-r--r--. 1 root root   2022 Oct  3 19:44 startup.bat
-rw-r--r--. 1 root root   1904 Oct  3 19:44 startup.sh
-rw-r--r--. 1 root root  49610 Oct  3 19:44 tomcat-juli.jar
-rw-r--r--. 1 root root 346588 Oct  3 19:44 tomcat-native.tar.gz
-rw-r--r--. 1 root root   4576 Oct  3 19:44 tool-wrapper.bat
-rw-r--r--. 1 root root   5540 Oct  3 19:44 tool-wrapper.sh
-rw-r--r--. 1 root root   2026 Oct  3 19:44 version.bat
-rw-r--r--. 1 root root   1908 Oct  3 19:44 version.sh
```

```
#chmod +x startup.sh shutdown.sh catalina.sh
```

```
# cd /opt/apache-tomcat-9.0.96/bin
# ./startup.sh
```

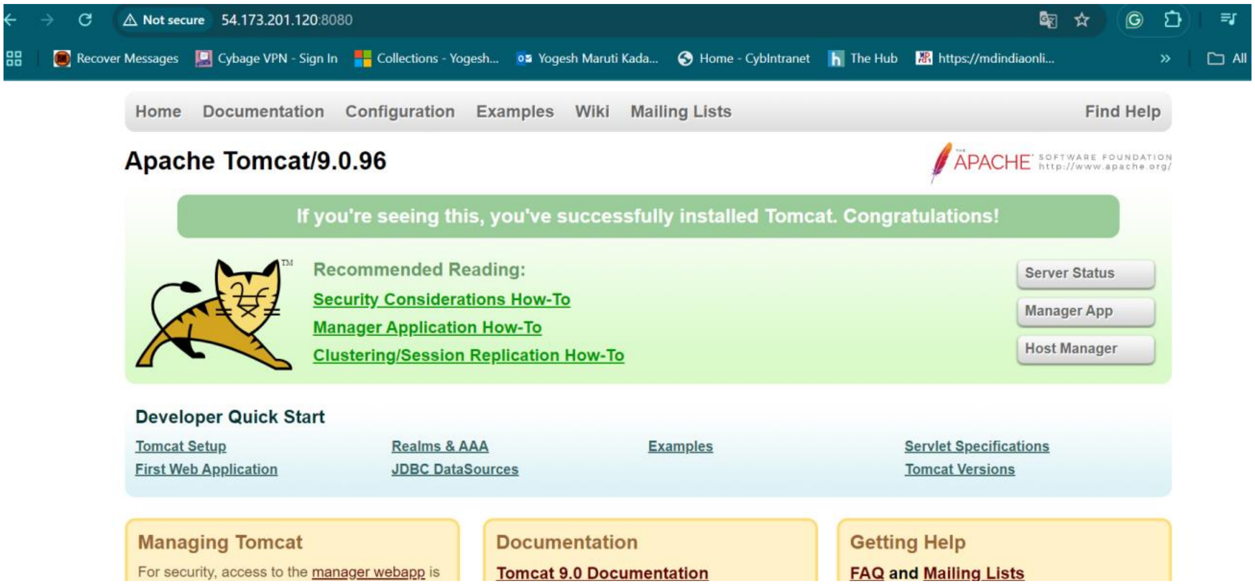
Tomcat Started

You can check using below command

```
# ps -ef | grep tomcat
```

Now, On To check on browser enter public ip and default port 8080 like

54.173.201.120:8080



Apache Tomcat/9.0.96

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

Managing Tomcat
For security, access to the manager webapp is

Documentation
Tomcat 9.0 Documentation

Getting Help
FAQ and Mailing Lists

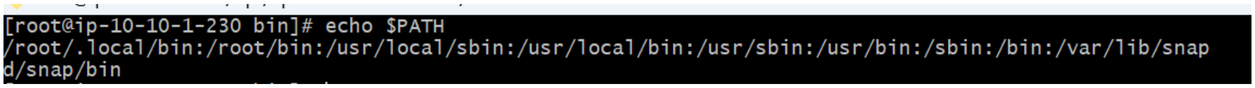
Instead of running startup and shutdown script from this location (/opt/apache-tomcat-9.0.96/bin), We can create **softlink** of this at /usr/local/bin

# ln -s /opt/apache-tomcat-9.0.96/bin/startup.sh /usr/local/bin/tomup

# ln -s /opt/apache-tomcat-9.0.96/bin/shutdown.sh /usr/local/bin/tomdown

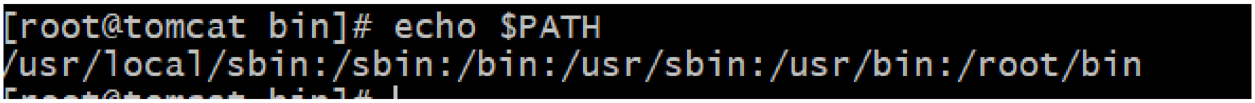
To check default commands path, we can use below command, we can use from it

#echo $PATH



```
[root@ip-10-10-1-230 bin]# echo $PATH
/root/.local/bin:/root/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/var/lib/snap
d/snap/bin
```

In same machine **/usr/local/bin** path is not available, **usr/bin** is available



```
[root@tomcat bin]# echo $PATH
/usr/local/sbin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
```

Then, set softlink path as /usr/bin/tomp and /usr/bin/tomdown,

So softlink path will be,

# ln -s /opt/apache-tomcat-9.0.96/bin/startup.sh /usr/local/bin/tomup

# ln -s /opt/apache-tomcat-9.0.96/bin/shutdown.sh /usr/local/bin/tomdown

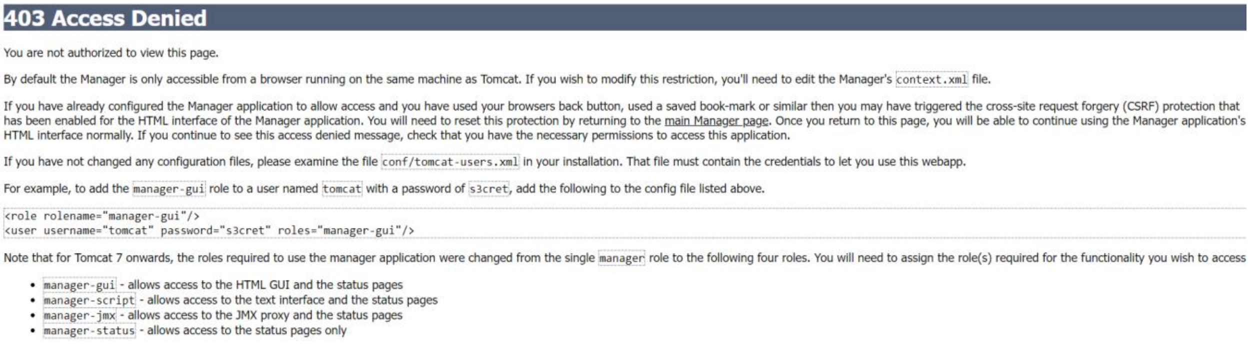
Now we can run tomup and tomdown command to start and shutdown tomcat

#tomup

#tomdown

Now, on tomcat web app, try to open **Manager App**

By Default, Access is denied, it has access from local host only



403 Access Denied

You are not authorized to view this page.

By default the Manager is only accessible from a browser running on the same machine as Tomcat. If you wish to modify this restriction, you'll need to edit the Manager's `context.xml` file.

If you have already configured the Manager application to allow access and you have used your browsers back button, used a saved book-mark or similar then you may have triggered the cross-site request forgery (CSRF) protection that has been enabled for the HTML interface of the Manager application. You will need to reset this protection by returning to the main Manager page. Once you return to this page, you will be able to continue using the Manager application's HTML interface normally. If you continue to see this access denied message, check that you have the necessary permissions to access this application.

If you have not changed any configuration files, please examine the file `conf/tomcat-users.xml` in your installation. That file must contain the credentials to let you use this webapp.

For example, to add the `manager-gui` role to a user named `tomcat` with a password of `s3cret`, add the following to the config file listed above.

```
<role rolename="manager-gui"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
```

Note that for Tomcat 7 onwards, the roles required to use the manager application were changed from the single `manager` role to the following four roles. You will need to assign the role(s) required for the functionality you wish to access

- `manager-gui` - allows access to the HTML GUI and the status pages
- `manager-script` - allows access to the text interface and the status pages
- `manager-jmx` - allows access to the JMX proxy and the status pages
- `manager-status` - allows access to the status pages only

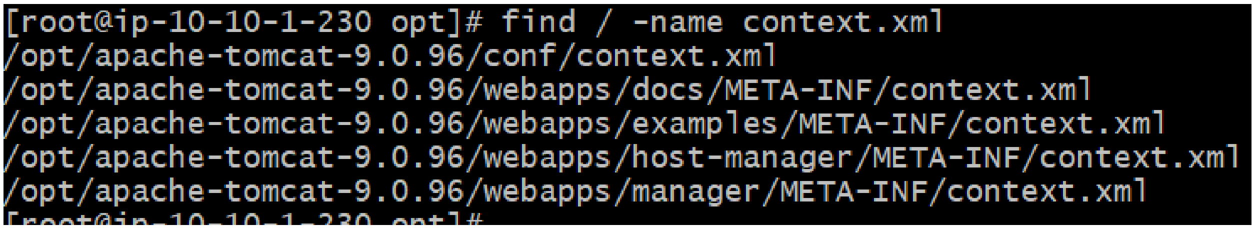
To get access to manage manager-gui from outside, we have to manage some roles and users and edit context.xml file.

# cd /opt

changing a default parameter in context.xml does address this issue

Search for context.xml

find / -name context.xml

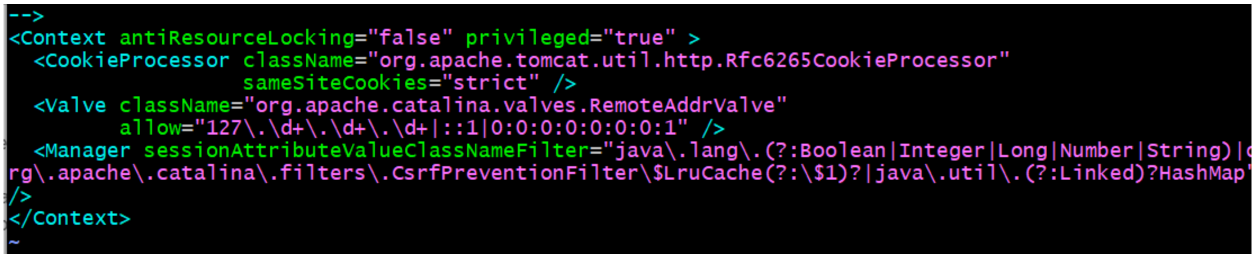


```
[root@ip-10-10-1-230 opt]# find / -name context.xml
/opt/apache-tomcat-9.0.96/conf/context.xml
/opt/apache-tomcat-9.0.96/webapps/docs/META-INF/context.xml
/opt/apache-tomcat-9.0.96/webapps/examples/META-INF/context.xml
/opt/apache-tomcat-9.0.96/webapps/host-manager/META-INF/context.xml
/opt/apache-tomcat-9.0.96/webapps/manager/META-INF/context.xml
[root@ip-10-10-1-230 opt]#
```

above command gives 3 context.xml files. comment or delete () Value ClassName field on files which are under webapp directory. After that restart tomcat services to effect these changes. At the time of writing this lecture below 2 files are updated.

/opt/apache-tomcat-9.0.96/webapps/host-manager/META-INF/context.xml

/opt/apache-tomcat-9.0.96/webapps/manager/META-INF/context.xml



```
-->
<Context antiResourceLocking="false" privileged="true" >
  <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
                   sameSiteCookies="strict" />
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
         allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|String)|o
rg\.apache\.catalina\.filters\.CsrfPreventionFilter\$LruCache(?:\$1)?|java\.util\.(?:Linked)?HashMap"
/>
</Context>
~
```

Delete these two lines from both files

<valve classname=……

Allow=127\……

Use dd command to delete lines

Restart tomcat services

# tomdown

# tomup

Update users' information in the tomcat-users.xml file. Go to tomcat home directory and add below users to conf/tomcat-users.xml file

# cd /opt/apache-tomcat-9.0.96/conf

# vi tomcat-users.xml

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

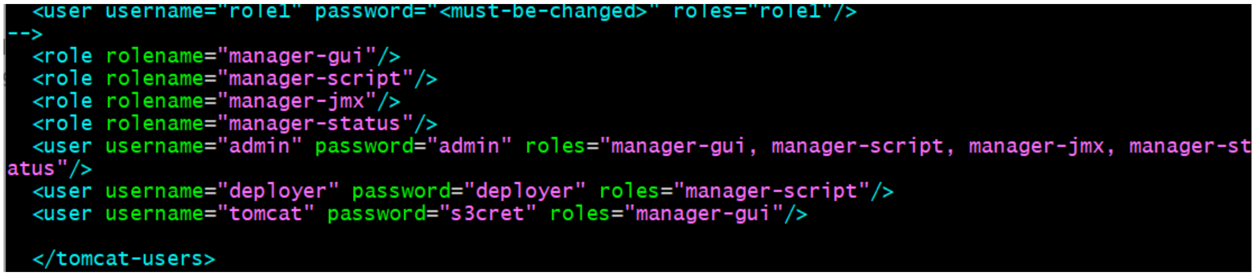
<role rolename="manager-jmx"/>

<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"/>

<user username="tomcat" password="s3cret" roles="manager-gui"/>



```
  <user username="role1" password="<must-be-changed>" roles="role1"/>
-->
  <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-st
atus"/>
  <user username="deployer" password="deployer" roles="manager-script"/>
  <user username="tomcat" password="s3cret" roles="manager-gui"/>

  </tomcat-users>
```

Restart service and try to login to tomcat application from the browser. This time it should be Successful

```
# tomdown
# tomup
```

Now,

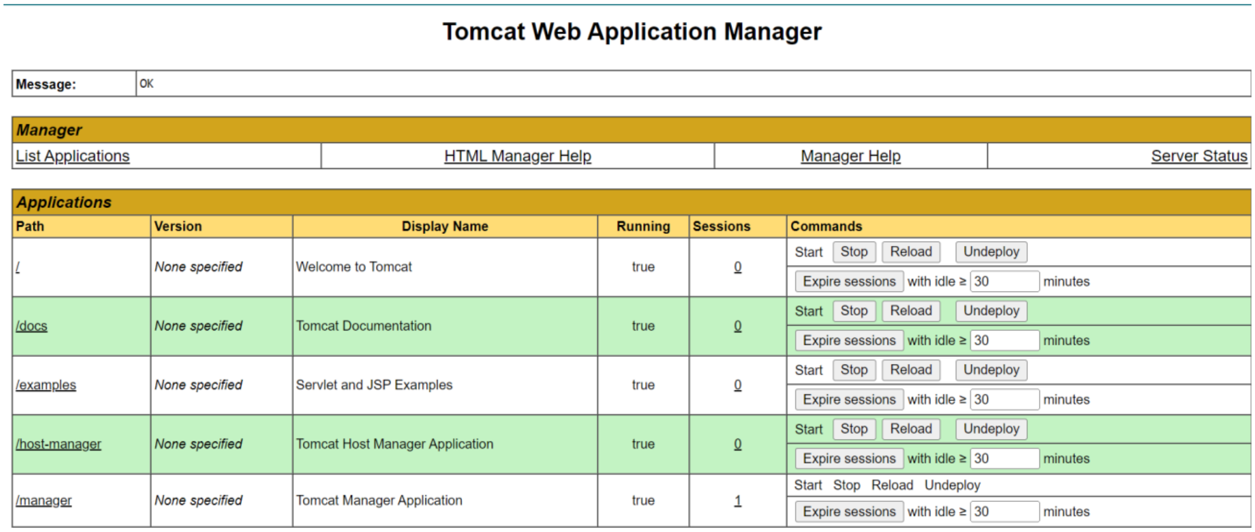
Try again

public_ip:8080 and click on **Manager App**

It will ask for credentials, Enter as we set in user role file

Username – admin

Password – admin



Tomcat Web Application Manager

Message:	OK

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Path	Version	Display Name	Running	Sessions	Commands
/	*None specified*	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	*None specified*	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	*None specified*	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	*None specified*	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	*None specified*	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

You can change tomcat port number by editing server.xml file.

Launch more 3 EC2 instances (Amazon Linux 2, 5.10):

2. Jenkins Server (Server 1)

Install Java 17 for **Jenkins** and Java 11 for **maven** and **Git**

```
#yum install java-17 -y
#yum install java-11 -y
```

```
#java –version
```

Java 17 should be default

Jenkins Installation:

Install Jenkins using Jenkins official website

https://www.jenkins.io/doc/book/installing/linux/#red-hat-centos

```
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
```

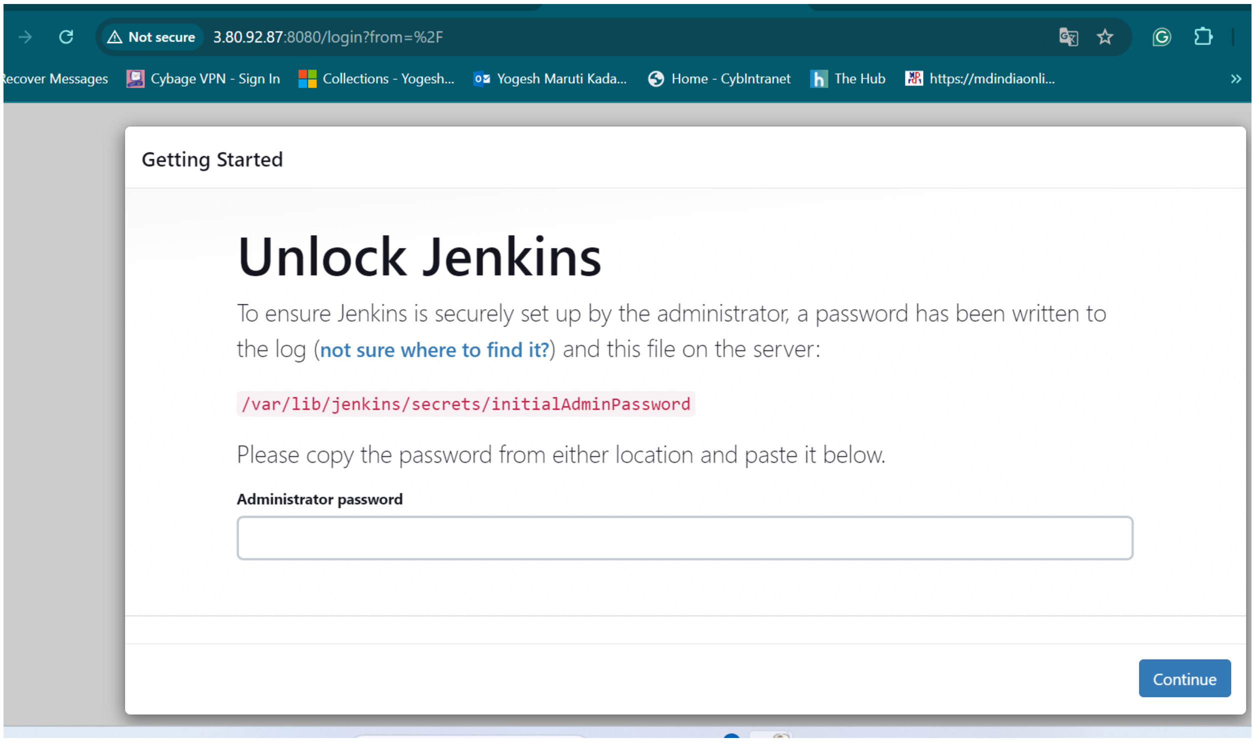
---- skip this if java 17 already installed

```
sudo yum install jenkins -y
```

```
# systemctl start jenkins; systemctl enable jenkins
```

Configure Jenkins:

#Start Jenkins and On browser type "public_ip:8080"



Getting Started

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:

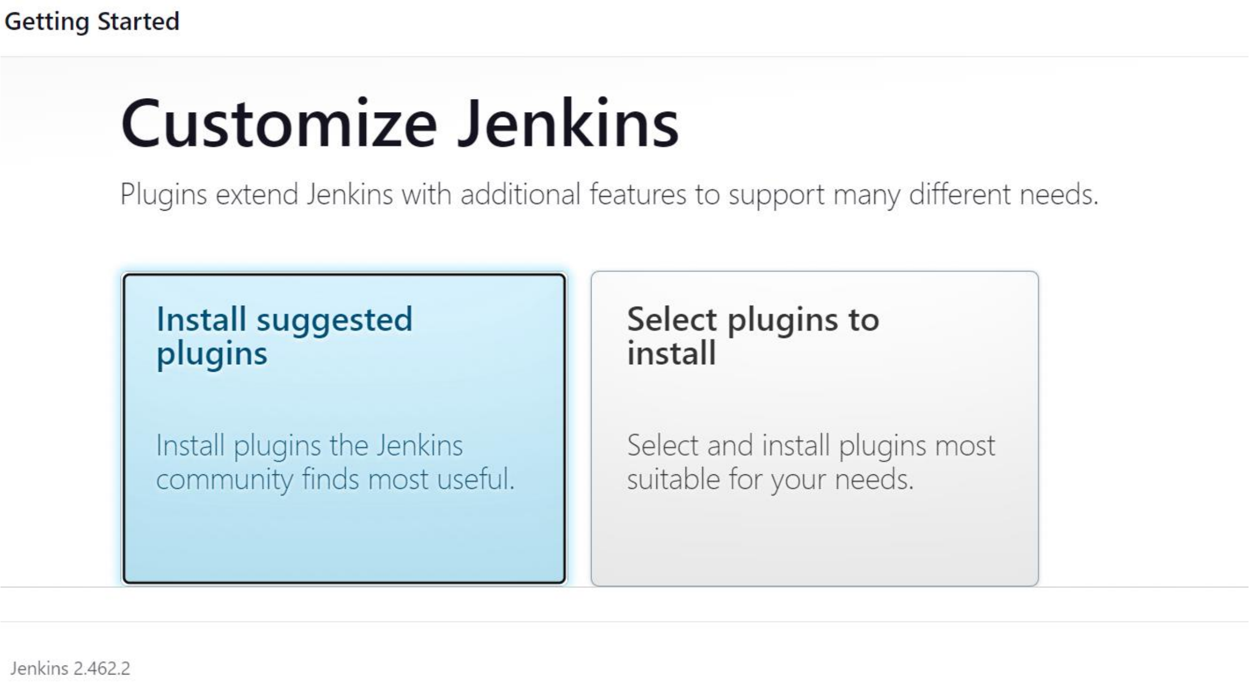
`/var/lib/jenkins/secrets/initialAdminPassword`

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

Administrator password

```
# cat /var/lib/jenkins/secrets/initialAdminPassword
```

Copy and paste password from this location



Getting Started

Customize Jenkins

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

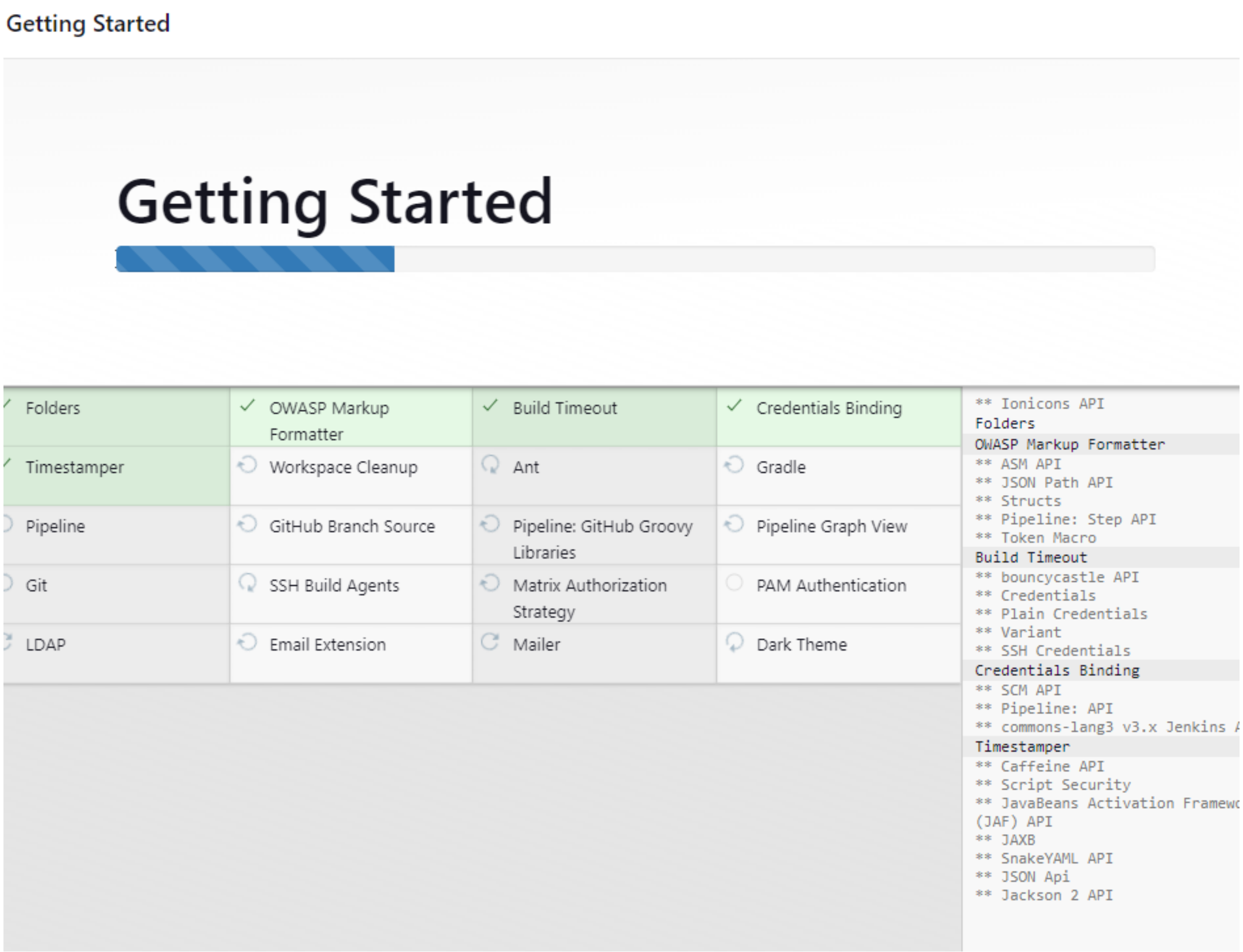
Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.462.2

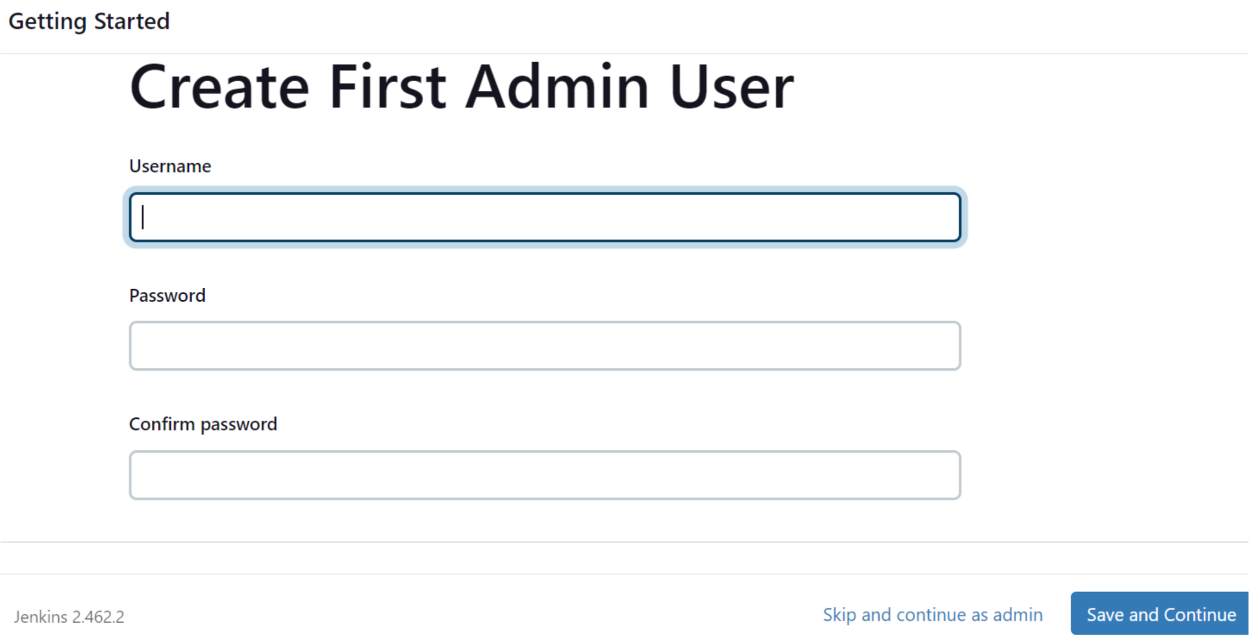


Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline Graph View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	Mailer	Dark Theme

```
** Ionicons API
Folders
OWASP Markup Formatter
** ASM API
** JSON Path API
** Structs
** Pipeline: Step API
** Token Macro
Build Timeout
** bouncycastle API
** Credentials
** Plain Credentials
** Variant
** SSH Credentials
Credentials Binding
** SCM API
** Pipeline: API
** commons-lang3 v3.x Jenkins A
Timestamper
** Caffeine API
** Script Security
** JavaBeans Activation Framewo
(JAF) API
** JAXB
** SnakeYAML API
** JSON Api
** Jackson 2 API
```



Getting Started

Create First Admin User

Username

Password

Confirm password

Install Git & Maven:

Git:

```
# yum install git -y
```

Maven:

Make java 11 default

```
# update-alternatives --config java
```

```
sudo wget http://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-maven.repo -O /etc/yum.repos.d/epel-apache-maven.repo

sudo sed -i s/\$releasever/6/g /etc/yum.repos.d/epel-apache-maven.repo

sudo yum install -y apache-maven
```

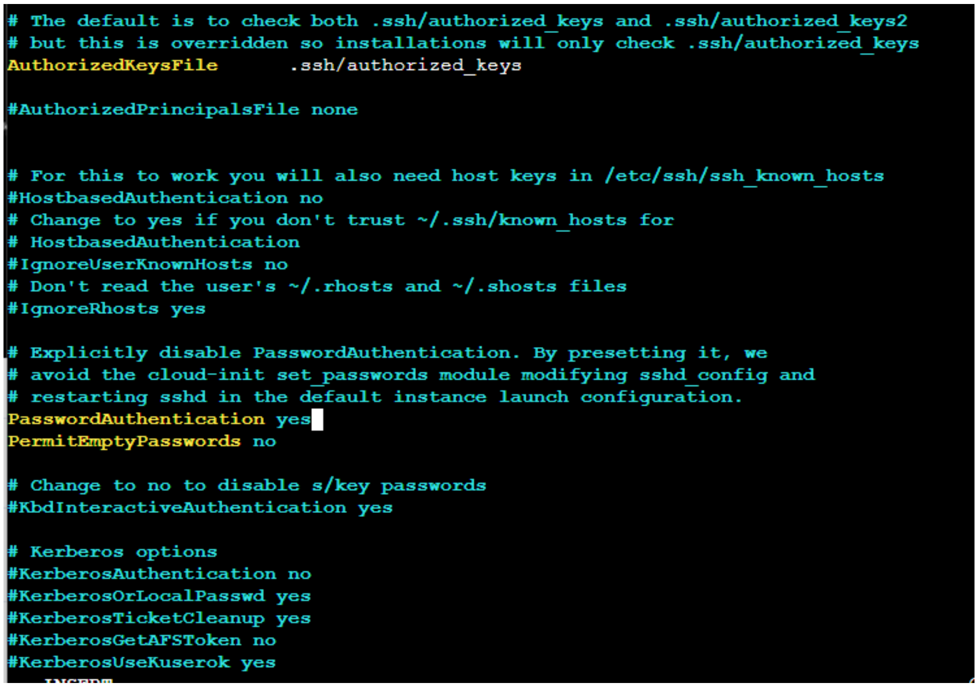
```
# mvn -version
```

Passwordless Authentication from Jenkins to Tomcat server:

We need to copy created artifacts from jenkins server to tomcat server automatically using pipeline. To do this we need password less authentication between both servers.

First with root user, Allow **passwd authentication** to **yes** from config file

```
# vi /etc/ssh/sshd_config
```



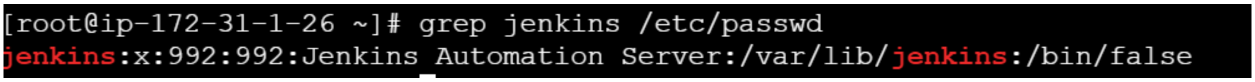
```
# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys

#AuthorizedPrincipalsFile none


# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# Explicitly disable PasswordAuthentication. By presetting it, we
# avoid the cloud-init set_passwords module modifying sshd_config and
# restarting sshd in the default instance launch configuration.
PasswordAuthentication yes
PermitEmptyPasswords no

# Change to no to disable s/key passwords
#KbdInteractiveAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no
#KerberosUseKuserok yes
```

# systemctl restart sshd

Jenkins has it's own jenkins user created by default but we cannot login to that user because this user is under /bin/false shell. So we need to change shell first

To check default shell of jenkins user

#grep Jenkins /etc/passwd



```
[root@ip-172-31-1-26 ~]# grep jenkins /etc/passwd
jenkins:x:992:992:Jenkins Automation Server:/var/lib/jenkins:/bin/false
```

To change shell,

# usermod -s /bin/bash jenkins

Set password for jenkins user

# passwd jenkins

Now try to login,

# su – jenkins

On Tomcat Server (Server 4):

Add one jenkins user,

```
# useradd jenkins
# passwd jenkins
# vi /etc/ssh/sshd_config
```

Set Password Authentication yes

```
# systemctl restart sshd
```

On Jenkins Server (Server 1):

```
# su – jenkins
# ssh-keygen
```

Default path to store key is /var/lib/jenkins/.ssh/id_rsa

```
# cd /var/lib/jenkins/.ssh
# ssh-copy-id jenkins@<private ip of tomcat>
```

It will ask the password. Password will be the password of jenkins user of tomcat server.

To check, try to login to jenkins user of tomcat server from jenkins user of jenkins server.

Create one file and try to copy from jenkins to jenkins user of tomcat tomcat.

Example –

```
# cd /tmp
# touch abc
# scp abc jenkins@<private ip of tomcat>:/var/tmp
```

abc file should be copy from jenkins to tomcat.

On Tomcat Server:

IMP – We have copied file to jenkins user of tomcat server but,

By default, tomcat is running on root user.

And we are going to copy artifacts to webapps directory using jenkins user.

As jenkins has not root privileges, it will not allow to copy.

To avoid this, we will assign root privileges to jenkins user

Login as a root user

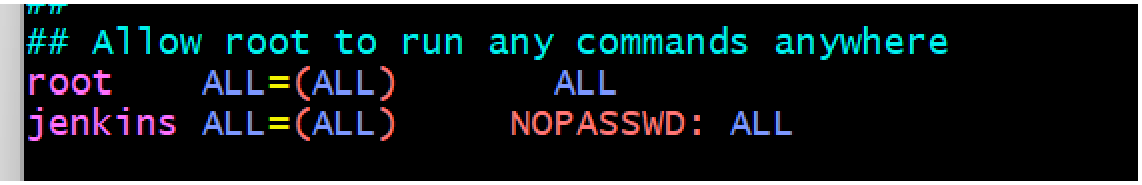
```
# su -i
# visudo
```

Below,

```
root        ALL=(ALL)       ALL
```

Add line,

```
Jenkins     ALL=(ALL)       NOPASSWD:   ALL
```



```
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
jenkins ALL=(ALL)     NOPASSWD: ALL
```

Now, try to copy abc file from /tmp folder to webapps folder

```
# sudo cp /var/tmp/abc /opt/apache-tomcat-9.0.96/webapps/
```

Test abc file should be copy

Remove this file

```
# sudo rm /opt/apache-tomcat-9.0.96/webapps/abc
```

3. Install SonarQube (Server 2):

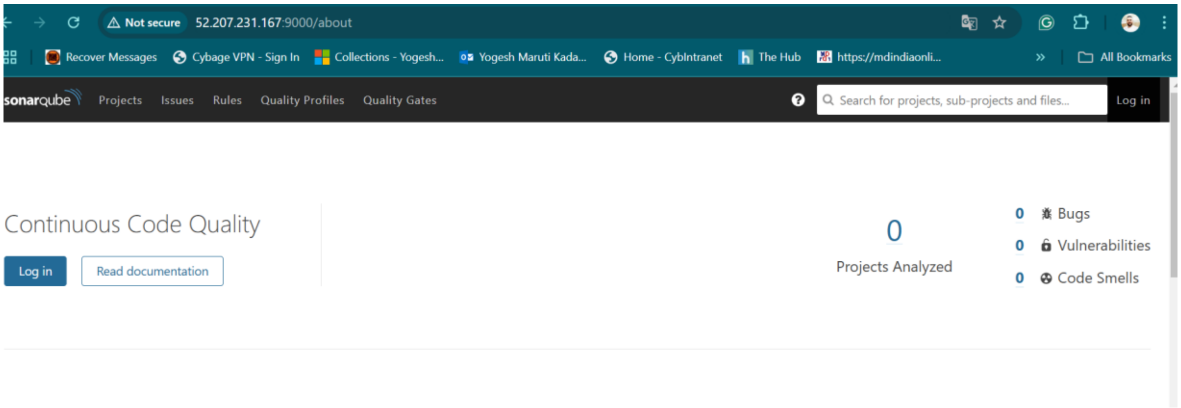
```
yum install java-11
sudo wget -O /etc/yum.repos.d/sonar.repo
http://downloads.sourceforge.net/project/sonar-pkg/rpm/sonar.repo
ls
sudo yum install sonar -y
service sonar start
cd /
ls
cd opt/
ls
```

```
      cd sonar/
      ls
      cd conf/
    # /opt/sonar/conf
      ls
      vi sonar.properties      ------ No need of this step for installation
      service sonar stop
      service sonar start
     # ps -ef | grep sonar     --------- To check
```

Now,

On browser

Publicip:9000



Continuous Code Quality

Log in | Read documentation

0 Projects Analyzed

0 Bugs

0 Vulnerabilities

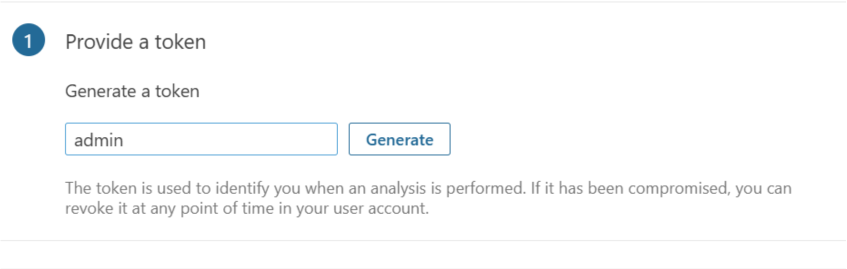
0 Code Smells

login

Username – admin

Password – admin

Create token - type admin – generate – continue

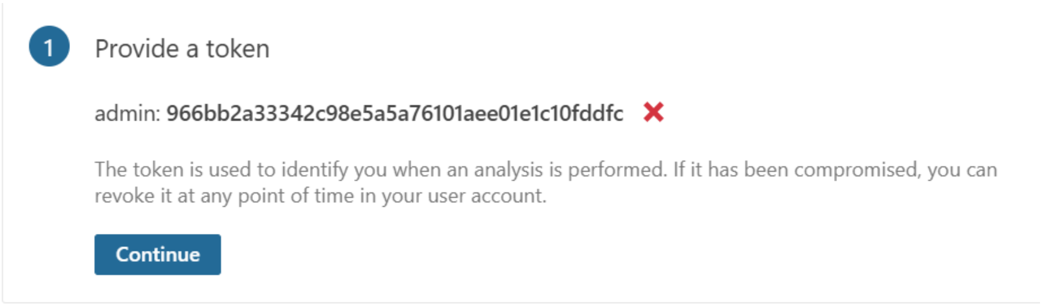


1 Provide a token

Generate a token

admin | Generate

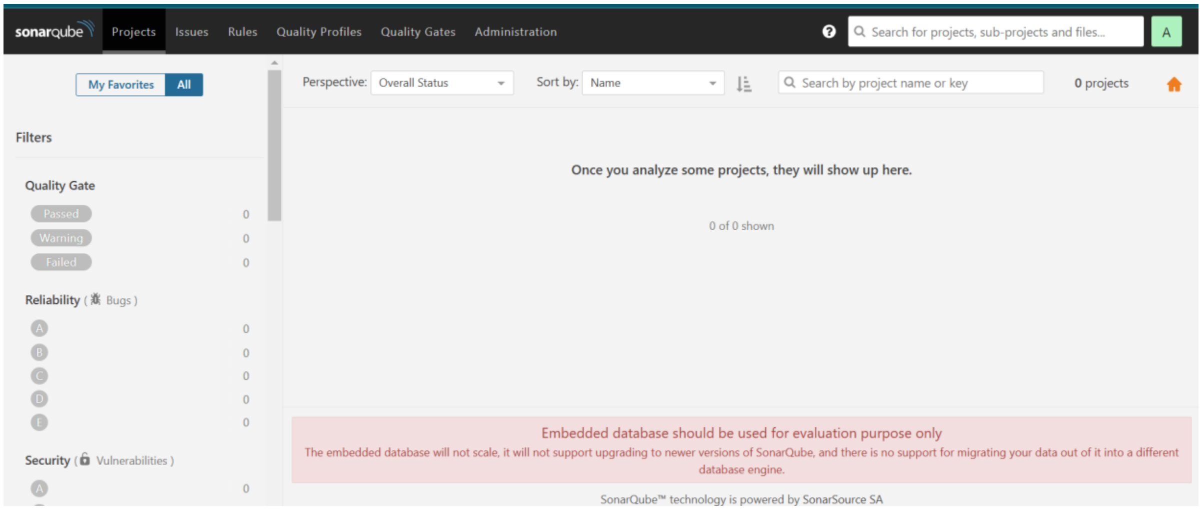
The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point of time in your user account.



1 Provide a token

admin: 966bb2a33342c98e5a5a76101aee01e1c10fddfc ✖

The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point of time in your user account.



Perspective: Overall Status

Sort by: Name

Search by project name or key

0 projects

Once you analyze some projects, they will show up here.

0 of 0 shown

Embedded database should be used for evaluation purpose only

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA

4. Install Nexus (Server 3):

```
cd /opt

sudo yum install java-11 -y

sudo wget https://download.sonatype.com/nexus/3/latest-unix.tar.gz

ls

tar -xvf latest-unix.tar.gz

ls

mv nexus-3.74.0-05 nexus3

ls -l

chown -R ec2-user:ec2-user nexus3 sonatype-work

ls -l

cd nexus3/
```

```
ls
cd bin
ls
vi nexus.rc
```

Uncomment the line & add “ec2-user”

```
run-as-user= “ec2-user”
```

```
ln -s /opt/nexus3/bin/nexus /etc/init.d/nexus
cd /etc/init.d/
```

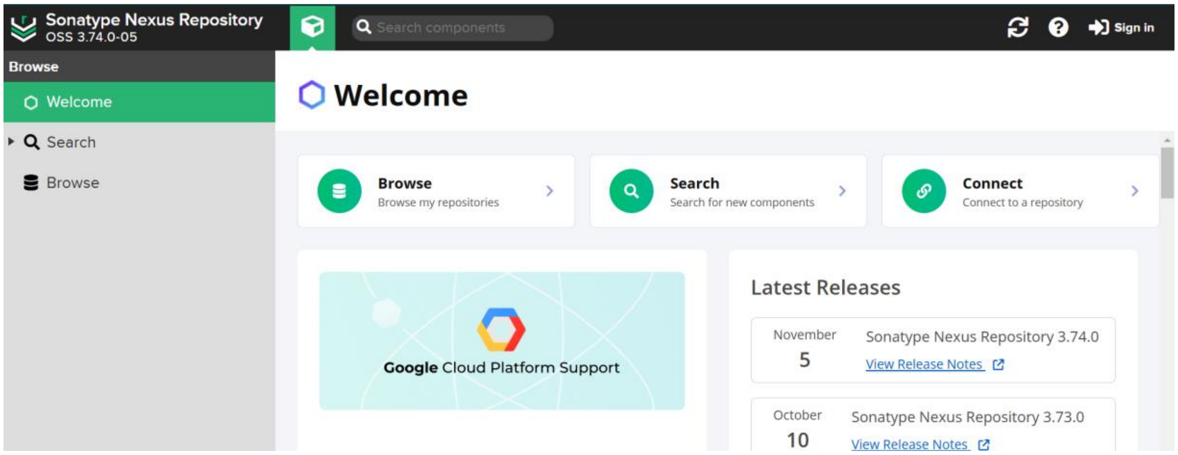
----- you can ignore this step

```
chkconfig --add nexus
chkconfig nexus on
sudo service nexus start
```

Now,

On browser

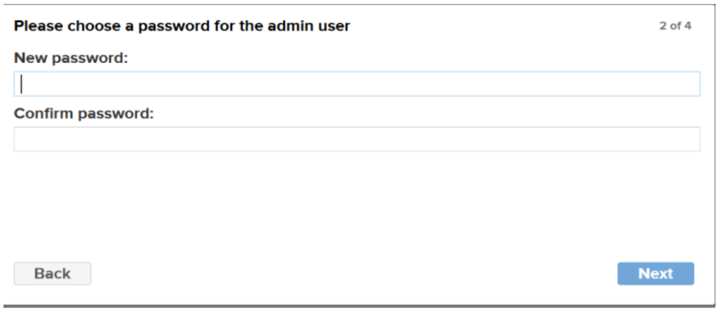
Publicip:8081



Sign In

Username – admin

Password – Located in Linux machine at /opt/sonatype-work/nexus3/admin.password



Please choose a password for the admin user 2 of 4

New password:

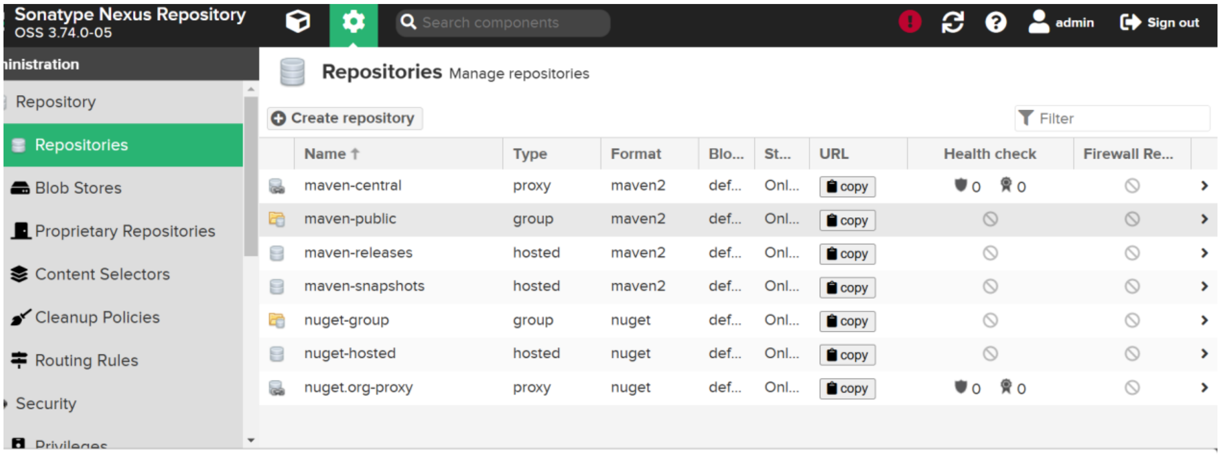
Confirm password:

- Set New Password
- Finish

Check On settings – Repository - Repository – Use these default one maven-release and maven-snapshot

Maven release – Ready to Release artifact

Maven snapshot – Staging artifact

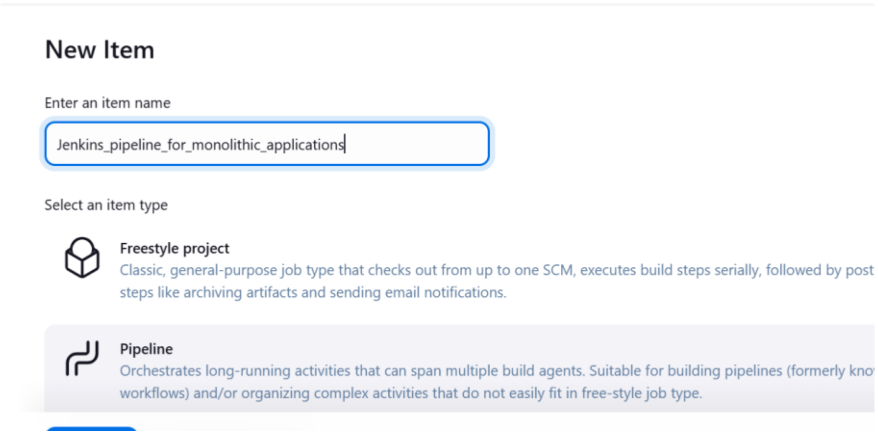


Repositories Manage repositories

Name ↑	Type	Format	Blo...	St...	URL	Health check	Firewall Re...
maven-central	proxy	maven2	def...	Onl...	copy	0 0	⃠
maven-public	group	maven2	def...	Onl...	copy	⃠	⃠
maven-releases	hosted	maven2	def...	Onl...	copy	⃠	⃠
maven-snapshots	hosted	maven2	def...	Onl...	copy	⃠	⃠
nuget-group	group	nuget	def...	Onl...	copy	⃠	⃠
nuget-hosted	hosted	nuget	def...	Onl...	copy	⃠	⃠
nuget.org-proxy	proxy	nuget	def...	Onl...	copy	0 0	⃠

Create Jenkins Pipeline:

- **New Item ---> Pipeline**



New Item

Enter an item name

Jenkins_pipeline_for_monolithic_applications

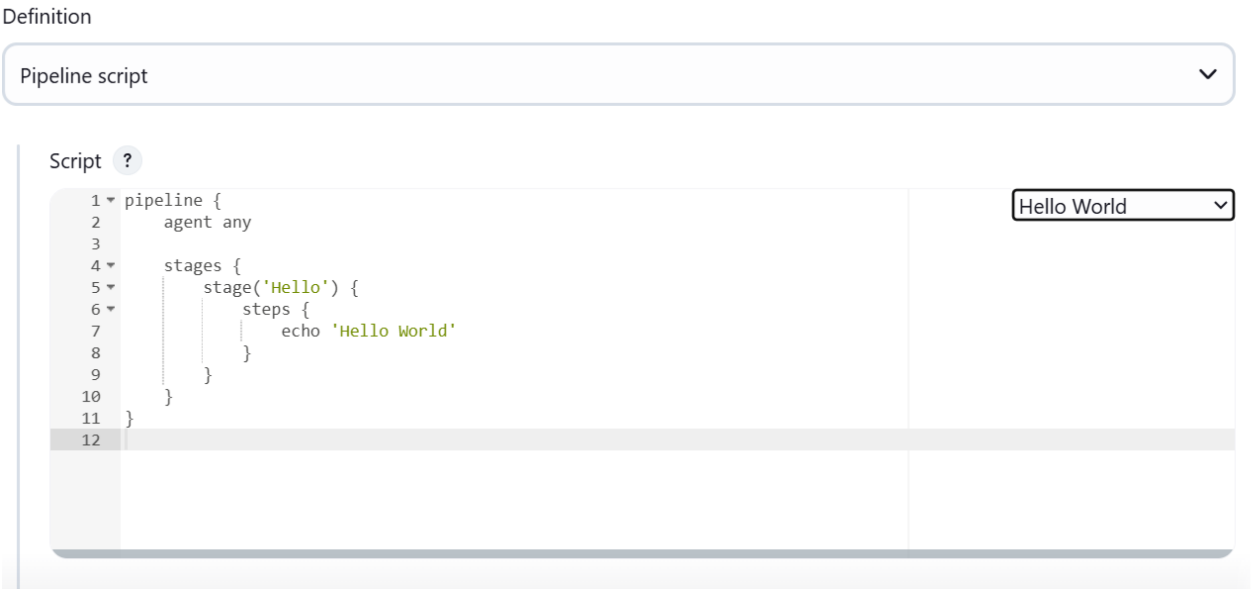
Select an item type

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

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

Pipeline:

This is simple Hello World groovy script.



Definition

Pipeline script

Script ?

Hello World

```
pipeline {
    agent any

    stages {
        stage('Hello') {
            steps {
                echo 'Hello World'
            }
        }
    }
}

```

We will add stages step by step:

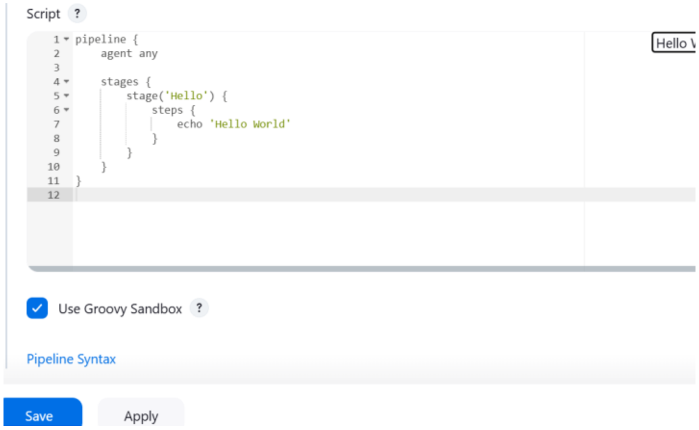
pipeline {

    agent any

    stages {

Stage 1: Git checkout

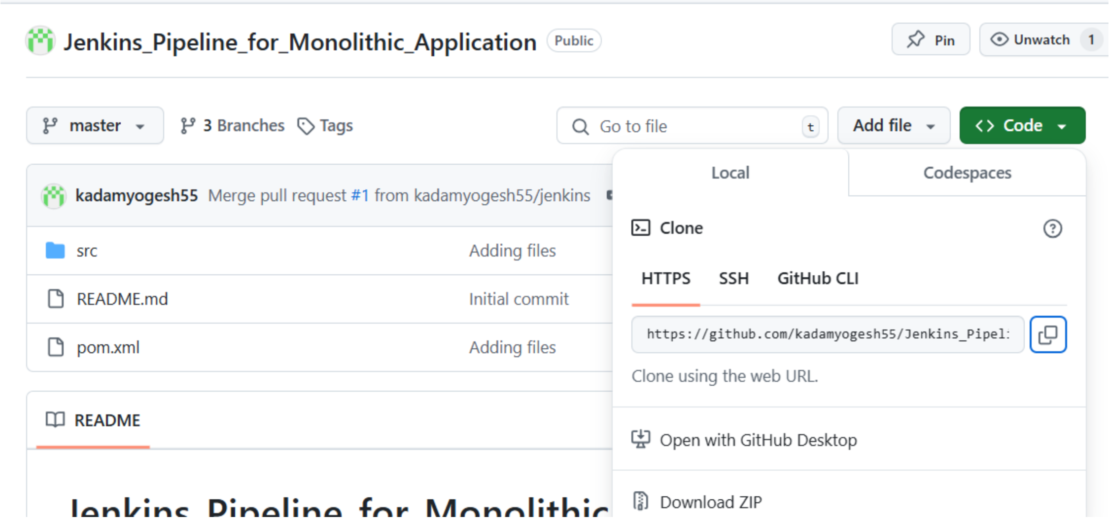
Need to generate pipeline script for git, click on Pipeline syntax



```
pipeline {
    agent any

    stages {
        stage('Hello') {
            steps {
                echo 'Hello World'
            }
        }
    }
}
```

☑ Use Groovy Sandbox ?

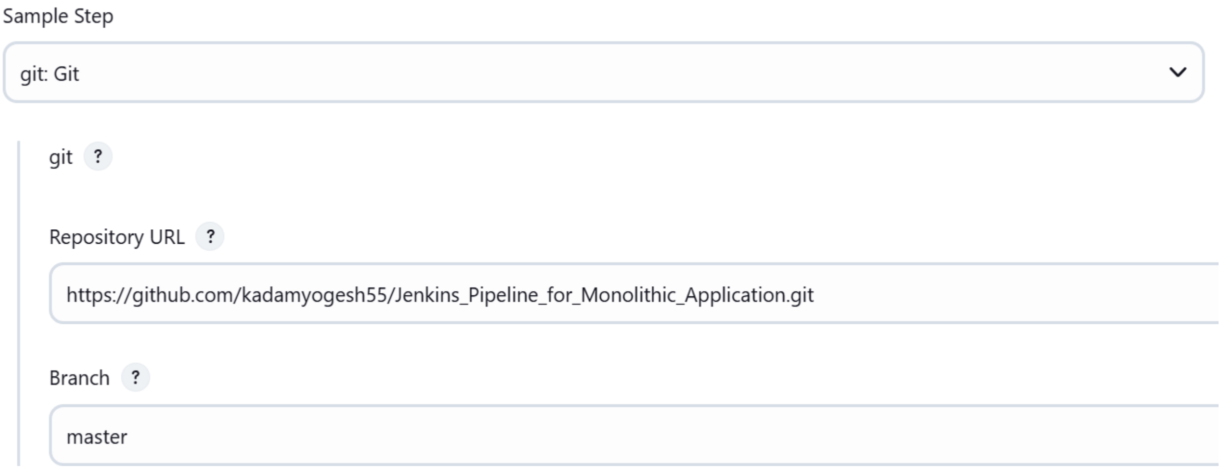
Copy your GitHub repository url as below



```
https://github.com/kadamyogesh55/Jenkins_Pipel:
```

Clone using the web URL.

In pipeline syntax paste it.



Sample Step

git: Git

git ?

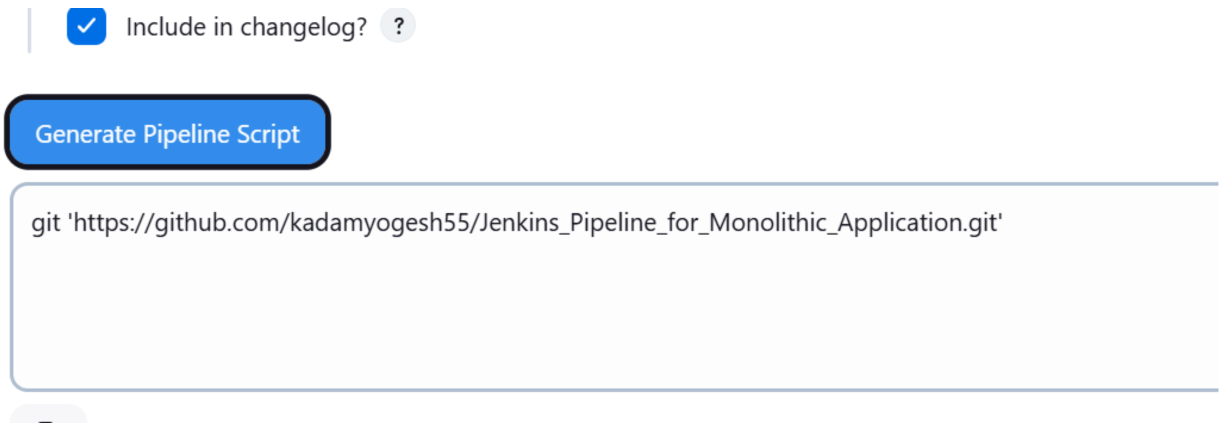
Repository URL ?

https://github.com/kadamyogesh55/Jenkins_Pipeline_for_Monolithic_Application.git

Branch ?

master

Click on Generate Pipeline Script



```
git 'https://github.com/kadamyogesh55/Jenkins_Pipeline_for_Monolithic_Application.git'
```

Copy this Syntax and write stage as below,

```
 stage('git checkout') {
        steps {
         git 'https://github.com/kadamyogesh55/Jenkins_Pipeline_for_Monolithic_Application.git'
        }
      }
```

Stage 2: Sonar Testing

```
      stage('Sonar testing') {
        steps {
         sh 'mvn sonar:sonar'
        }
      }
```

For this stage we will modify pom.xml file and provide IP address of Sonar Server, so it will automate

Stage 3: Maven Build

```
      stage('maven Build') {
        steps {
         sh 'mvn clean package'
        }
      }
```

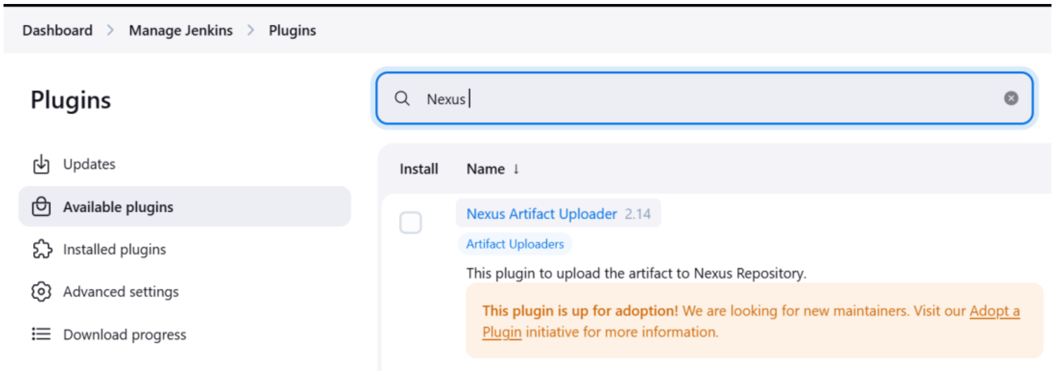
We will change the build version in pom.xml file as we set in the groovy script or vice versa

Stage 4: Nexus Upload

To generate script for Nexus, we need to install **Nexus artifact uploader** Plugins**:**

Install Plugins for Nexus Artifact Uploader and add credentials:

- Go to **Manage Jenkins**
- Search for Nexus Artifact Uploader
- Install Plugin



Nexus|

□ Nexus Artifact Uploader 2.14

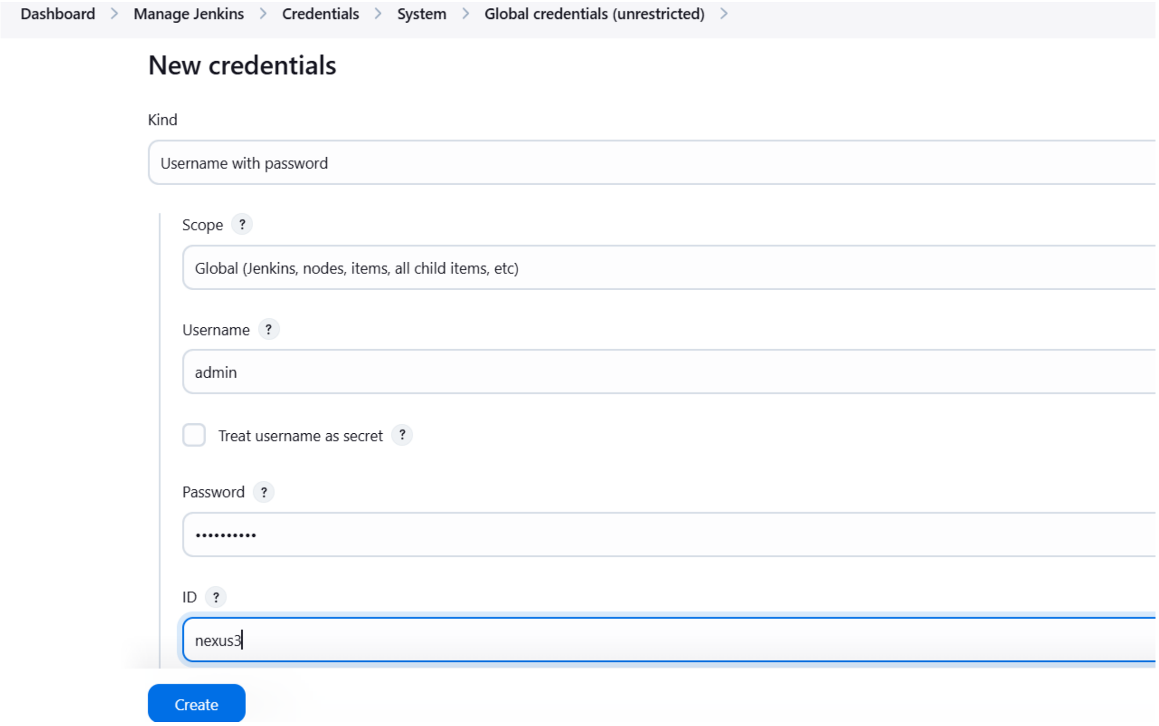
Artifact Uploaders

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.

Now, Create **Credentials:**

- Go to Manage Jenkins
- Credentials
- Global
- Add Credentials
- Username – admin
- Password – Set as you wish
- ID – nexus3
- Create



Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

New credentials

Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

admin

□ Treat username as secret ?

Password ?

••••••••••

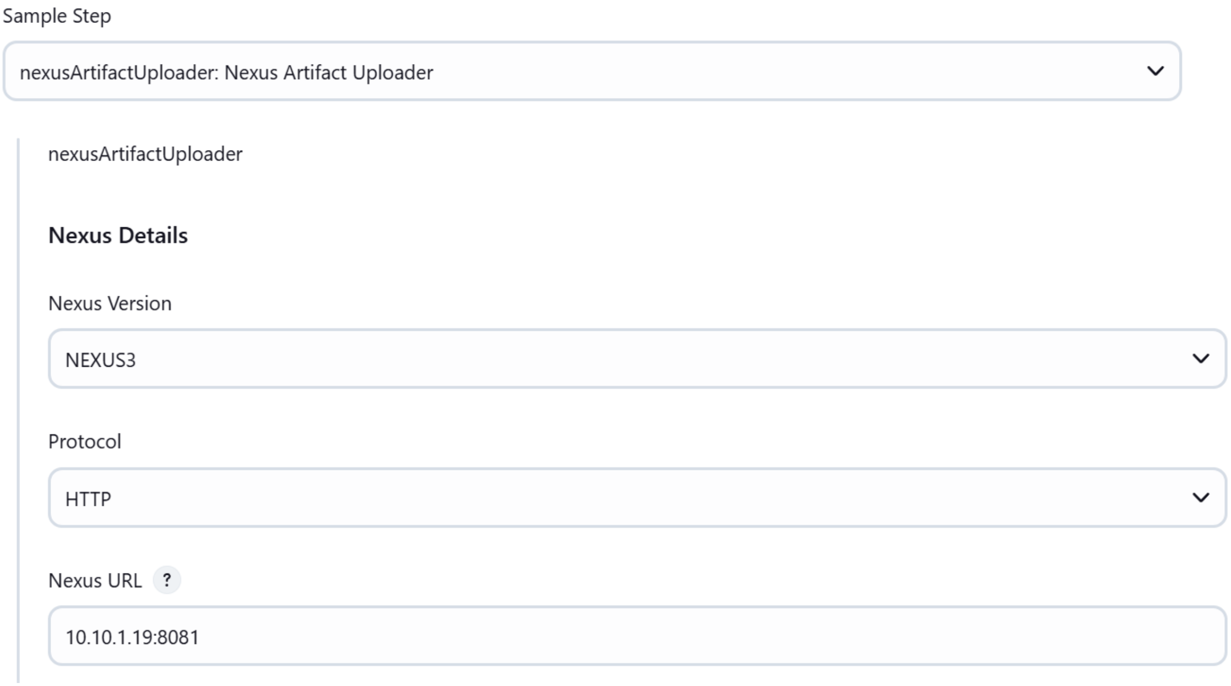
ID ?

nexus3

Now,

To create script for nexus, click on **Pipeline Syntax**

Select nexus from dropdown,



Sample Step

nexusArtifactUploader: Nexus Artifact Uploader

nexusArtifactUploader

Nexus Details

Nexus Version

NEXUS3

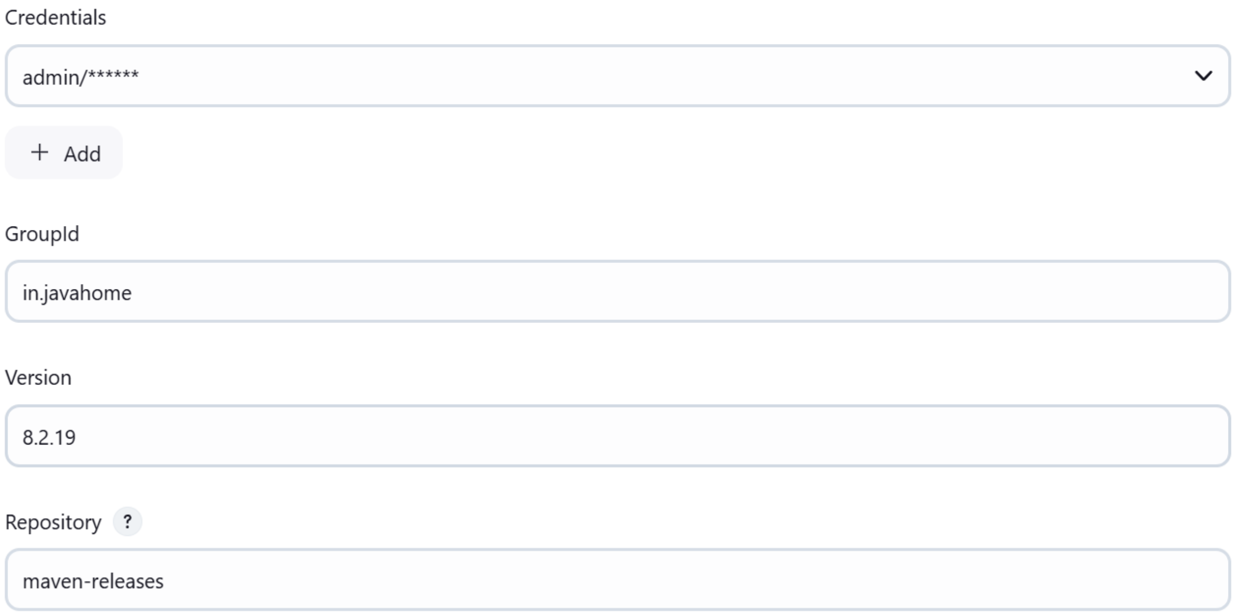
Protocol

HTTP

Nexus URL ?

10.10.1.19:8081

In URL add private ip address of nexus.



Credentials

admin/******

GroupId

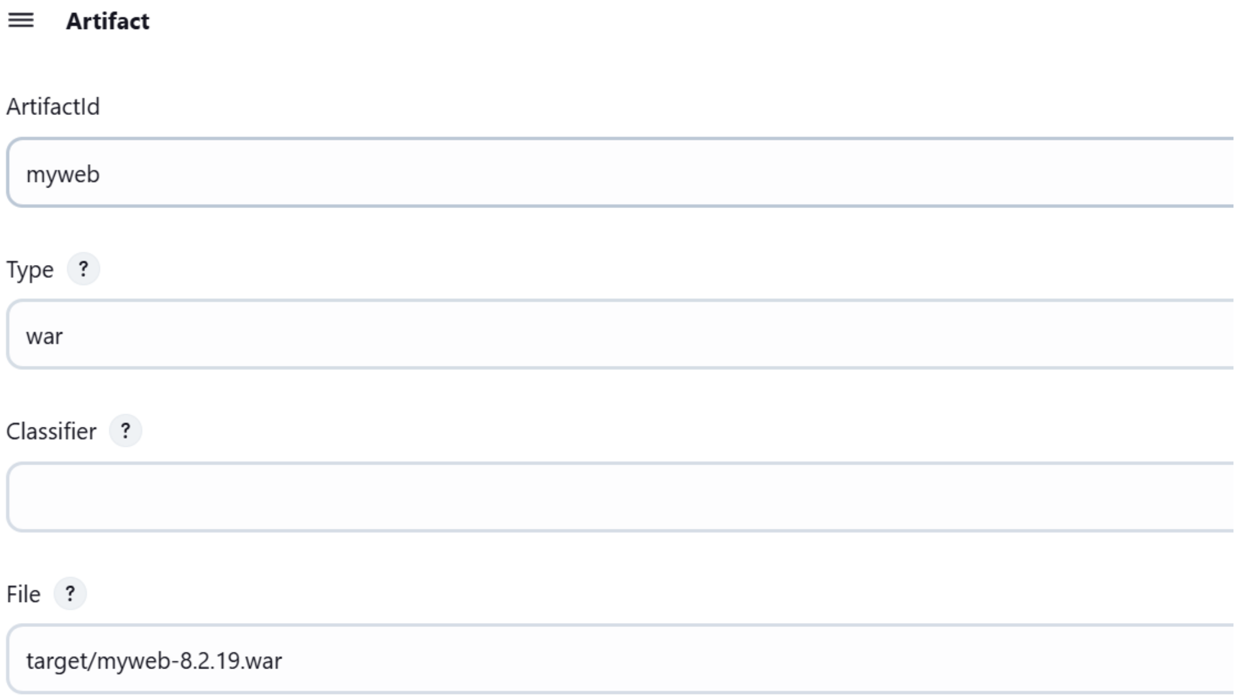
in.javahome

Version

8.2.19

Repository ?

maven-releases



Artifact

ArtifactId

myweb

Type ?

war

Classifier ?

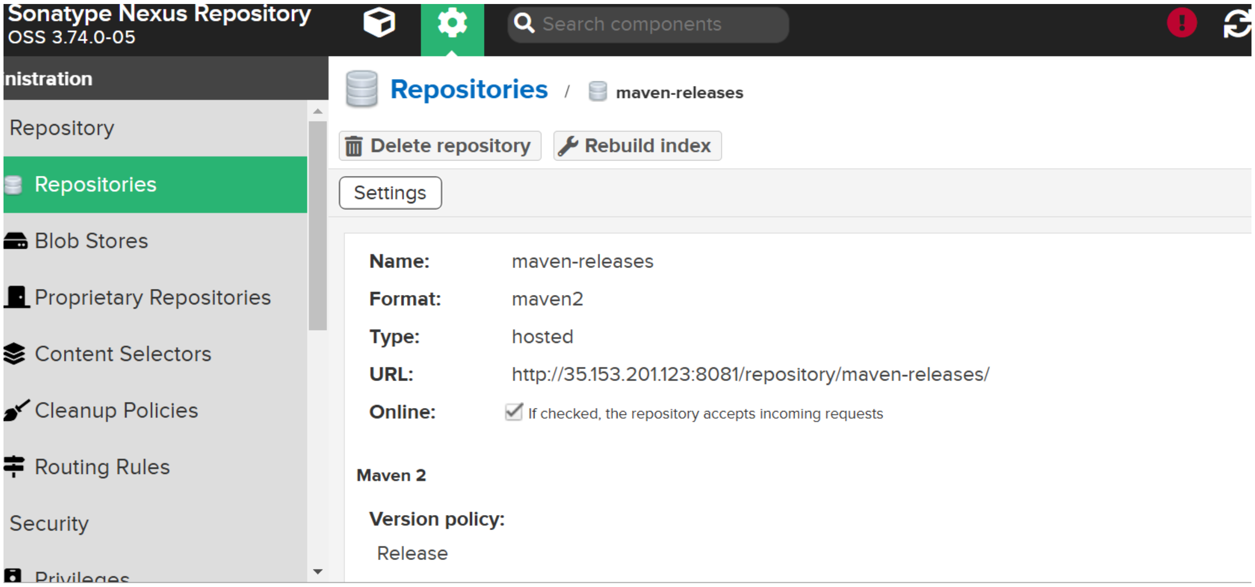
File ?

target/myweb-8.2.19.war

- Select credentials created earlier
- In **pom.xml,** we will get group id, version, artifact id, type & file

```
<groupId>in.javahome</groupId>
<artifactId>myweb</artifactId>
<packaging>war</packaging>
<version>8.2.19</version>
```

For **Repository,** we will get it from Nexus server. **nexus-release** or **nexus-snapshot**



Name: maven-releases

Format: maven2

Type: hosted

URL: http://35.153.201.123:8081/repository/maven-releases/

Online: ☑ If checked, the repository accepts incoming requests

Maven 2

Version policy:

Release

Click on **Generate script**

script like below will be generated

nexusArtifactUploader artifacts: [[artifactId: 'myweb', classifier: '', file: 'target/myweb-8.2.19.war', type: 'war']], credentialsId: 'nexus3', groupId: 'in.javahome', nexusUrl: '10.10.1.19:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'maven-releases', version: '8.2.19'

Add this script in this stage

```
stage('Nexus upload') {
  steps {
    nexusArtifactUploader artifacts: [[artifactId: 'myweb', classifier: '', file: 'target/myweb-8.2.19.war', type: 'war']], credentialsId: 'nexus3', groupId: 'in.javahome', nexusUrl: '10.10.1.19:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'maven-releases', version: '8.2.19'
  }
}
```

Stage 5: Tomcat Deployment

Copy **private IP Address** of tomcat server and paste here

```
    stage('Tomcat Deployment') {
        steps {
          sh """
             ssh jenkins@10.10.1.49 sudo rm -rf /var/tmp/*.war
             ssh jenkins@10.10.1.49 sudo rm -rf /opt/apache-tomcat-9.0.96/webapps/*.war
             scp /var/lib/jenkins/workspace/Jenkins_pipeline_for_monolithic_application/target/*.war jenkins@10.10.1.49:/var/tmp
             ssh jenkins@10.10.1.49 sudo cp /var/tmp/*war /opt/apache-tomcat-9.0.96/webapps/
             ssh jenkins@10.10.1.49 sudo tomdown
             ssh jenkins@10.10.1.49 sudo tomup
             """
          }
        }
      }
    }
```

Final Pipeline for My repo:

Final Pipeline:

```
pipeline {
    agent any

    stages {
      stage('git checkout') {
        steps {
          git 'https://github.com/kadamyogesh55/Jenkins_Pipeline_for_Monolithic_Application.git'
         }
      }
      stage('Sonar testing') {
```

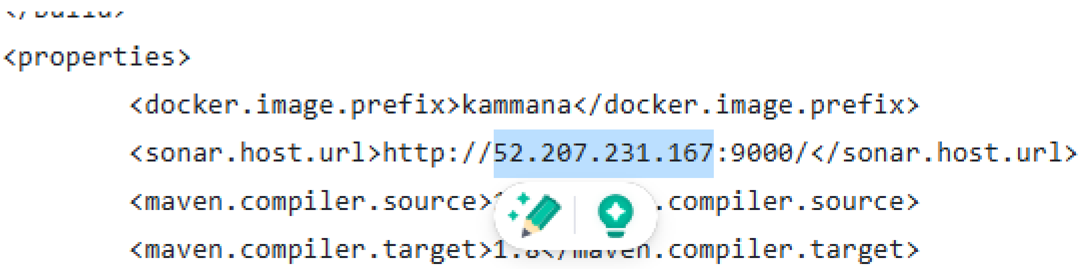
```
      steps {
        sh 'mvn sonar:sonar'
      }
    }
    stage('maven Build') {
      steps {
        sh 'mvn clean package'
      }
    }
    stage('Nexus upload') {
      steps {
        nexusArtifactUploader artifacts: [[artifactId: 'myweb', classifier: '', file: 'target/myweb-8.2.19.war', type: 'war']], credentialsId: 'nexus3', groupId: 'in.javahome', nexusUrl: '10.10.1.19:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'maven-releases', version: '8.2.19'
      }
    }
   stage('Tomcat Deployment') {
      steps {
        sh """
          ssh jenkins@10.10.1.49 sudo rm -rf /var/tmp/*.war
          ssh jenkins@10.10.1.49 sudo rm -rf /opt/apache-tomcat-9.0.96/webapps/*.war
          scp /var/lib/jenkins/workspace/Jenkins_pipeline_for_monolithic_application/target/*.war jenkins@10.10.1.49:/var/tmp
          ssh jenkins@10.10.1.49 sudo cp /var/tmp/*war /opt/apache-tomcat-9.0.96/webapps/
          ssh jenkins@10.10.1.49 sudo tomdown
          ssh jenkins@10.10.1.49 sudo tomup

          """
        }
      }
    }
  }
```

- In this groovy script , for nexus change artifacts **version** number and **private ip** address
- Change **scp** path

Modify pom.xml file for IP address and artifacts version:

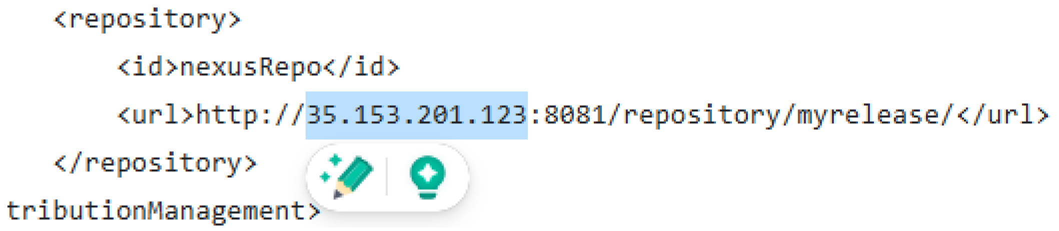
- On Git Hub modify pom.xml file for Sonar & Nexus
- Open pom.xml file in GitHub
- Change Sonarqube IP Address
- Change IP Address for nexus two times (for snapshot & release)
- Change artifact version as you set in pipeline (e.g myweb-8.2.19.war')
- Commit

-



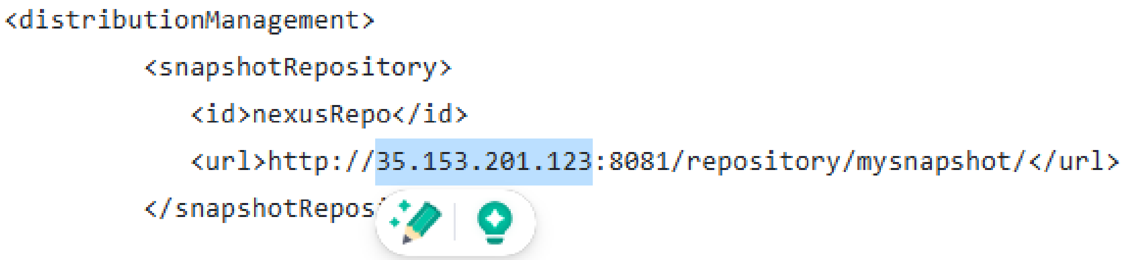
```
</build>
<properties>
        <docker.image.prefix>kammana</docker.image.prefix>
        <sonar.host.url>http://52.207.231.167:9000/</sonar.host.url>
        <maven.compiler.source>          .compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
```

-



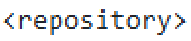
```
<repository>
    <id>nexusRepo</id>
    <url>http://35.153.201.123:8081/repository/myrelease/</url>
</repository>
tributionManagement>
```

-

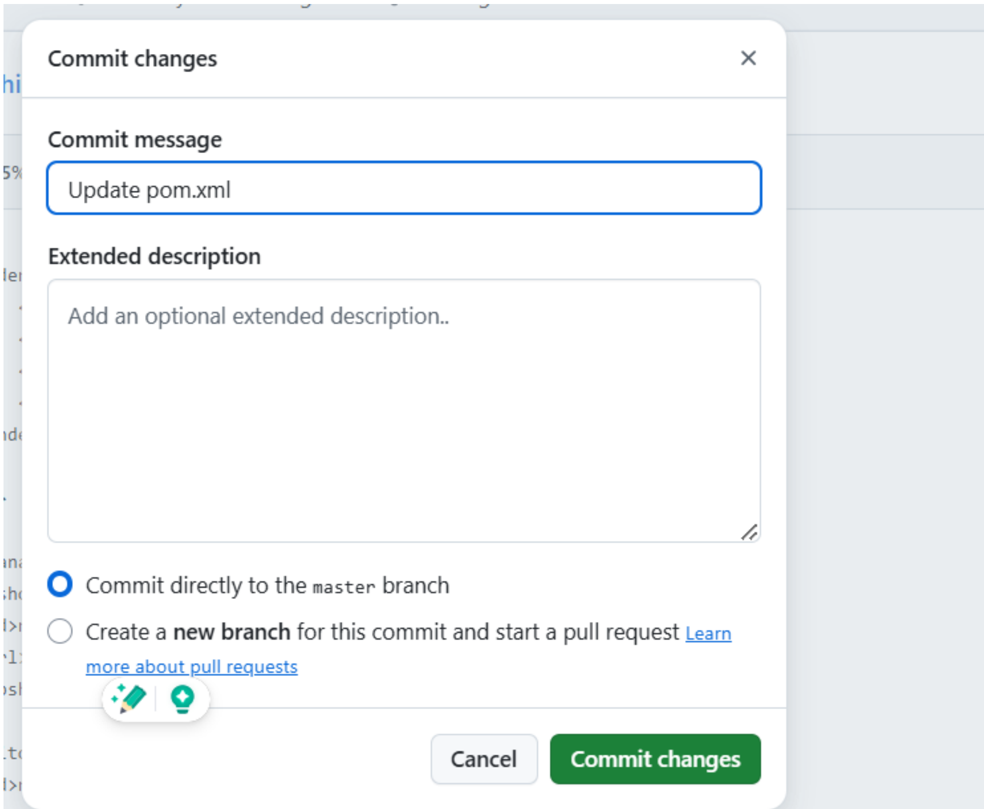


```
<distributionManagement>
         <snapshotRepository>
            <id>nexusRepo</id>
            <url>http://35.153.201.123:8081/repository/mysnapshot/</url>
         </snapshotRepos
```

-



```
<repository>
```



Commit message

Update pom.xml

Extended description

Add an optional extended description..

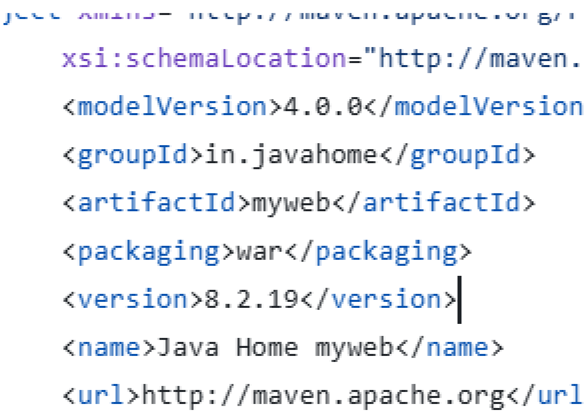
● Commit directly to the `master` branch

○ Create a **new branch** for this commit and start a pull request Learn more about pull requests

-

IMP Note –

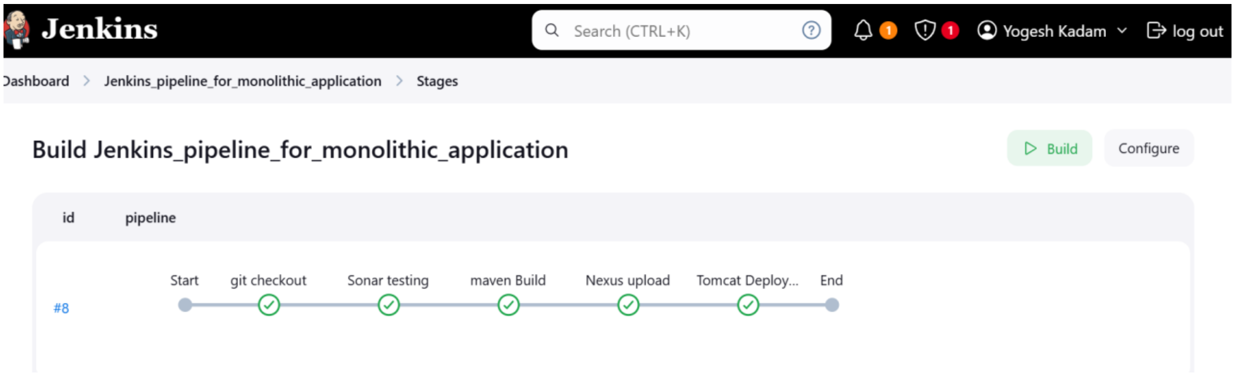
The version in pom.xml file in GitHub and in Jenkins groovy script should be same.



```
ject xmlns="http://maven.apache.org/
    xsi:schemaLocation="http://maven.
    <modelVersion>4.0.0</modelVersion
    <groupId>in.javahome</groupId>
    <artifactId>myweb</artifactId>
    <packaging>war</packaging>
    <version>8.2.19</version>
    <name>Java Home myweb</name>
    <url>http://maven.apache.org</url
```

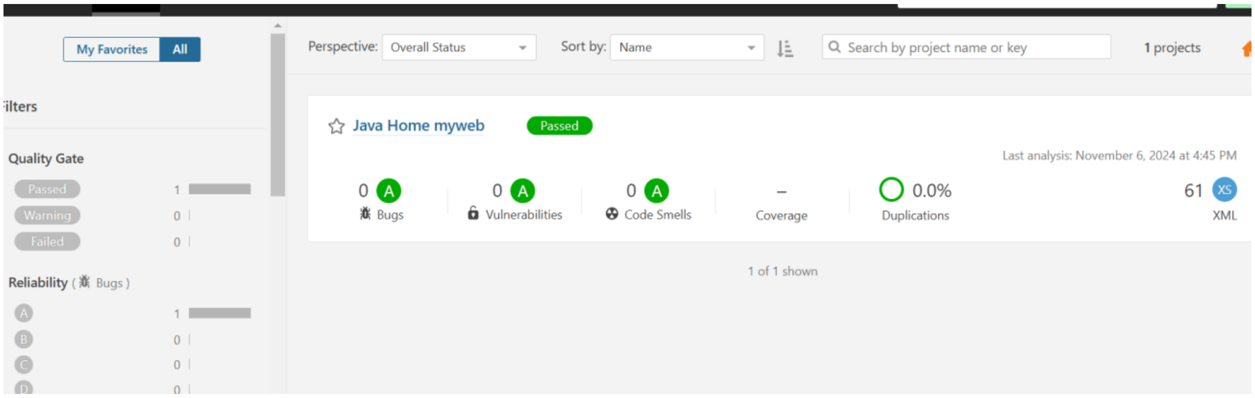
```
        nexusArtifactUploader artifacts: [[artifactId: 'myweb', classifier: '', file: 'target/myweb-8.2.19.war', type: 'war']], credentialsId: 'nexus3', groupId: 'in.javahome', nexusUrl: '10.10.1.19:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'maven-releases', version: '8.2.19'

          }

        }
```

- Make Java 11 default
- **Build Now**


Jenkins
Search (CTRL+K)
Yogesh Kadam
log out
Dashboard > Jenkins_pipeline_for_monolithic_application > Stages
Build
Configure

Build Jenkins_pipeline_for_monolithic_application

id	pipeline
#8	Start → git checkout ✓ → Sonar testing ✓ → maven Build ✓ → Nexus upload ✓ → Tomcat Deploy... ✓ → End

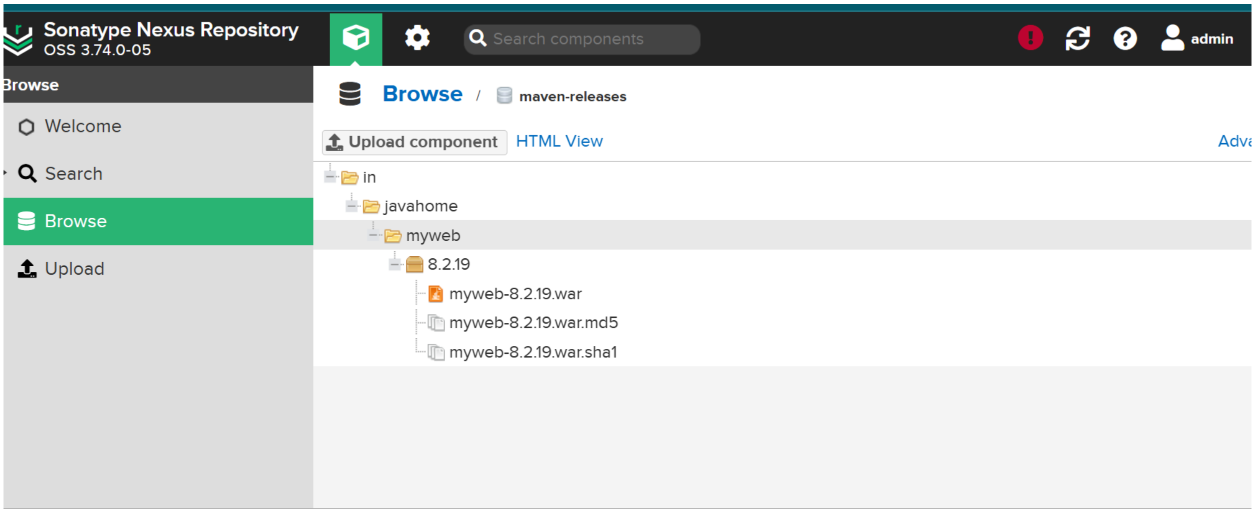


Java Home myweb — Passed

Last analysis: November 6, 2024 at 4:45 PM

Bugs	Vulnerabilities	Code Smells	Coverage	Duplications	XML
0 A	0 A	0 A	–	0.0%	61 XS

1 of 1 shown

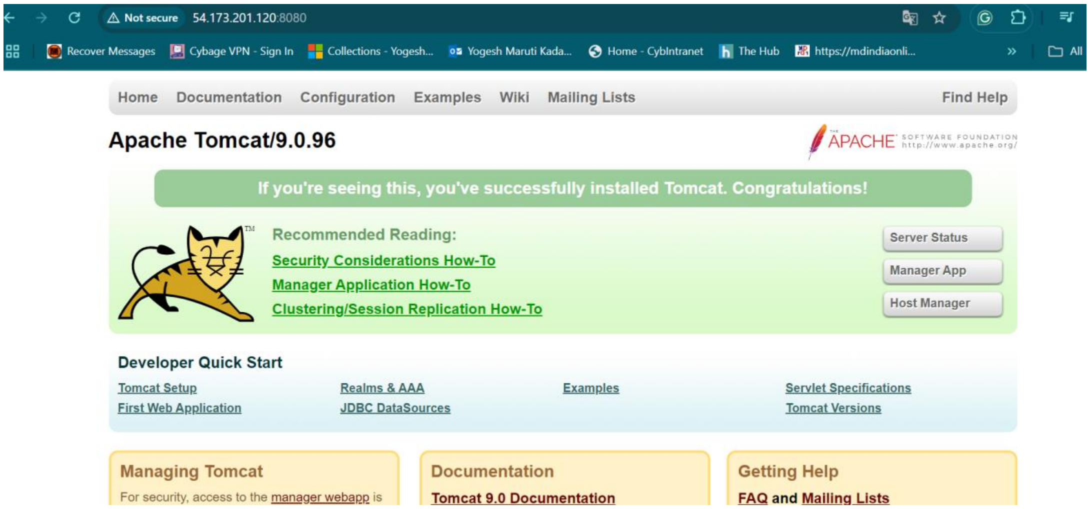


Browse / maven-releases

```
in
  javahome
    myweb
      8.2.19
        myweb-8.2.19.war
        myweb-8.2.19.war.md5
        myweb-8.2.19.war.sha1
```

To check,

- Copy public ip of tomcat server
- **Public_ip:8080** on browser



Apache Tomcat/9.0.96

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

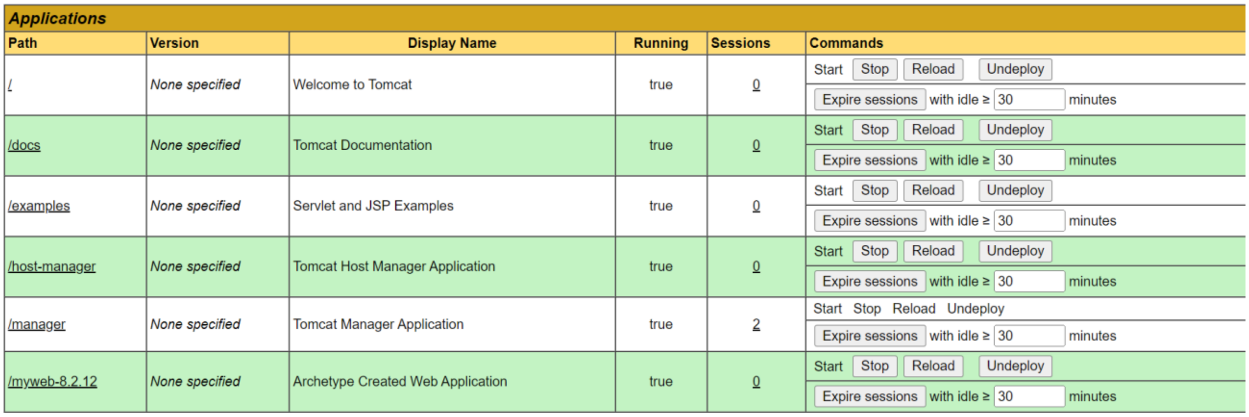
Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat
For security, access to the manager webapp is

Documentation
Tomcat 9.0 Documentation

Getting Help
FAQ and Mailing Lists

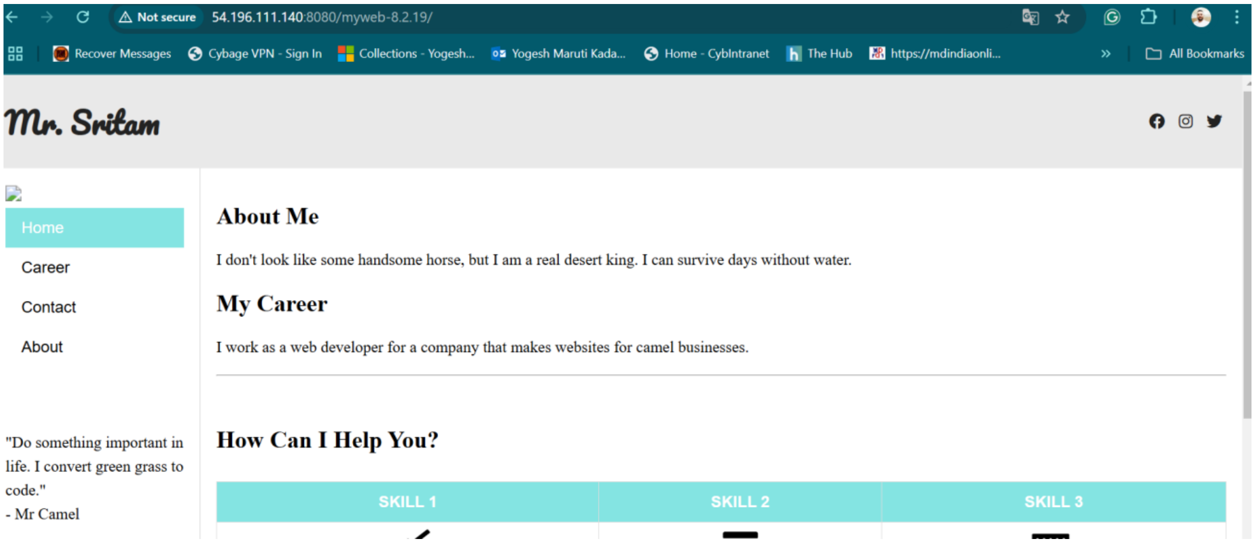
- Click on Manager App



Applications

Path	Version	Display Name	Running	Sessions	Commands
/	*None specified*	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	*None specified*	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	*None specified*	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	*None specified*	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	*None specified*	Tomcat Manager Application	true	2	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/myweb-8.2.12	*None specified*	Archetype Created Web Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Once we click on myweb-8.2.19, Application will display like this



About Me

I don't look like some handsome horse, but I am a real desert king. I can survive days without water.

My Career

I work as a web developer for a company that makes websites for camel businesses.

How Can I Help You?

"Do something important in life. I convert green grass to code."
- Mr Camel

To Run Pipeline on Other than Master Branch:

- Open Pipeline Syntax
- Repository url
- Branch
- Select branch
- Copy this Syntax in groovy script