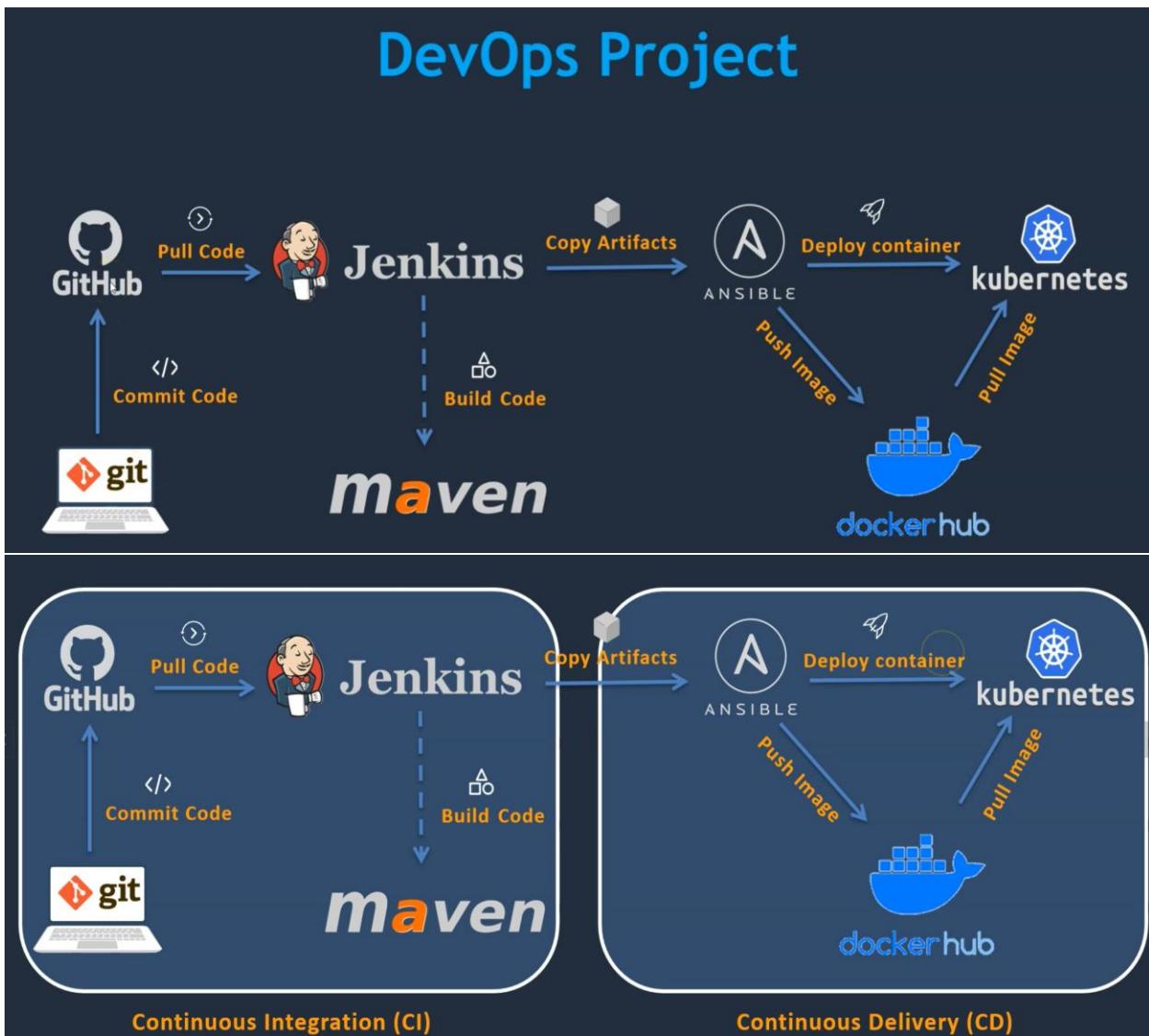


# DevOps Project



Github URL for this project use

<https://github.com/rajivsiddiqui/hello-world>  
<https://github.com/rajivsiddiqui/Simple-DevOps-Project>

**For Jenkins ec2 make sure use at least t2 medium**

Steps need to do.

1. Create a git hub account.
2. Install git bash in our local pc
3. Create a VM or Ec2 for install Jenkins
  - a. First install java

- b. second install Jenkins
- c. install nagrok (to get the local ip in online)
- 4. Integrate git with Jenkins.
- 5. Pull the code from github in jenkins server
- 6. Integrate Maven with Jenkins
  - a. Setup Maven on Jenkins server
  - b. Setup Environment Variables
  - c. JAVA\_HOME, M2, 7M2\_HOME
  - d. Install Maven Plugin
  - e. Configure Maven and Java
- 7. Install tomcat server
- 8. Integrate tomcat server with Jenkins
- 9. Install git bash in your local pc
- 10. Clone the project from github
- 11. Deploy a VM and install docker
- 12. Integrate docker with Jenkins
- 13.

### Phase 1: Build and Deploy on Tomcat Server

- Setup CI/CD with GitHub, Jenkins, Maven and Tomcat
- Setup Jenkins
  - Setup & configure Maven and Git
  - Setup Tomcat Server
  - Integrating GitHub, Maven, Tomcat Server with Jenkins
  - Create a CI and CD job
  - Test the deployment



### Phase 2: Deploy Artifact on a container:

➤ Setup CI/CD with GitHub, Jenkins, Maven and Docker

- Setting up Docker environment
- Write Dockerfile
- Create an image and container on docker host
- Integrate docker host with Jenkins
- Create CI/CD job on Jenkins to build and deploy on a container



Phase 3: Deploy Artifact on a container by using maven:

➤ CI/CD with GitHub, Jenkins, Maven, Ansible and Docker

- Setup Ansible server
- Integrate Docker host with Ansible
- Ansible playbook to create image
- Ansible playbook to create container
- Integrate Ansible with Jenkins
- CI/CD job to build code on ansible and deploy it on docker container



Phase 4: Deploy Artifact on Kubernetes:

## ➤ CI/CD with GitHub, Jenkins, Maven, Ansible and Kubernets

- Setup Kubernetes (EKS)
- Write pod, service and deployment manifest files
- Integrate Kubernetes with Ansible
- Ansible playbooks to create deployment and service
- CI/CD job to build code on ansible and deploy it on Kubernetes



How CICD work:

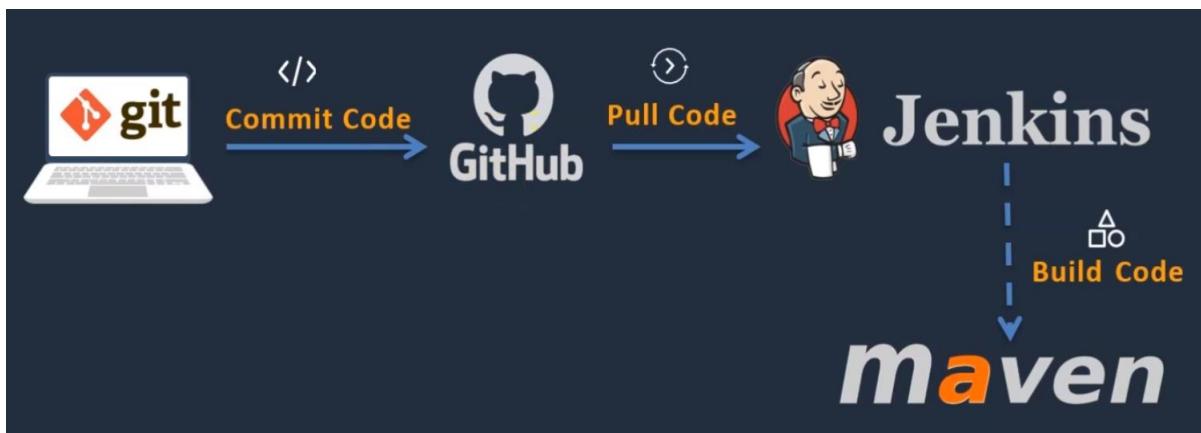


URL: <https://aws.amazon.com/devops/continuous-integration/>

### Steps before start the project

1. Install git bash
2. Clone the source code
3. Install mobaxterm

Now build the code



We have already git and maven installed

Now we are going to setup Jenkins

#### Setup Jenkins server

- Setup a Linux EC2 Instance
- Install Java
- Install Jenkins
- Start Jenkins
- Access Web UI on port 8080



# Jenkins

#### Jenkins default password location

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

The Jenkins default username is Admin

now create a Jenkins first jobs

Dashboard >

[+ New Item](#)

People

Build History

Manage Jenkins

My Views

**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**

**Build Queue** ▾  
No builds in the queue.

**Create a job** →

**Build Executor Status** ▾  
1 Idle  
2 Idle

**Set up a distributed build**

**Set up an agent** →

**Configure a cloud** →

**Learn more about distributed builds** ↗

Dashboard >

**Enter an item name**

» Required field

**Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**OK**

# Configure

 Poll SCM ?

 General

 Source Code Management

 Build Triggers

 Build Environment

 Build Steps

 Post-build Actions

## Build Environment

- Delete workspace before build starts
- Use secret text(s) or file(s) ?
- Add timestamps to the Console Output
- Inspect build log for published build scans
- Terminate a build if it's stuck
- With Ant ?

## Build Steps

Add build step ▾

Filter

- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit

# Configure

 Poll SCM [?](#)

 General

 Source Code Management

 Build Triggers

 Build Environment

 Build Steps

 Post-build Actions

## Build Environment

Delete workspace before build starts

Use secret text(s) or file(s) [?](#)

Add timestamps to the Console Output

Inspect build log for published build scans

Terminate a build if it's stuck

With Ant [?](#)

## Build Steps

### Execute shell [?](#)

Command

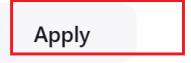
See [the list of available environment variables](#)

```
echo "rajiv"  
uptime
```

1

3

 Save

 Apply

2

[Status](#)[Changes](#)[Workspace](#)[Build Now](#) **1**[Configure](#)[Delete Project](#)[Rename](#)

## Project HellowWorldJob

hellow world job

### Permalinks

[Build History](#)

trend ▾

 Filter builds... /[#1](#)**2**

| Aug 18, 2023, 12:57 PM

[Atom feed for all](#) [Atom feed for failures](#)[Status](#)[Changes](#)[Console Output](#)[View as plain text](#)[Edit Build Information](#)[Delete build '#1'](#)

## Console Output

```

Started by user rajiv siddiqui
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/HellowWorldJob
[HellowWorldJob] $ /bin/sh -xe /tmp/jenkins2415057380822631631.sh
+ echo rajiv
rajiv
+ uptime
12:57:48 up 32 min, 0 users, load average: 0.59, 0.57, 0.50
Finished: SUCCESS

```

## Integrate git with Jenkins

1. Install git on jenkins
2. Install GitHub plugin on jenkins
3. Configure git on Jenkins GUI

### **Install git on jenkins serevr**

Check git is installed or not

```
#git --version
```

if not then installed

```
#yum install git      =install git on Jenkins server  
#git --version       =see the version of git
```

### Chage the hostname

```
#vi /etc/hostname
```

give the name Jenkins

```
sudo - su
```

```
#init 6  = to reboot the server
```

## Install GitHub plugin on jenkins

The screenshot shows the Jenkins dashboard. At the top left, there's a 'Dashboard' button followed by a right-pointing arrow. Below it is a search bar with a magnifying glass icon. To the right of the search bar are three buttons: 'New Item' (with a plus sign), 'Add description' (with a pencil icon), and 'Manage Jenkins' (which is highlighted with a red border). Further down are links for 'People', 'Build History', and 'My Views'. The main area features a table titled 'Build Queue' with one entry: 'helloworldJob' (Status: S, Last Success: 10 min ago, Last Failure: N/A, Last Duration: 0.13 sec). Below this table are sections for 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (listing two idle executors: '1 Idle' and '2 Idle'). At the bottom right of the dashboard, there are links for 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'.

1

Available plugins

2

github

3

GitHub 1.37.3

External Site/Tool Integrations github

This plugin integrates GitHub to Jenkins.

4

Install without restart

Download now and install after restart

Released

Install	Name	Released
<input type="checkbox"/>	GitHub API 1.314-431.v78d72a_3fe4c3 github Library plugins (for use by other plugins)	3 mo 2 days ago
<input checked="" type="checkbox"/>	GitHub 1.37.3 External Site/Tool Integrations github This plugin integrates GitHub to Jenkins.	3 days 17 hr ago
<input type="checkbox"/>	GitHub Branch Source 1732.v3f1889a_c475b_ pipeline github Source Code Management Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc.	8 days 0 hr ago
<input type="checkbox"/>	Caffeine API 3.1.8-133.v17b_1ff2e0599 Library plugins (for use by other plugins) Caffeine api plugin for use by other Jenkins plugins.	6 hr 2 min ago
<input type="checkbox"/>	OkHttp 4.11.0-157.v6852a_a_fa_ec11 This plugin provides OkHttp for other plugins	7 days 5 hr ago

Update information obtained: 1 hr 49 min ago Check now

Trilead API	 Success
SSH Credentials	 Success
Credentials Binding	 Success
Pipeline: SCM Step	 Success
Mina SSHD API :: Common	 Success
Mina SSHD API :: Core	 Success
Apache HttpComponents Client 4.x API	 Success
Caffeine API	 Success
Script Security	 Success
Git client	 Success
SCM API	 Success
Jakarta Activation API	 Success
Jakarta Mail API	 Success
Display URL API	 Success
Mailer	 Success
Git	 Success
Token Macro	 Success
GitHub	 Success
Loading plugin extensions	 Success

→ [Go back to the top page](#)

(you can start using the installed plugins right away)

→  Restart Jenkins when installation is complete and no jobs are running

## Add git

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links: '+ New Item', 'People', 'Build History', 'Manage Jenkins' (which is highlighted with a red box), and 'My Views'. Below the sidebar are two expandable sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). At the top right, there are buttons for 'Add description' and 'All' (with '+' and a search icon). To the right of the sidebar is a table showing build status for 'helloworldjob': Last Success was 19 min ago, Last Failure is N/A, and Last Duration is 0.13 sec. There are also links for 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'.

The screenshot shows the 'Manage Jenkins' configuration page. The left sidebar includes '+ New Item', 'People', 'Build History', 'Manage Jenkins' (highlighted with a red box), and 'My Views'. Below the sidebar are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). The main content area has a heading 'Manage Jenkins'. It features a warning message: 'Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#)'. Below this are two buttons: 'Set up agent' and 'Set up cloud', with a 'Dismiss' button. A note states: 'Jenkins URL is empty but is required for the proper operation of many Jenkins features like email notifications, PR status update, and environment variables such as BUILD\_URL.' Another note says: 'Please provide an accurate value in [Jenkins configuration](#)'. The 'System Configuration' section contains four items: 'System' (Configure global settings and paths), 'Tools' (Configure tools, their locations and automatic installers, highlighted with a red box), 'Nodes and Clouds' (Add, remove, control and monitor the various nodes that Jenkins runs jobs on), and 'Plugins' (Add, remove, disable or enable plugins that can extend the functionality of Jenkins).

JDK

JDK installations

List of JDK installations on this system

[Add JDK](#)

---

Git installations

Git

Name **1**

Path to Git executable **2**

Install automatically **?**

[Add Git](#)

**4** [Save](#) [Apply](#) **3**

If that not work then we can write the full path of git by using following command

```
rajiv@jenkins-server:~$ whereis git
git: /usr/bin/git /usr/share/man/man1/git.1.gz
```

## Pull code from github

Dashboard > Manage Jenkins

[+ New Item](#)

[People](#)

[Build History](#)

**Manage Jenkins**

Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#).

[Set up agent](#) [Set up cloud](#) [Dismiss](#)

Jenkins URL is empty but is required for the proper operation of many Jenkins features like email notifications, PR status update, and environment variables such as `BUILD_URL`.

Please provide an accurate value in [Jenkins configuration](#).

[Dismiss](#)

**Build Queue**

No builds in the queue.

**Build Executor Status**

1 Idle  
2 Idle

**System Configuration**

**System**  
Configure global settings and paths.

**Tools**  
Configure tools, their locations and automatic installers.

**Nodes and Clouds**  
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

**Plugins**  
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

**Enter an item name**

» Required field

**Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

If you want to create a new item from other existing, you can use this option:

Copy from

**OK**

Dashboard > pullcodeFromGitHub > Configuration

**Configure** **General** Enabled

**General**

Description

[Plain text] [Preview](#)

Discard old builds ?  
 GitHub project  
 This project is parameterized ?  
 Execute concurrent builds if necessary ?

[Advanced](#)

**Source Code Management**

None

Git ? **1**

Repositories ?

Repository URL ? **2**

Credentials ? **3**

- none -

Add ▾

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ? **4**

Add Branch **5**

Save **5**

Apply **4**

This screenshot shows a configuration interface for a Git repository. At the top left is a 'Git' icon with a question mark. Below it is a 'Repositories' section with a 'Repository URL' field containing 'https://github.com/rajivsiddiqui/hello-world.git'. A red box labeled '1' highlights the 'Git' icon. A red box labeled '2' highlights the 'Repository URL' field. In the 'Credentials' section, there's a dropdown menu set to '- none -' with a red box labeled '3' highlighting it. Below this is an 'Add' button with a dropdown arrow. An 'Advanced' button with a dropdown arrow is also present. An 'Add Repository' button is located below the 'Repositories' section. The next section is 'Branches to build' with a 'Branch Specifier' field containing '\*/master'. A red box labeled '4' highlights the 'Branch Specifier' field. Below it is an 'Add Branch' button with a red box labeled '5' highlighting it. At the bottom are two buttons: a blue 'Save' button and a grey 'Apply' button.

Status      **Project pullcodeFromGitHub**

</> Changes      pull code from git hub

Workspace

**Build Now**      1

Configure

Delete Project

Rename

**Build History**      trend ▾

Filter builds... /

#1      Aug 15, 2023, 4:41 PM

Atom feed for all Atom feed for failures

Status      **Console Output**

</> Changes

Console Output      here the code is downloaded we can get the from the location

View as plain text

Edit Build Information

Delete build '#1'

Git Build Data

```

Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/pullcodeFromGitHub
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
  > git init /var/lib/jenkins/workspace/pullcodeFromGitHub # timeout=10
Fetching upstream changes from https://github.com/rajivsiddiqui/hello-world.git
  > git --version # timeout=10
  > git --version # 'git' version 2.40.1*
  > git fetch --tags --force --progress -- https://github.com/rajivsiddiqui/hello-world.git +refs/heads/*:refs/remotes/origin/* # timeout=10
  > git config remote.origin.url https://github.com/rajivsiddiqui/hello-world.git # timeout=10
  > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
  > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 1b447195e49839e421060e33ed99058bfe60fc98 (refs/remotes/origin/master)
  > git config core.sparsecheckout # timeout=10
  > git checkout -f 1b447195e49839e421060e33ed99058bfe60fc98 # timeout=10
Commit message: "Update index.jsp"
First time build. Skipping changelog.
Finished: SUCCESS

```

```

root@ip-172-31-32-75:/var/lib/jenkins/workspace
root@ip-172-31-32-75 ec2-user]# whereis git
git: /usr/bin/git /usr/share/man/man1/git.1.gz
root@ip-172-31-32-75 ec2-user]# cd /var/lib/jenkins/workspace
root@ip-172-31-32-75 workspace]# ll
total 0
rwxr-xr-x 2 jenkins jenkins 6 Aug 15 15:28 helloworldJob
rwxr-xr-x 5 jenkins jenkins 147 Aug 15 16:41 pullcodeFromGitHub
root@ip-172-31-32-75 workspace]#

```

## (6)Integrate Maven with Jenkins

- a) Setup Maven on Jenkins server
- b) Setup Environment Variables
  - a. JAVA\_HOME,M2,M2\_HOME
- c) Install Maven Plugin
- d) Configure Maven and Java

### Maven download URL

<https://maven.apache.org/>

The screenshot shows the Apache Maven Project website at <https://maven.apache.org/>. The main navigation bar includes links for Welcome, License, ABOUT MAVEN, What is Maven?, Features, Download (which is highlighted), Use, Release Notes, DOCUMENTATION, Maven Plugins, Maven Extensions, Index (category), User Centre, Plugin Developer Centre, Maven Repository Centre, Maven Developer Centre, Books and Resources, Security, COMMUNITY, Community Overview, Project Roles, How to Contribute, and Getting Help.

The central content area is titled "Downloading Apache Maven 3.9.6". It states that Apache Maven 3.9.6 is the latest release and recommends it for all users. Below this is a "System Requirements" section with the following table:

Requirement	Description
Java Development Kit (JDK)	Maven 3.9+ requires JDK 8 or above to execute. It still allows you to build against 1.3 and other JDK versions by using toolchains.
Memory	No minimum requirement
Disk	Approximately 10MB is required for the Maven installation itself. In addition to that, disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 500MB.
Operating System	No minimum requirement. Start up scripts are included as shell scripts (tested on many Unix flavors) and Windows batch files.

The "Files" section provides links to various Maven distributions and their checksums and signatures:

Format	Link	Checksums	Signature
Binary tar.gz archive	<a href="#">apache-maven-3.9.6-bin.tar.gz</a>	apache-maven-3.9.6-bin.tar.gz.sha512	apache-maven-3.9.6-bin.tar.gz.asc
Binary zip archive	<a href="#">apache-maven-3.9.6-bin.zip</a>	apache-maven-3.9.6-bin.zip.sha512	apache-maven-3.9.6-bin.zip.asc
Source tar.gz archive	<a href="#">apache-maven-3.9.6-src.tar.gz</a>	apache-maven-3.9.6-src.tar.gz.sha512	apache-maven-3.9.6-src.tar.gz.asc
Source zip archive	<a href="#">apache-maven-3.9.6-src.zip</a>	apache-maven-3.9.6-src.zip.sha512	apache-maven-3.9.6-src.zip.asc

### a)Download and install maven in Jenkins server

```

#sudo su
#cd /opt
#wget https://dlcdn.apache.org/maven/maven-3/3.9.4/binaries/apache-maven-3.9.4-bin.tar.gz

```

```
#tar -xvzf apache-maven-3.9.4-bin.tar.gz  
#ll  
#mv apache-maven-3.9.4 maven  
#ll  
#cd maven/  
#ll  
#cd bin/  
#./mvn -v
```

### **b)Set home path in Jenkins server**

```
cd /root  
vi .bash_profile = if its ec2 instance  
or  
vi .profile =if its ubuntu os  
now write the following line in this file
```

```
M2_HOME='/opt/maven'  
M2='/opt/maven/bin'  
  
JAVA_HOME='/usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.amzn2.0.1.x86_64'  
or  
JAVA_HOME='/usr/lib/jvm/java-11-openjdk-amd64'  
  
PATH="$PATH:$HOME/bin:$JAVA_HOME:$M2_HOME:$M2"  
  
export PATH
```

Right one-----

```
M2_HOME='/opt/maven'  
M2='/opt/maven/bin'  
JAVA_HOME='/usr/lib/jvm/java-11-openjdk-amd64'  
PATH="$PATH:$HOME/bin:$JAVA_HOME:$M2_HOME:$M2"  
export PATH
```

```
#echo $PATH  
#source .bash_profile if aws ec2  
or  
#source.profile if vm ubuntu  
  
#mvn -v
```

```

# ~/.profile: executed by Bourne-compatible login shells.

if [ "$BASH" ]; then
  if [ -f ~/.bashrc ]; then
    . ~/.bashrc
  fi
fi

mesg n 2> /dev/null || true

M2_HOME='/opt/maven'
M2='/opt/maven/bin'

JAVA_HOME='/usr/lib/jvm/java-17-openjdk-amd64'

PATH=$PATH:$HOME/bin:$JAVA_HOME:$M2_HOME:$M2"

export PATH

```

```

M2_HOME='/opt/maven'
M2='/opt/maven/bin'

JAVA_HOME='/usr/lib/jvm/java-17-openjdk-amd64'

PATH=$PATH:$HOME/bin:$JAVA_HOME:$M2_HOME:$M2"

export PATH

```

**For getting the java home path we can use the following way**

```

#find / -name jvm
#cd /usr/lib/jvm
#find / -name java-11*

JAVA_HOME when use local vm /usr/lib/jvm/java-11-openjdk-amd64
JAVA_HOME when use aws ec2 /usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.amzn2.0.1.x86_64

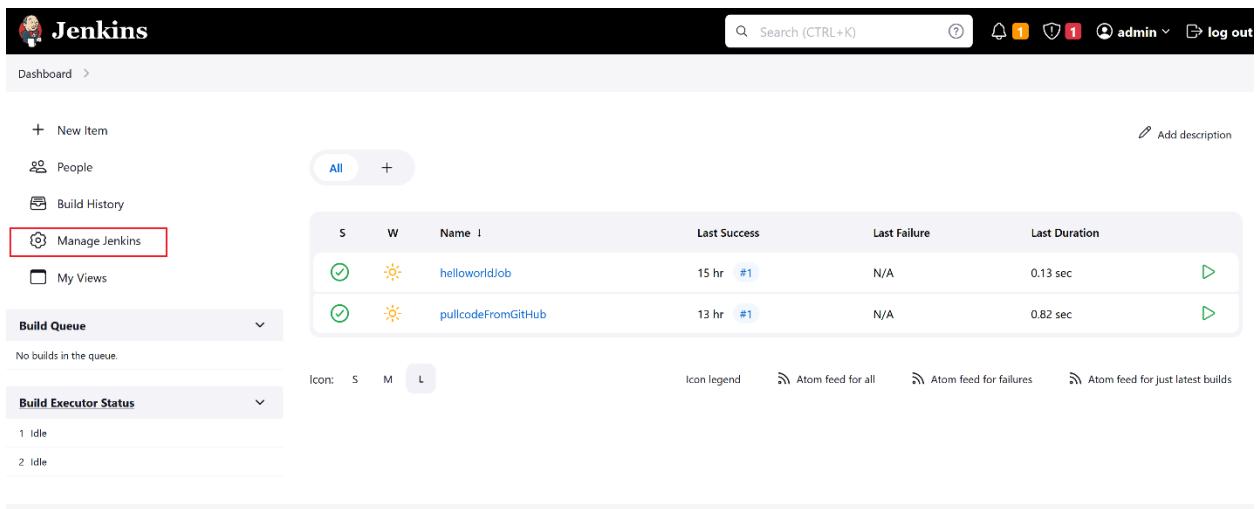
```

```

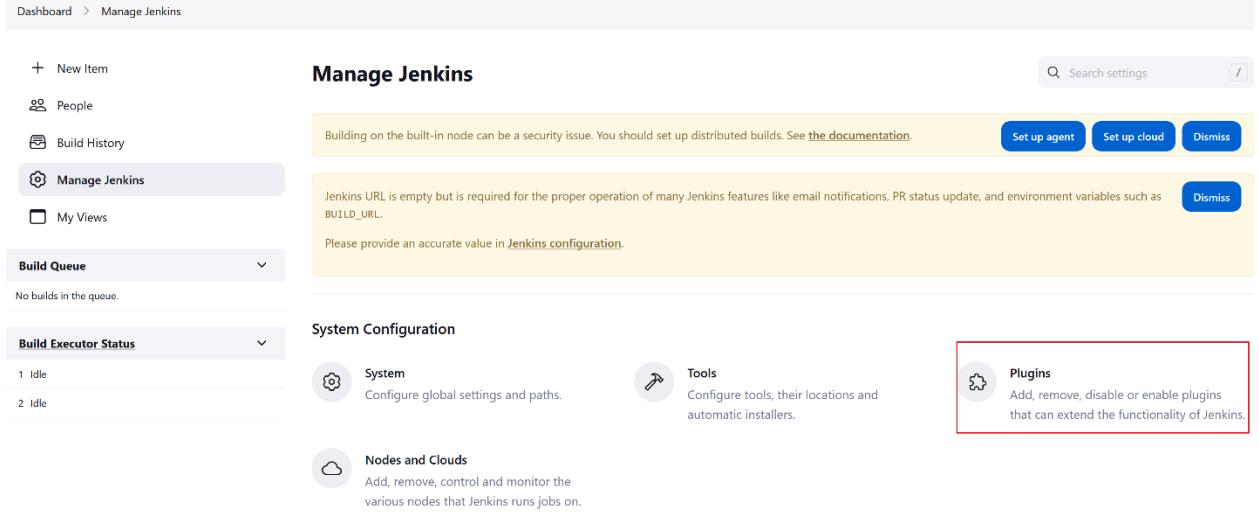
root@ubuntu-light: /usr/share/gdb/auto-load/usr/lib/jvm
root@ubuntu-light:/usr/lib/jvm/java-11-openjdk-amd64# cd /root/
root@ubuntu-light:~# pwd
/root
root@ubuntu-light:~# find / -name jvm
/usr/share/gdb/auto-load/usr/lib/jvm
/usr/lib/debug/usr/lib/jvm
/usr/lib/jvm
root@ubuntu-light:~# cd /usr/share/gdb/auto-load/usr/lib/jvm
root@ubuntu-light:/usr/share/gdb/auto-load/usr/lib/jvm# ll
total 16
lrwxr-xr-x 4 root root 4096 Aug 18 13:27 .
lrwxr-xr-x 4 root root 4096 Aug 18 13:26 ..
lrwxr-xr-x 3 root root 4096 Aug 18 13:27 java-11-openjdk-amd64/
lrwxr-xr-x 3 root root 4096 Aug 18 13:26 java-17-openjdk-amd64/
root@ubuntu-light:/usr/share/gdb/auto-load/usr/lib/jvm# find / -name java-11*
/usr/share/gdb/auto-load/usr/lib/jvm/java-11-openjdk-amd64
/usr/lib/debug/usr/lib/jvm/java-11-openjdk-amd64
/usr/lib/jvm/java-11-openjdk-amd64 this path we need
root@ubuntu-light:/usr/share/gdb/auto-load/usr/lib/jvm#

```

### c) Now Install Maven Plugin in jenkins



The screenshot shows the Jenkins dashboard. On the left sidebar, the 'Manage Jenkins' link is highlighted with a red box. The main area displays a table of build items: 'helloworldjob' (last success 15 hr ago, #1, N/A duration) and 'pullcodeFromGitHub' (last success 13 hr ago, #1, N/A duration). Below the table, there are sections for 'Build Queue' (empty), 'Build Executor Status' (1 Idle, 2 Idle), and 'Atom feeds'.

The screenshot shows the 'Manage Jenkins' configuration page. The 'System Configuration' section includes links for 'System', 'Tools', 'Nodes and Clouds', and 'Plugins'. The 'Plugins' section is highlighted with a red box and contains the text: 'Add, remove, disable or enable plugins that can extend the functionality of Jenkins.' It also includes a 'Set up agent' button.

Dashboard > Manage Jenkins > Plugins

## Plugins

Available plugins maven

Installed plugins Advanced settings Download progress

Install	Name	Released
<input checked="" type="checkbox"/>	Maven Integration 3.23	11 days ago
<input type="checkbox"/>	Config File Provider 952.va_544a_6234b_46	24 days ago
<input type="checkbox"/>	Jira 3.10	2 mo 16 days ago
<input type="checkbox"/>	Pipeline Maven Integration 1322.v9ef317a_3e0a_9	7 days 9 hr ago

[Install without restart](#) [Download now and install after restart](#) Update information obtained: 16 hr ago Check now

Dashboard > Manage Jenkins > Plugins

## Download progress

Available plugins  Installed plugins  Advanced settings  Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

OkHttp	<input checked="" type="checkbox"/> Success
JavaBeans Activation Framework (JAF) API	<input checked="" type="checkbox"/> Success
JAXB	<input checked="" type="checkbox"/> Success
SnakeYAML API	<input checked="" type="checkbox"/> Success
Jackson 2 API	<input checked="" type="checkbox"/> Success
commons-lang3 v3.x Jenkins API	<input checked="" type="checkbox"/> Success
GitHub API	<input checked="" type="checkbox"/> Success
bouncycastle API	<input checked="" type="checkbox"/> Success
Instance Identity	<input checked="" type="checkbox"/> Success

Dashboard >

+ New item  Add description

People All +

Build History

Manage Jenkins

My Views

**Build Queue** ▼  
No builds in the queue.

S	W	Name	Last Success	Last Failure	Last Duration
<input checked="" type="checkbox"/>		helloworldjob	15 hr #1	N/A	0.13 sec <span style="float: right;">▶</span>
<input checked="" type="checkbox"/>		pulcodeFromGitHub	14 hr #1	N/A	0.82 sec <span style="float: right;">▶</span>

Icon: S M L

Icon legend Atom feed for all Atom feed for failures Atom feed for just latest builds

**Build Executor Status** ▼

1 Idle  
2 Idle

Dashboard > Manage Jenkins

## Manage Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#).

Jenkins URL is empty but is required for the proper operation of many Jenkins features like email notifications, PR status update, and environment variables such as `BUILD_URL`.  
Please provide an accurate value in [Jenkins configuration](#).

**System Configuration**

- System**: Configure global settings and paths.
- Tools**: Configure tools, their locations and automatic installers. **Tools** is highlighted with a red box.
- Nodes and Clouds**: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Plugins**: Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

### JDK

JDK installations

List of JDK installations on this system

**Add JDK** **1**

**JDK**

Name
java-11 <b>2</b>

**JAVA\_HOME**

/usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.amzn2.0.1.x86_64 <b>3</b>
--

If we use vm on ubuntu then java path will be **3 /usr/lib/jvm/java-11-openjdk-amd64**

**1** /usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.amzn2.0.1.x86\_64 doesn't look like a JDK directory

Install automatically ?

**Add JDK**

**If we use vm on ubuntu then java path will be**

/usr/lib/jvm/java-11-openjdk-amd64

## Maven

### Maven installations

List of Maven installations on this system

[Add Maven](#)

**Maven**

Name  X

[Install automatically](#) ? uncheck this one we do it manually

**Install from Apache**

Version  ▼

[Add Installer](#) ▼

[Add Maven](#)

## Maven

### Maven installations

List of Maven installations on this system

[Add Maven](#) 1

**Maven**

Name  2

MAVEN\_HOME  3

[Install automatically](#) ?

[Add Maven](#)

5 Save Apply 4

[Create a Maven project](#)

Dashboard > Manage Jenkins

+ New Item

Manage Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#).

Set up agent Set up cloud Dismiss

Jenkins URL is empty but is required for the proper operation of many Jenkins features like email notifications, PR status update, and environment variables such as `BUILD_URL`.

Please provide an accurate value in [Jenkins configuration](#).

Dismiss

Build Queue

No builds in the queue.

Build Executor Status

1 Idle  
2 Idle

System Configuration

System Configure global settings and paths.

Tools Configure tools, their locations and automatic installers.

Nodes and Clouds Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Plugins Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Dashboard > All >

Enter an item name

FirstMavenProject 1

» Required field

**Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration. 2

If you want to create a new item from other existing, you can use this option:

Copy from

Type to autocomplete

OK 3

**Configure**

**General**

Enabled 

**Description**

My first maven project.

[Plain text] [Preview](#)

Discard old builds 

GitHub project

This project is parameterized 

Execute concurrent builds if necessary 

**Source Code Management**

None

Git 

**Repositories** 

**Repository URL** 

https://github.com/rajivsiddiqui/hello-world.git

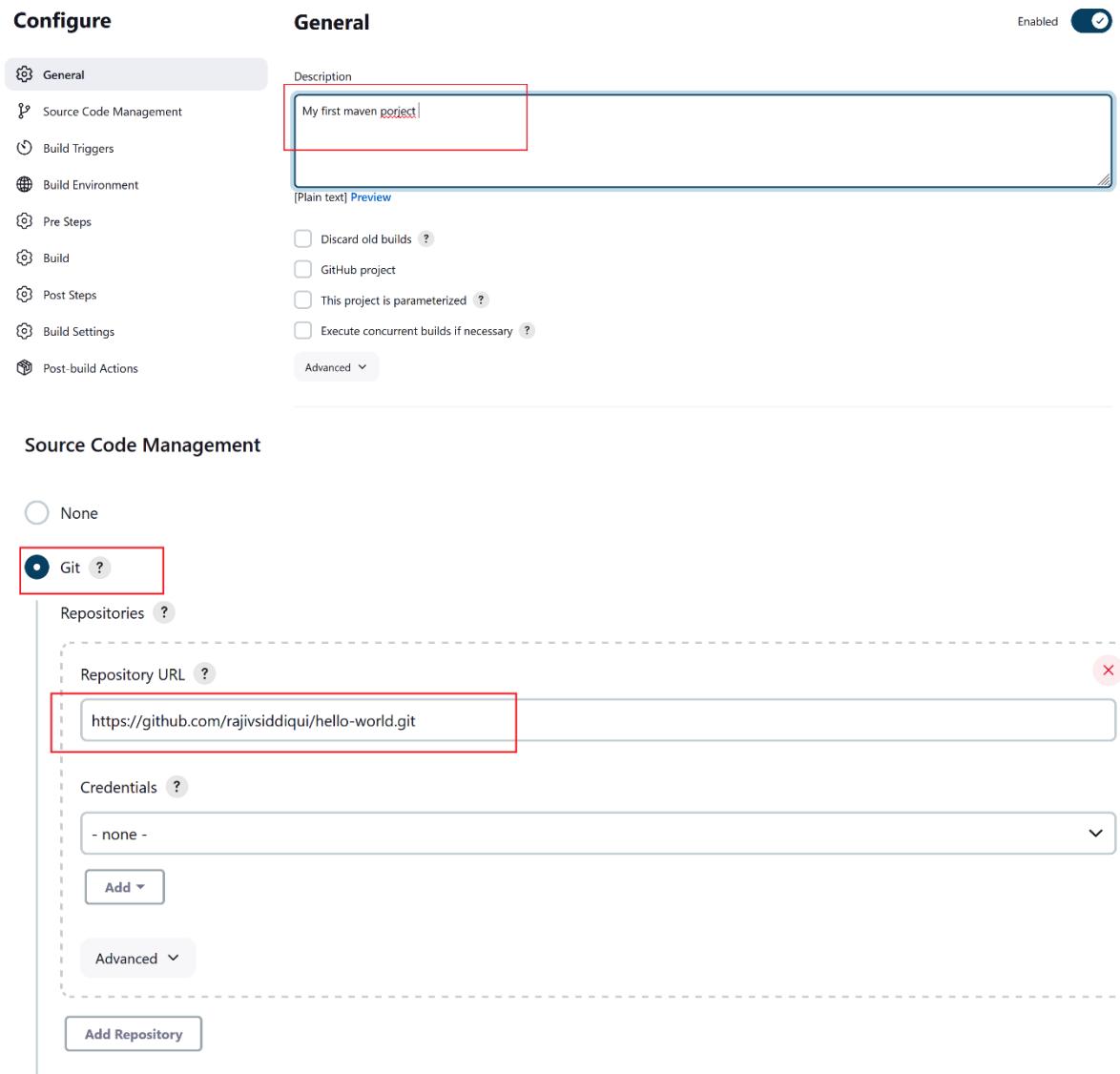
**Credentials** 

- none -

Add 

**Advanced** 

Add Repository



scroll down and add

**Configure**

**Pre Steps**

Add pre-build step ▾

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

**Build**

Root POM ?

pom.xml

Goals and options ?

clean install **1**

Advanced ▾

**Post Steps**

Run only if build succeeds

Run only if build succeeds or is unstable

**3** **Save** **2** Apply

### Maven Goals list

<https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html>

**Maven project FirstMavenProject**

Status Changes My first maven porject

Workspace

**Build Now** **1**

Configure

Delete Maven project

Modules

Rename

**Build History** trend ▾

Filter builds / it will take some time to get the build history after click the build now

#1 Aug 16, 2023, 7:06 AM **2**

Atom feed for all Atom feed for failures

**Status**  **Build #1 (Aug 16, 2023, 7:06:21 AM)**

- </> Changes
-  Console Output
-  Edit Build Information
-  Delete build '#1'
-  Git Build Data
-  Redeploy Artifacts
-  Test Result
-  See Fingerprints

**</>** No changes.

**⌚** Started by user [admin](#)

**git** **Revision:** 1b447195e49839e421060e33ed99058bfe60fc98  
**Repository:** <https://github.com/rajivsiddiqui/hello-world.git>

- refs/remotes/origin/master

**Test Result** (no failures)

### Module Builds

 Maven Project	4.5 sec
 Server	9.9 sec
 Webapp	3.5 sec

**Status**

- </> Changes
-  **Workspace** 1
-  Build Now
-  Configure
-  Delete Maven project
-  Modules
-  Rename

**location of the file**

FirstMavenProject / webapp / target / [maven-archiver](#) [surefire](#) [webapp](#) [webapp.war](#) Aug 16, 2023, 7:06:53 AM 2.31 KB  

 **this the file** 

**Build History** **trend** ▾

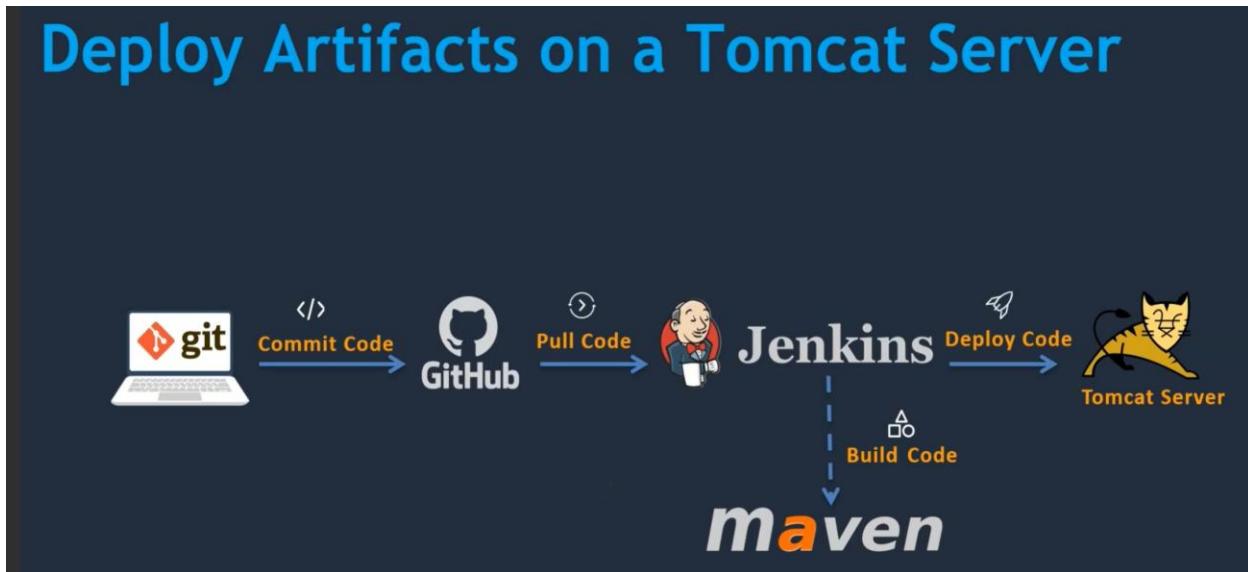
Filter builds... /

#1 Aug 16, 2023, 7:06 AM [Atom feed for all](#) [Atom feed for failures](#)

Location of the index file:

/var/lib/jenkins/workspace/FirstMavenProject/webapp/src/main/webapp

## Phase 1: Deploy artifacts on Tomcat server



- a) Setup a linux Ec2
- b) Install java
- c) Configure Tomcat
- d) Start Tomcat server
- e) Access Web UI on port 8080
- f) Integrate tomcat with jenkins

a)Setup a linux ec2-follow ec2 doc  
b) Install java from java install doc file

c)configure tomcat

**First Install Java -amazon linux-2 ios use**

**Second install tomcat**

```
#sudo su
#cd /opt
#ll
not available now #wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.79/bin/apache-tomcat-9.0.79.tar.gz

#wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.84/bin/apache-tomcat-9.0.84.tar.gz

#tar -xvzf apache-tomcat-9.0.79.tar.gz
# mv apache-tomcat-9.0.79 tomcat
##cd /opt/tomcat
#ll
```

```
#cd /opt/tomcat/bin  
# ./startup.sh
```

Now go to the browser and type your ip with 8080 port

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Apache Tomcat™ 9.0.79

Recommended Reading:

- [Security Considerations How-To](#)
- [Manager Application How-To](#)
- [Clustering/Session Replication How-To](#)

Developer Quick Start

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

Managing Tomcat

For security, access to the [manager webapp](#) is restricted. Users are defined in: `$CATALINA_HOME/conf/tomcat-users.xml`

In Tomcat 9.0 access to the manager application is split between different users. [Read more...](#)

[Release Notes](#)    [Changelog](#)    [Migration Guide](#)    [Security Notices](#)

Documentation

[Tomcat 9.0 Documentation](#)    [Tomcat 9.0 Configuration](#)    [Tomcat Wiki](#)

Find additional important configuration information in: `$CATALINA_HOME/RUNNING.txt`

Developers may be interested in:

- [Tomcat 9.0 Bug Database](#)
- [Tomcat 9.0 JavaDocs](#)
- [Tomcat 9.0 Git Repository at GitHub](#)

Getting Help

[FAQ and Mailing Lists](#)

The following mailing lists are available:

- [tomcat-announce](#)    Important announcements, releases, security vulnerability notifications. (Low volume).
- [tomcat-user](#)    User support and discussion.
- [taglibs-user](#)    User support and discussion for [Apache Taglibs](#).
- [tomcat-dev](#)    Development mailing list, including commit messages.

Now click Manage app but we are not able to access it for that we need to do the following things.

```
#cd /opt/tomcat  
# find / -name context.xml
```

[root@ip-172-31-17-65 bin]# find / -name context.xml  
/opt/tomcat/conf/context.xml  
/opt/tomcat/webapps/docs/META-INF/context.xml  
/opt/tomcat/webapps/examples/META-INF/context.xml  
/opt/tomcat/webapps/host-manager/META-INF/context.xml  
/opt/tomcat/webapps/manager/META-INF/context.xml

Need to edit this 2 file

```
# vi /opt/tomcat/webapps/host-manager/META-INF/context.xml
```

```
root@ip-172-31-46-215:/opt/tomcat
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to you under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

-->
<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|0:0:0:0:0:1" /> -->

  <Manager sessionAttributeValueClassNameFilter="java\\.lang\\.\\(?:Boolean|Integer|Long|Number|String)|org\\.apache\\.catalina\\.filters\\.CsrfPreventionFilter\\$LruCache\\?\\:\\$1\\?|java\\.util\\.\\(?:Linked)\\?HashMap\\?"/>
</Context>
```

```
# vi /opt/tomcat/webapps/manager/META-INF/context.xml
```

```
root@ip-172-31-46-215:/opt/tomcat
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to you under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

-->
<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|0:0:0:0:0:1" /> -->
  <Manager sessionAttributeValueClassNameFilter="java\\.lang\\.\\(?:Boolean|Integer|Long|Number|String)|org\\.apache\\.catalina\\.filters\\.CsrfPreventionFilter\\$LruCache\\?\\:\\$1\\?|java\\.util\\.\\(?:Linked)\\?HashMap\\?"/>
</Context>
```

```
#cd /opt/tomcat/bin
#./shutdown.sh
#./startup.sh
```

**Now stop and start tomcat and try from browser and we can get the page but now we need to configure username and password for that need to do the following things**

#### Create users for access the server

```
#cd /opt/tomcat/conf
```