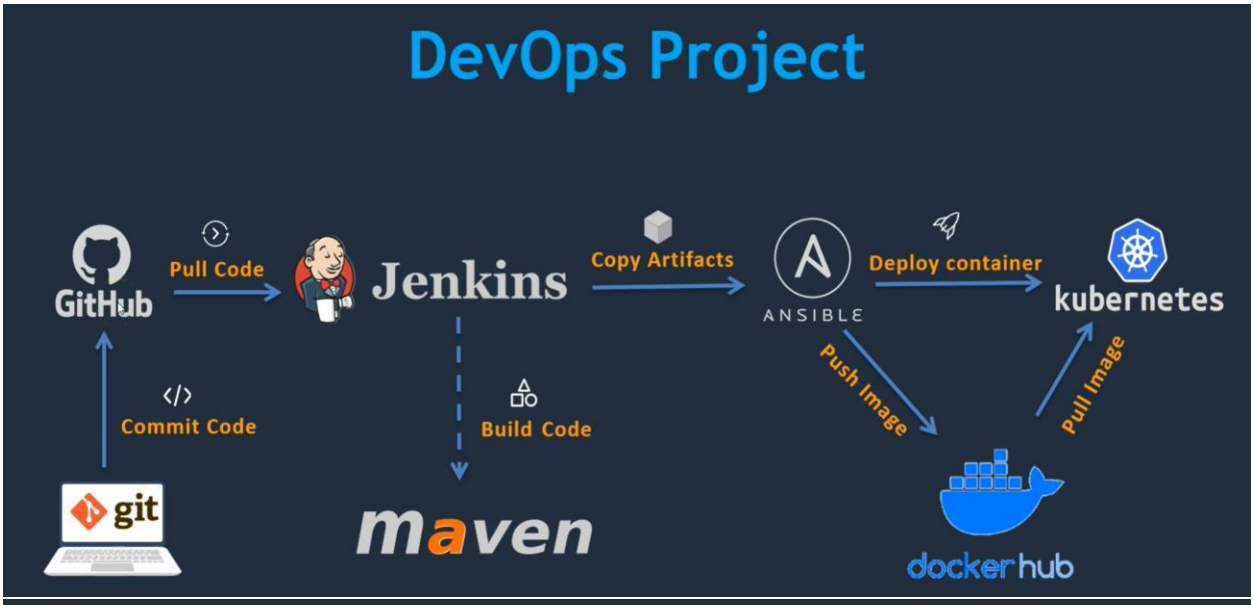
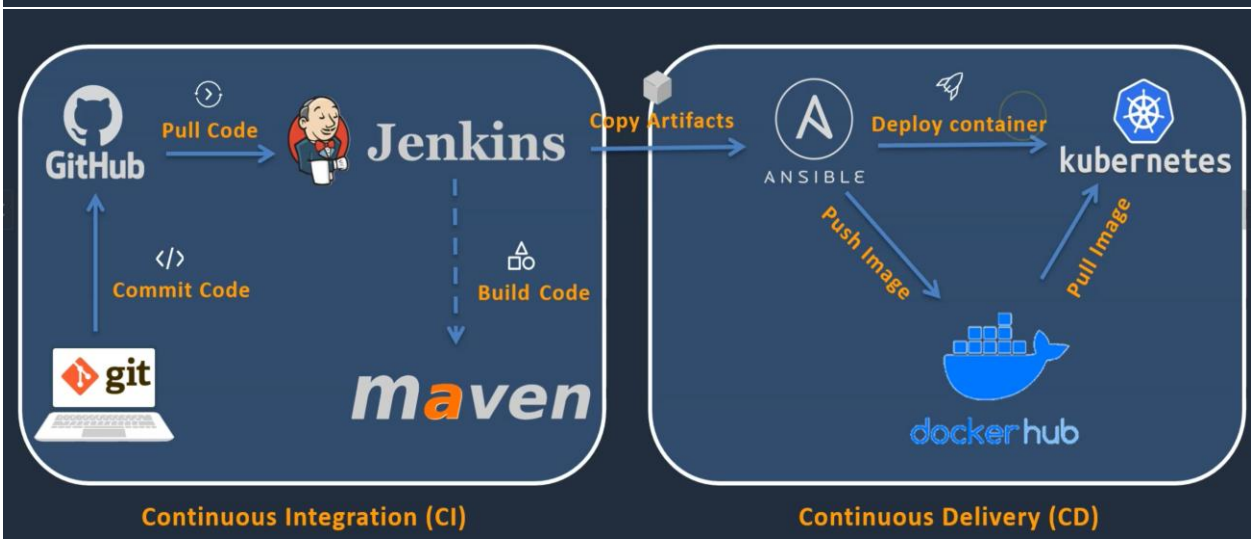




Github URL for this project use

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

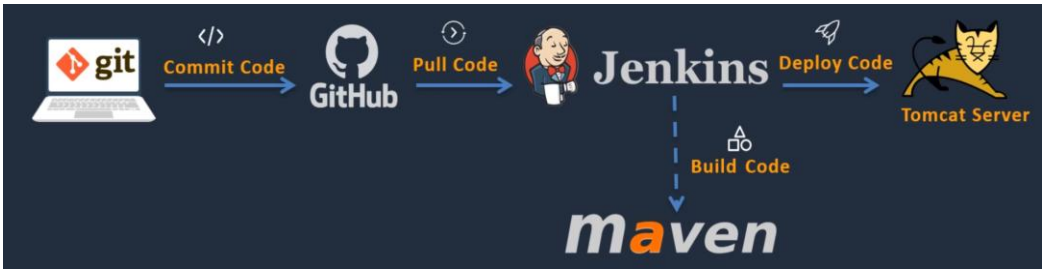
For Jenkins ec2 make sure use at least t2 medium

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

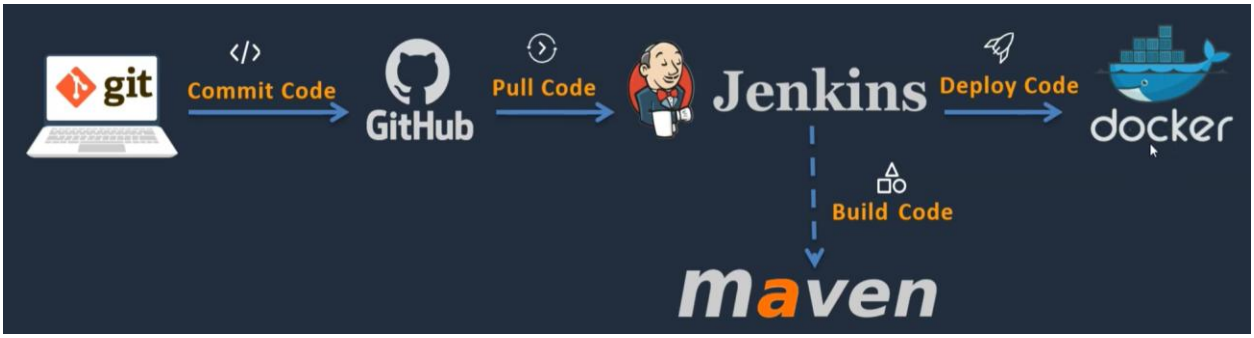
    b. second install Jenkins
    c. install nagrok (to get the local ip in online)
4. Integrate git with Jenkins.
5. Pull the code from github in jenkins server
6. Integrate Maven with Jenkins
    a. Setup Maven on Jenkins server
    b. Setup Environment Variables
    c. JAVA_HOME, M2, 7M2_HOME
    d. Install 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



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:

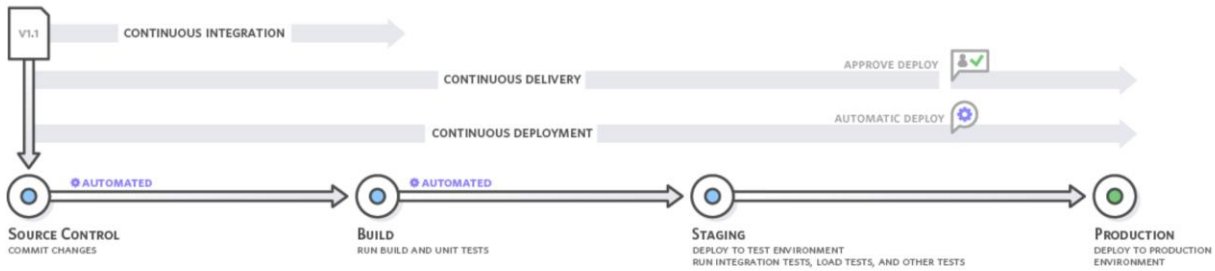
- CI/CD with GitHub, Jenkins, Maven, Ansible and Docker
  - Setup Ansible server
  - Integrate Docker host with Ansible
  - Ansible playbook to create image
  - Ansible playbook to create continuer
  - Integrate Ansible with Jenkins
  - CI/CD job to build code on ansible and deploy it on docker container

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:

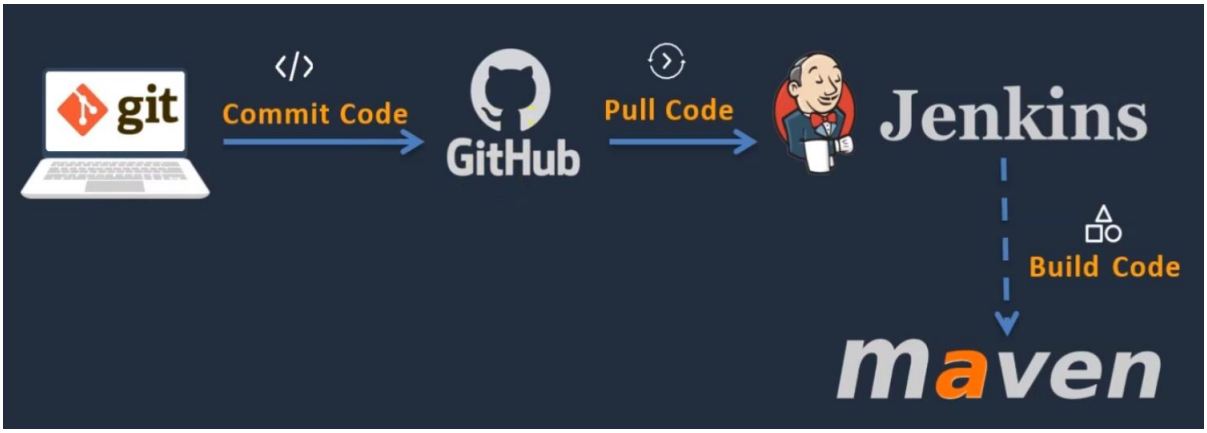


URL: https://aws.amazon.com/devops/continuous-integration/

Steps before start the project

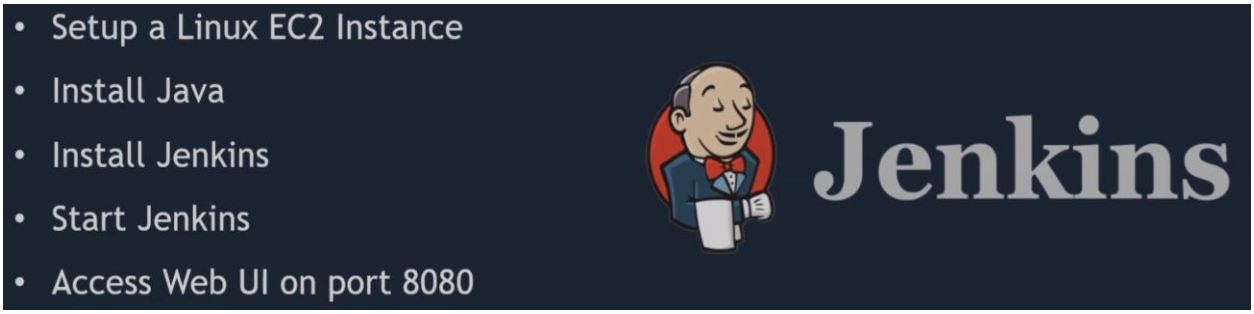
1. Install git bash
2. Clone the source code
3. Install mobaxterm

Now build the code



We have already git and maven installed
Now we are going to setup Jenkins

Setup Jekins server

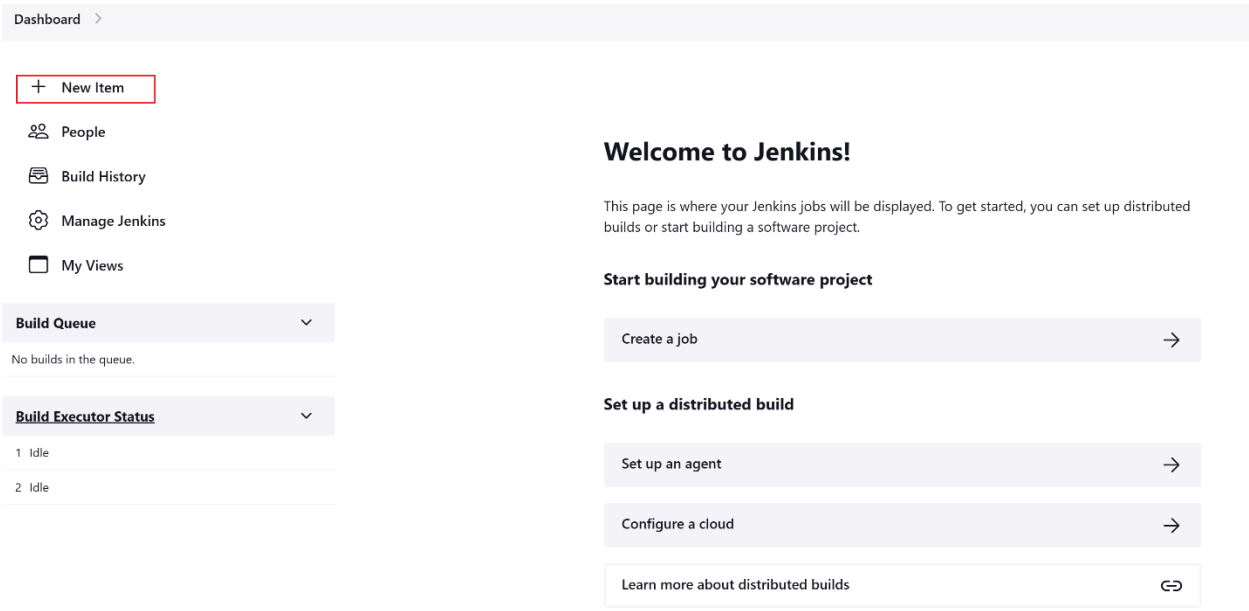


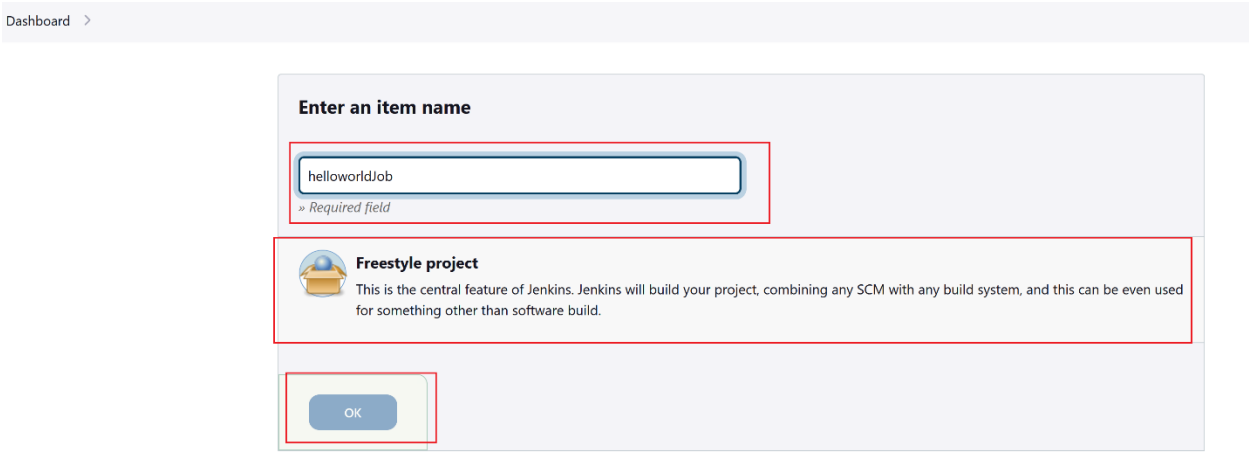
Jekins default passpword location

```
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
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Welcome to Jenkins!
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.
Start building your software project
Create a job
Set up a distributed build
Set up an agent
Configure a cloud
Learn more about distributed builds


Dashboard
Enter an item name
helloworldJob
» 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

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

Poll SCM ?

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

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

Poll SCM ?

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

Dashboard > HellowWorldJob >

- Status
- Changes
- Workspace
- Build Now 1
- Configure
- Delete Project
- Rename

Build History trend

Filter builds... /

#1 2

Aug 18, 2023, 12:57 PM

Atom feed for all Atom feed for failures

Project HellowWorldJob

hellow world job

Permalinks

Dashboard > HellowWorldJob > #1 > Console Output

- 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

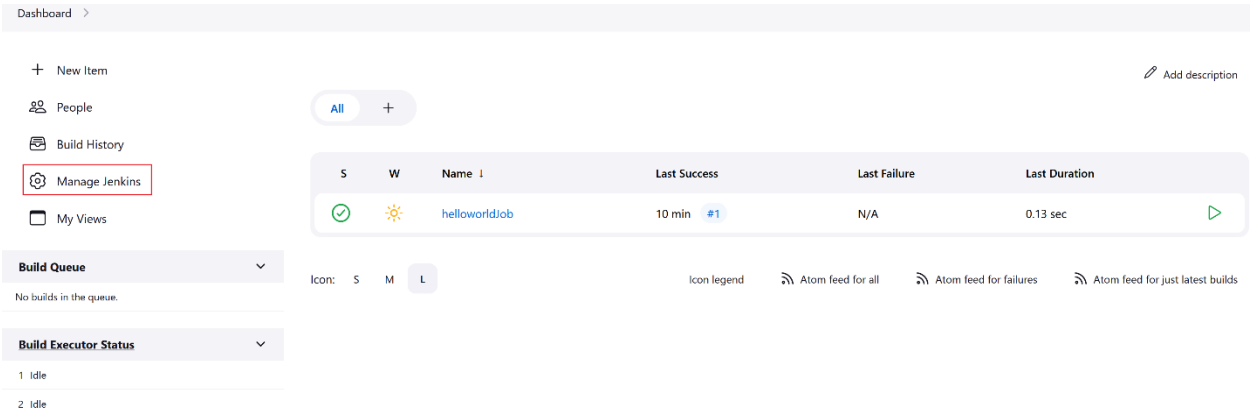
Check git is installed or not
#git –version

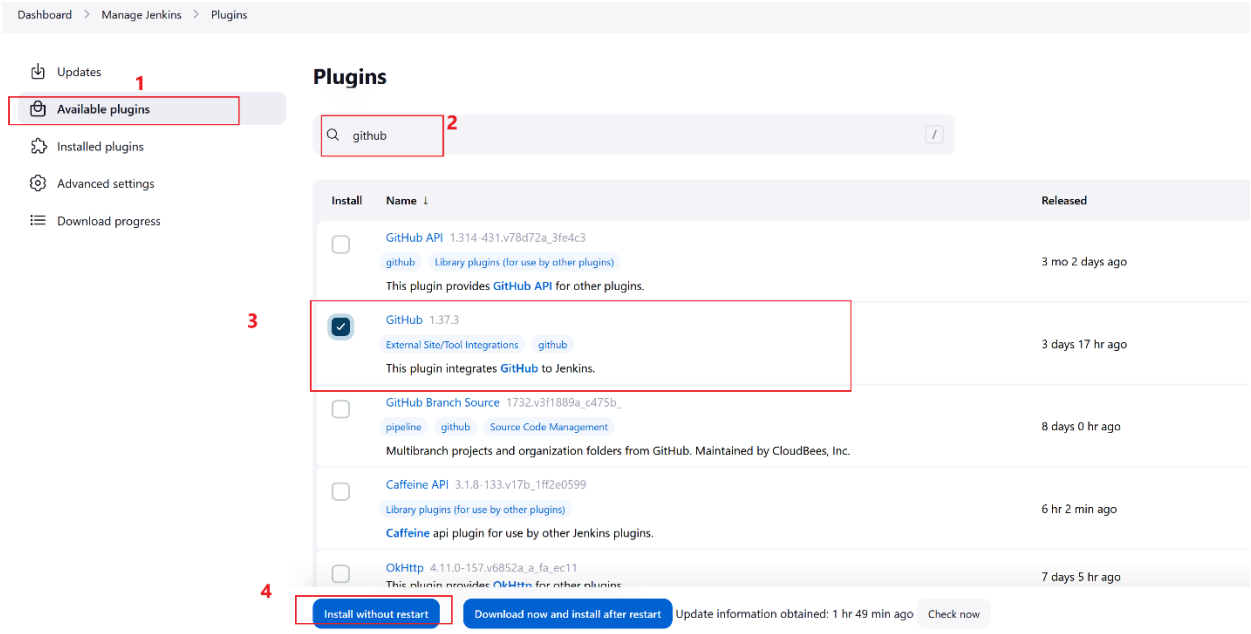
if not then installed

#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




Dashboard > Manage Jenkins > Plugins
Updates
1
Available plugins
Installed plugins
Advanced settings
Download progress
Plugins
2
github
/
Install
Name ↓
Released
GitHub API 1.314-431.v78d72a_3fe4c3
github
Library plugins (for use by other plugins)
This plugin provides GitHub API for other plugins.
3 mo 2 days ago
3
GitHub 1.37.3
External Site/Tool Integrations
github
This plugin integrates GitHub to Jenkins.
3 days 17 hr ago
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
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
OkHttp 4.11.0-157.v6852a_a_fa_ec11
7 days 5 hr ago
4
Install without restart
Download now and install after restart
Update information obtained: 1 hr 49 min ago
Check now

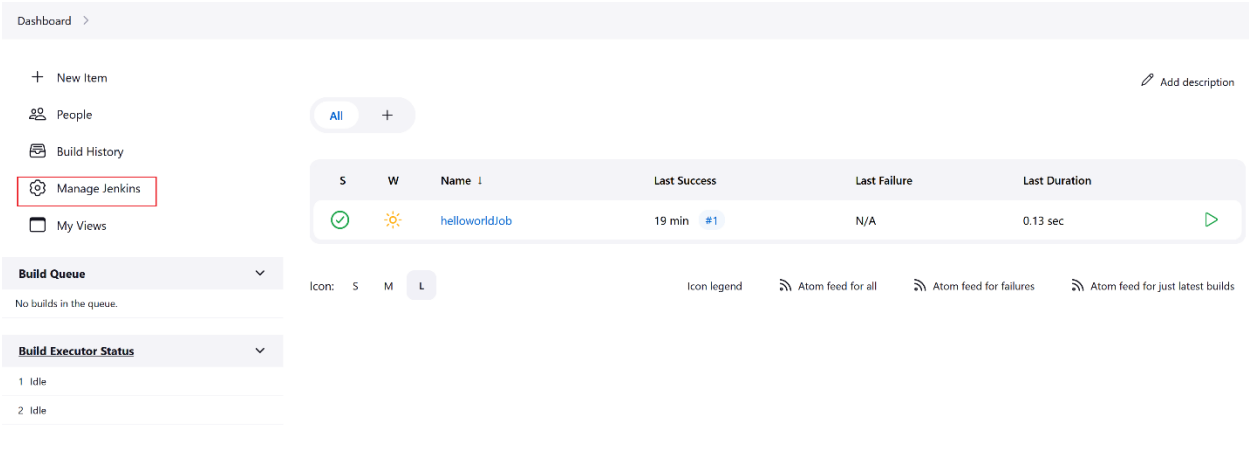
Dashboard > Manage Jenkins > Plugins

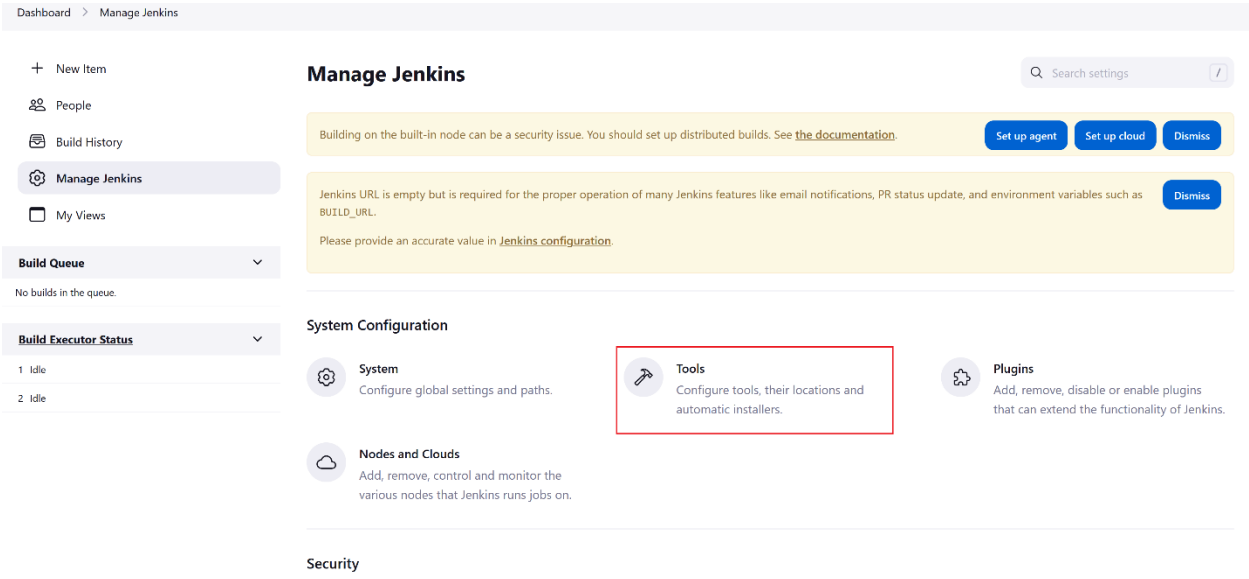
Trilead API	Success
SSH Credentials	Success
Credentials Binding	Success
Pipeline: SCM Step	Success
Mina SSHD API :: Common	Success
Mina SSHD API :: Core	Success
Apache HttpComponents Client 4.x API	Success
Caffeine API	Success
Script Security	Success
Git client	Success
SCM API	Success
Jakarta Activation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

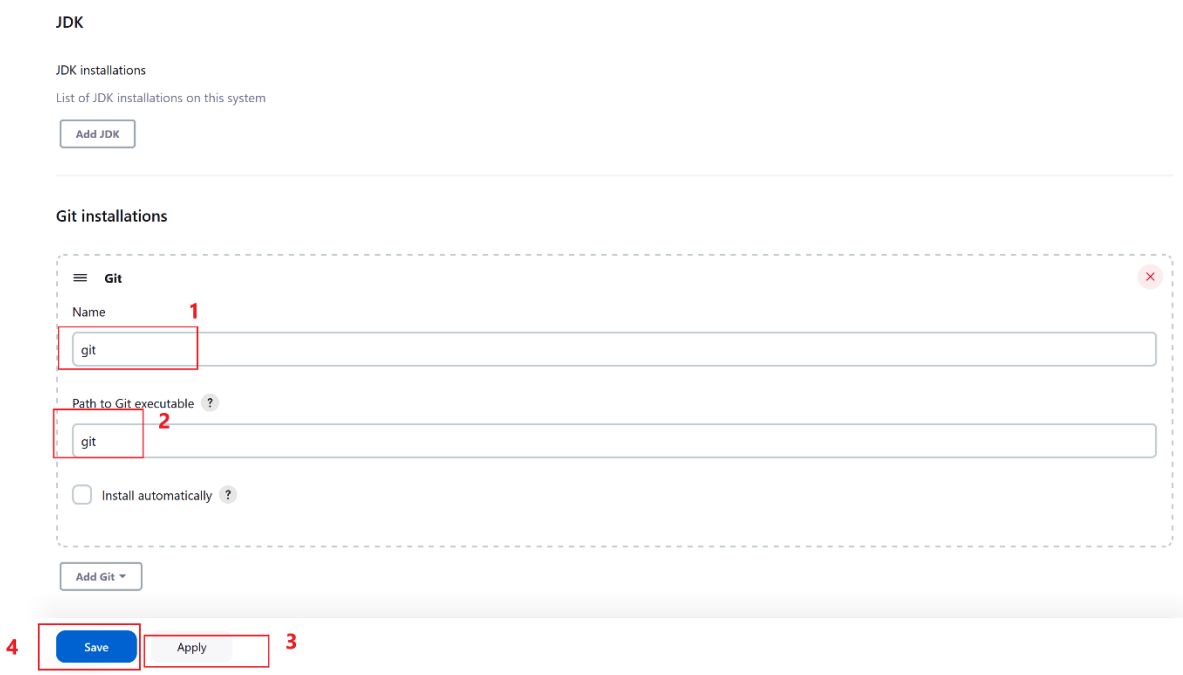
→ Go back to the top page
(you can start using the installed plugins right away)

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

Add git


Dashboard
New Item
People
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Add description
All
S
W
Name
Last Success
Last Failure
Last Duration
helloworldJob
19 min
#1
N/A
0.13 sec
Icon:
S
M
L
Icon legend
Atom feed for all
Atom feed for failures
Atom feed for just latest builds

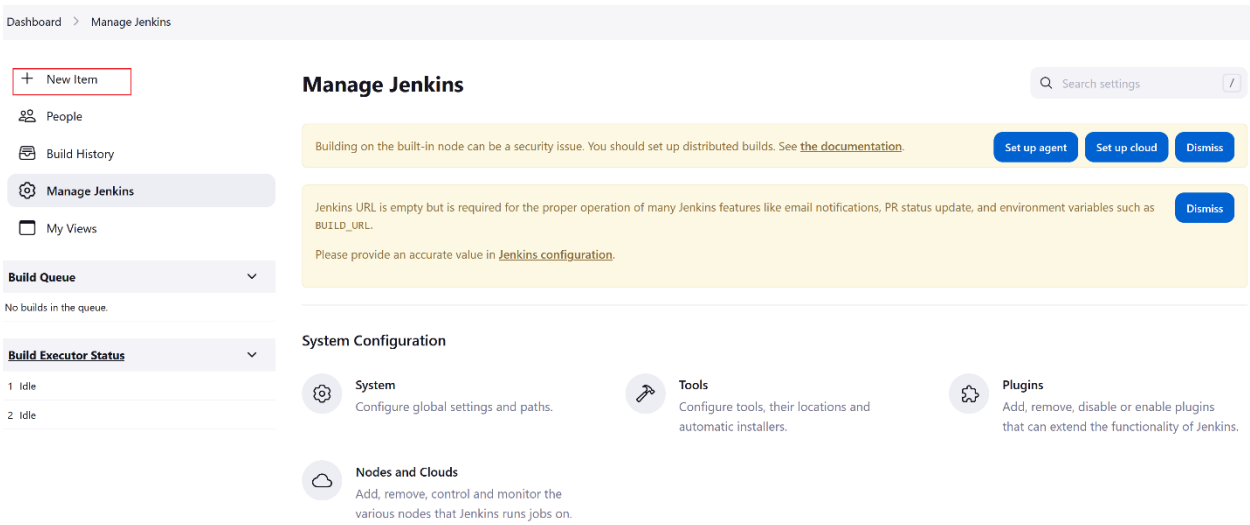

Dashboard
Manage Jenkins
New Item
People
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Manage Jenkins
Search settings
Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.
Set 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
System Configuration
System
Configure global settings and paths.
Tools
Configure tools, their locations and automatic installers.
Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
Security

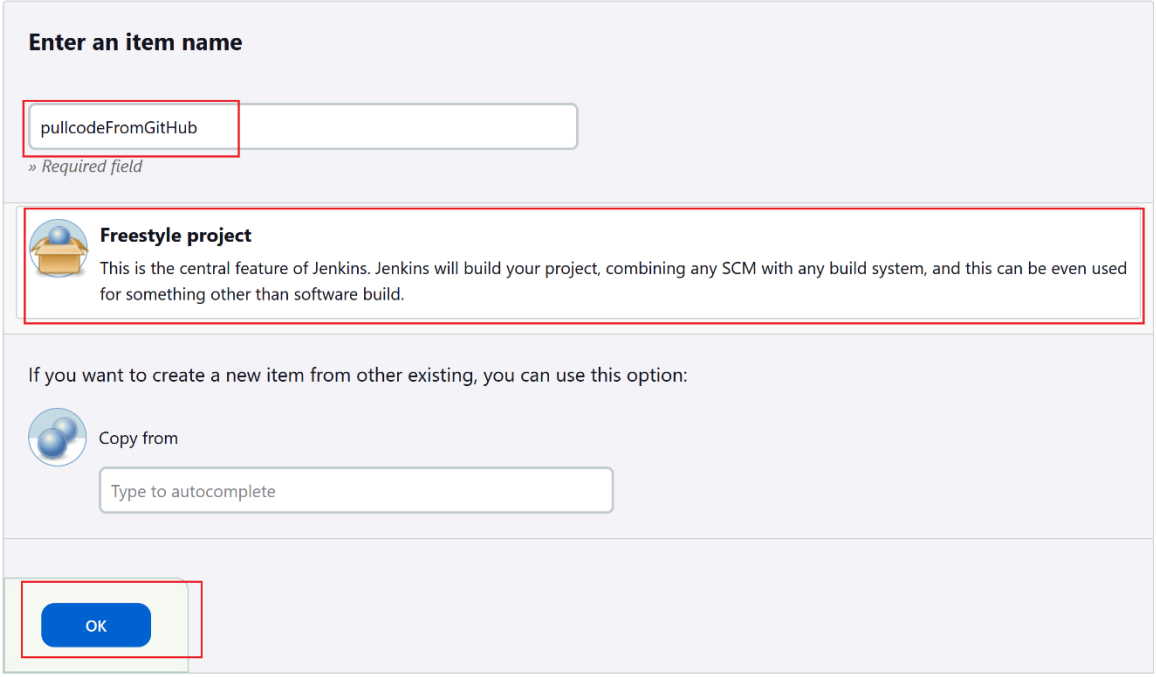


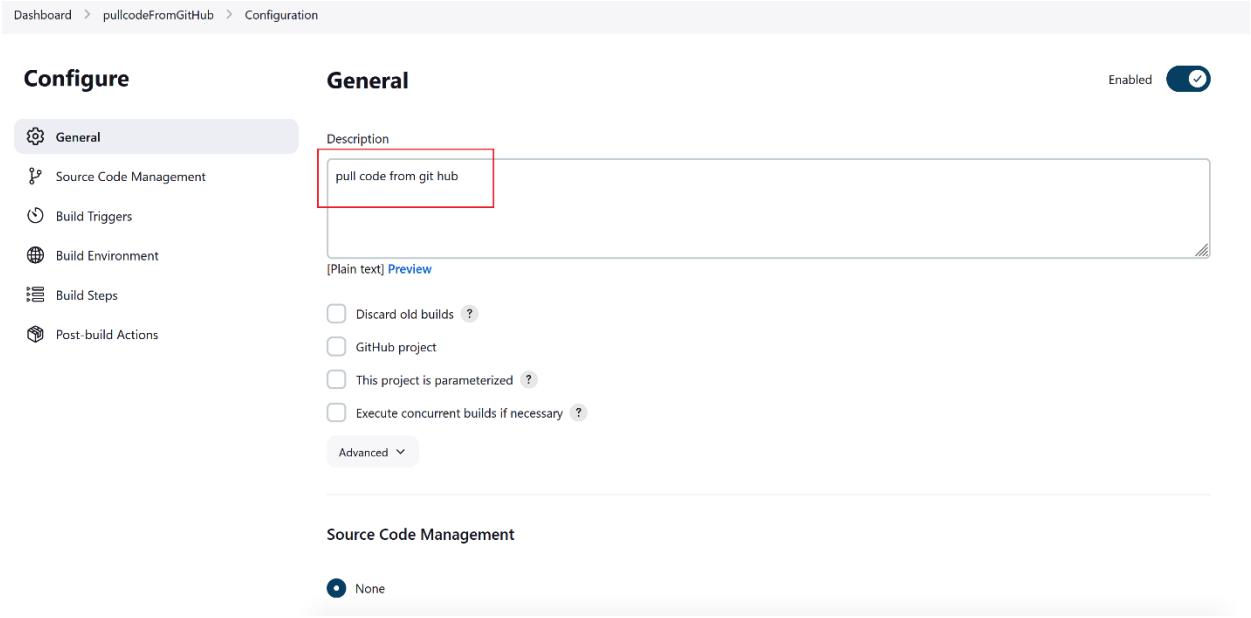
If that not work then we can write the full path of git by using following command

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

Pull code from github




Enter an item name
pullcodeFromGitHub
» 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
Type to autocomplete
OK


Dashboard > pullcodeFromGitHub > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
General
Enabled
Description
pull code from git hub
[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

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

Credentials ?

3

- none -

Add

Advanced

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/master

Add Branch

5

Save

Apply

4

Dashboard > pullcodeFromGitHub >

- Status
- Changes
- Workspace
- Build Now 1
- Configure
- Delete Project
- Rename

Project pullcodeFromGitHub

pull code from git hub

Permalinks

Build History trend

Filter builds... /

2 #1 Aug 15, 2023, 4:41 PM

Atom feed for all Atom feed for failures

Dashboard > pullcodeFromGitHub > #1 > Console Output

- Status
- Changes
- Console Output
- View as plain text
- Edit Build Information
- Delete build '#1'
- Git Build Data

Console Output

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

```
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
Cloning repository https://github.com/rajivsiddiqui/hello-world.git
 > 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
```

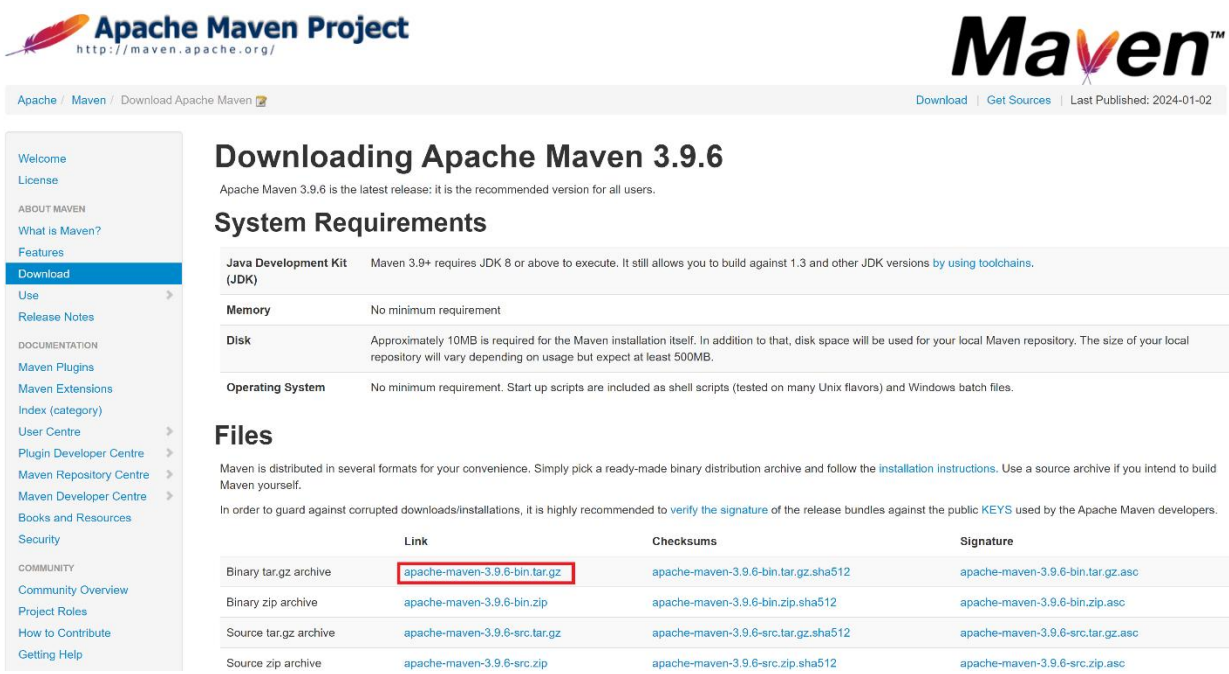
```
root@ip-172-31-32-75:/var/lib/jenkins/workspace
[root@ip-172-31-32-75 ec2-user]# whereis git
git: /usr/bin/git /usr/share/man/man1/git.1.gz
[root@ip-172-31-32-75 ec2-user]# cd /var/lib/jenkins/workspace
[root@ip-172-31-32-75 workspace]# ll
total 0
drwxr-xr-x 2 jenkins jenkins   6 Aug 15 15:28 helloworldJob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



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

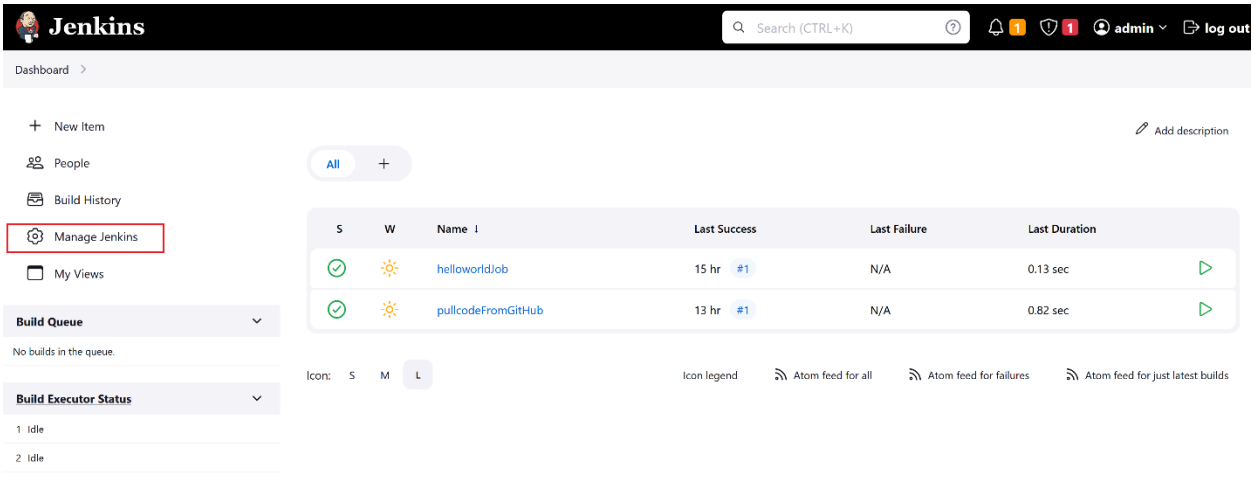
```
#find / -name jvm
#cd /usr/lib/jvm
#find / -name java-11*
```

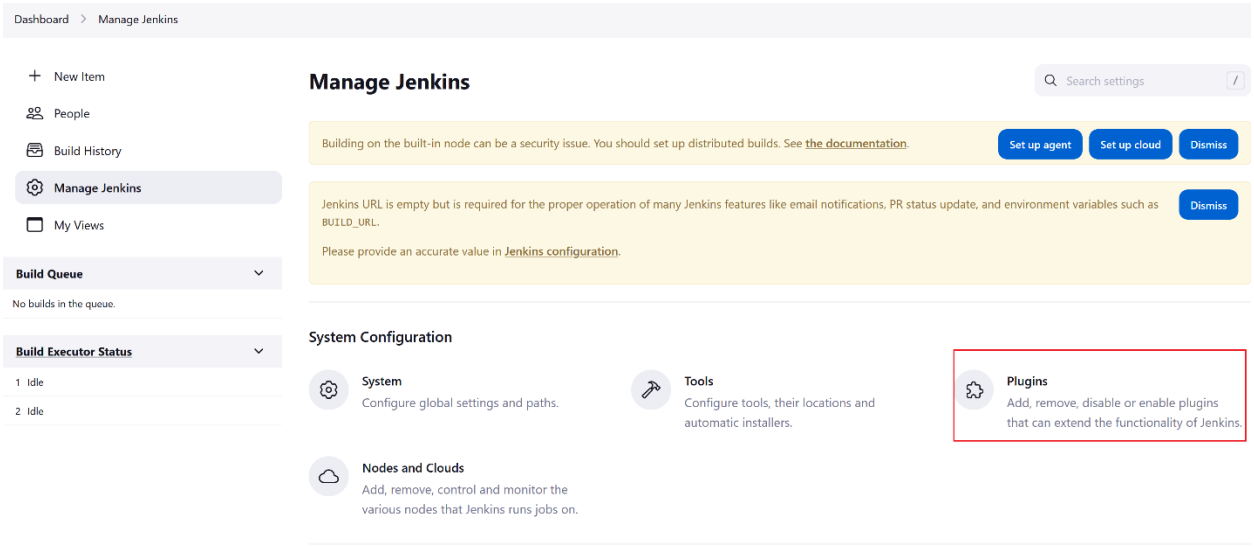
JAVA_HOME when use local VM /usr/lib/jvm/java-11-openjdk-amd64
JAVA_HOME when use aws ec2 /usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.amzn2.0.1.x86_64

```
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
drwxr-xr-x 4 root root 4096 Aug 18 13:27 ./
drwxr-xr-x 4 root root 4096 Aug 18 13:26 ../
drwxr-xr-x 3 root root 4096 Aug 18 13:27 java-11-openjdk-amd64/
drwxr-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
root@ubuntu-light:/usr/share/gdb/auto-load/usr/lib/jvm#
```

this path we need

c) Now Install Maven Plugin in jenkins





Dashboard > Manage Jenkins > Plugins

- Updates
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

Plugins

maven

Install	Name	Released
☑	Maven Integration 3.23 Build Tools This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTs as well as the automated configuration of various Jenkins publishers such as Junit.	11 days ago
☐	Config File Provider 952.va_544a_6234b_46 Groovy-related External Site/Tool Integrations Maven Ability to provide configuration files (e.g. settings.xml for maven, XML, groovy, custom files,...) loaded through the UI which will be copied to the job workspace.	24 days ago
☐	Jira 3.10 External Site/Tool Integrations Maven jira This plugin integrates Jenkins to Atlassian Jira.	2 mo 16 days ago
☐	Pipeline Maven Integration 1322.v9ef317a_3e0a_9 pipeline Maven This plugin provides integration with Pipeline, configures maven environment to use within a pipeline job by calling sh mvn	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

- Updates
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

Download progress

Preparation
- Checking internet connectivity
- Checking update center connectivity
- Success

OkHttp	Success
JavaBeans Activation Framework (JAF) API	Success
JAXB	Success
SnakeYAML API	Success
Jackson 2 API	Success
commons-lang3 v3.x Jenkins API	Success
GitHub API	Success
bouncycastle API	Success
Instance Identity	Success

Dashboard >

- New Item
- People
- Build History
- Manage Jenkins
- My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

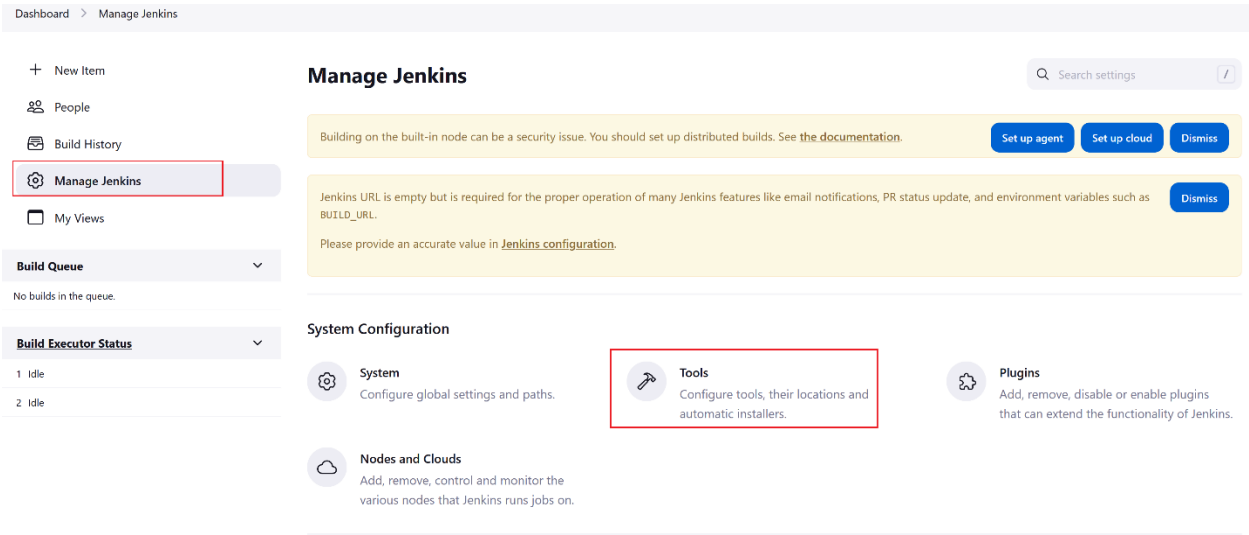
2 Idle

Add description

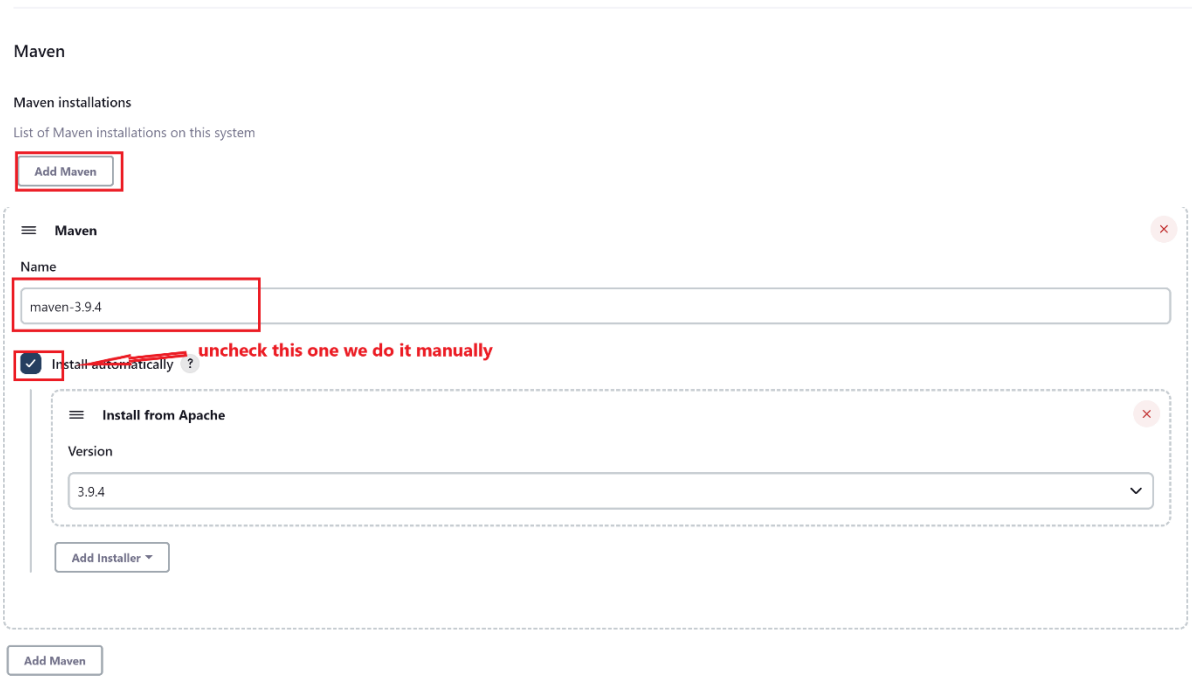
All +

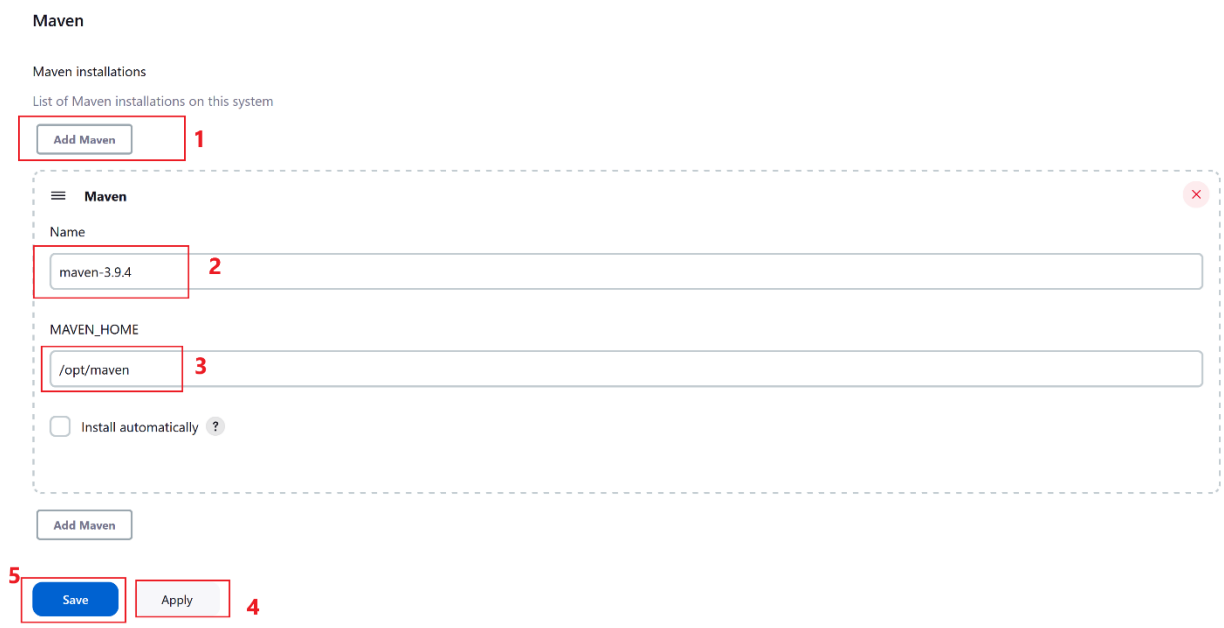
S	W	Name	Last Success	Last Failure	Last Duration
✓	☼	helloworldJob	15 hr #1	N/A	0.13 sec
✓	☼	pullcodeFromGitHub	14 hr #1	N/A	0.82 sec

Icon: S M L | Icon legend | Atom feed for all | Atom feed for failures | Atom feed for just latest builds



<u>If we use vm on ubuntu then java path will be</u>
/usr/lib/jvm/java-11-openjdk-amd64





Create a Maven project

Dashboard > Manage Jenkins

New Item
People
Build History
Manage Jenkins
My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

Manage Jenkins

Search settings /

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

System Configuration

System
Configure global settings and paths.

Tools
Configure tools, their locations and automatic installers.

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

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

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.

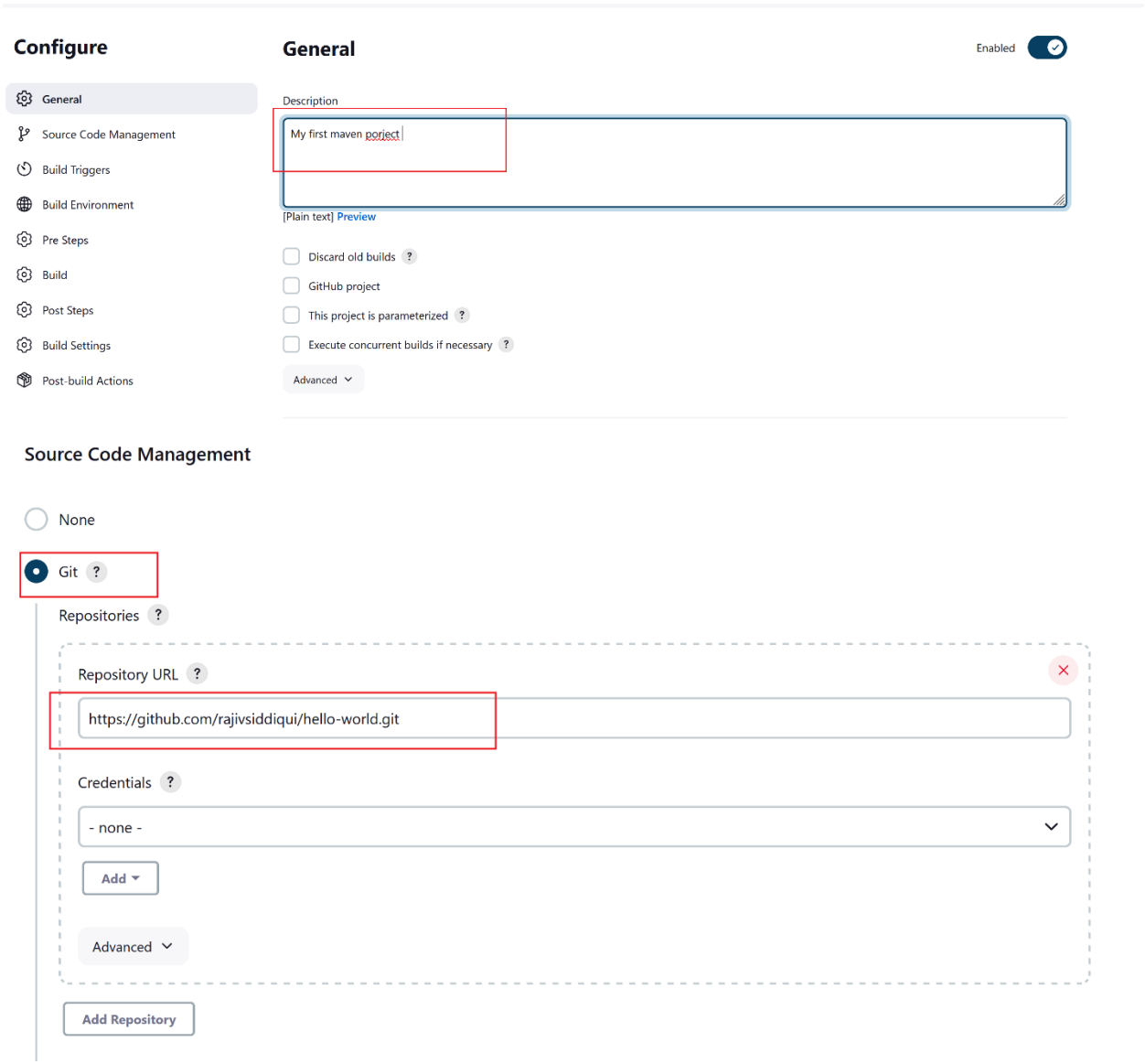
Maven project 2
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

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

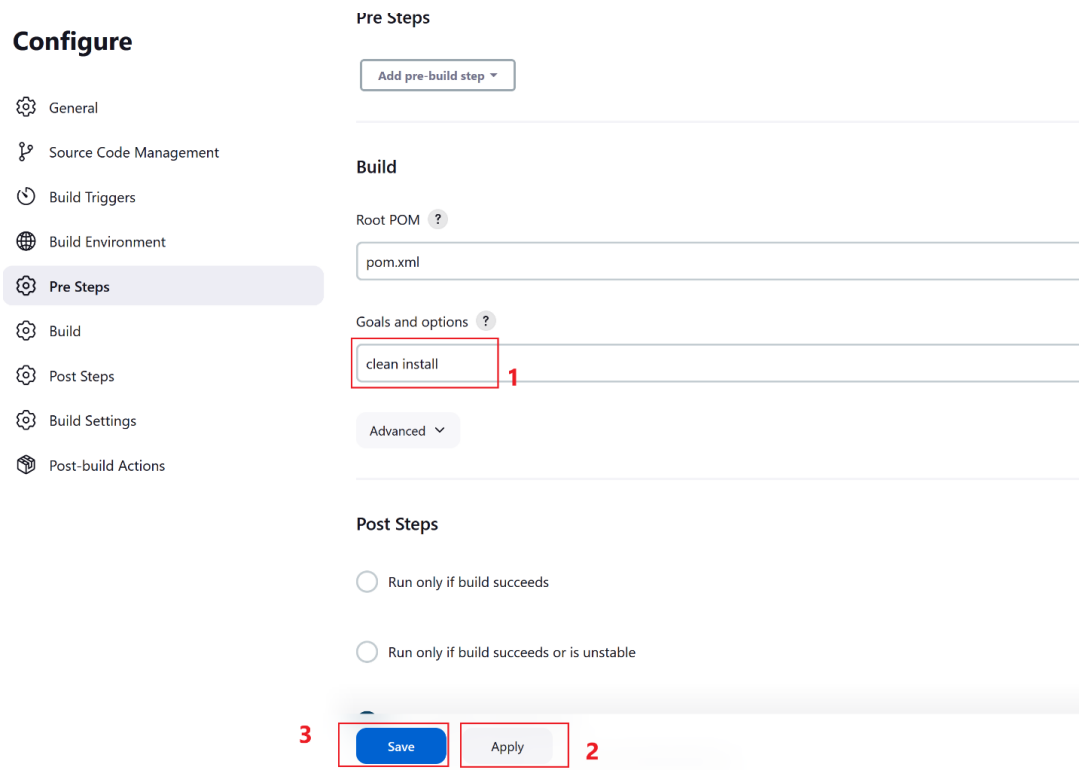
Copy from

Type to autocomplete

OK 3

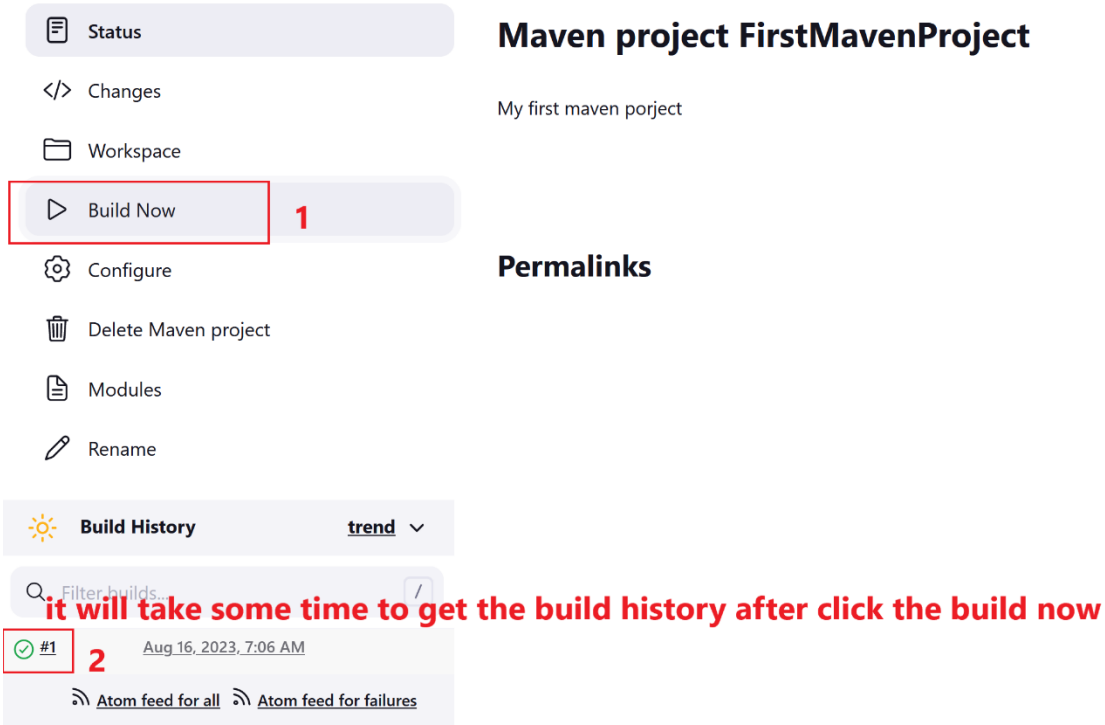


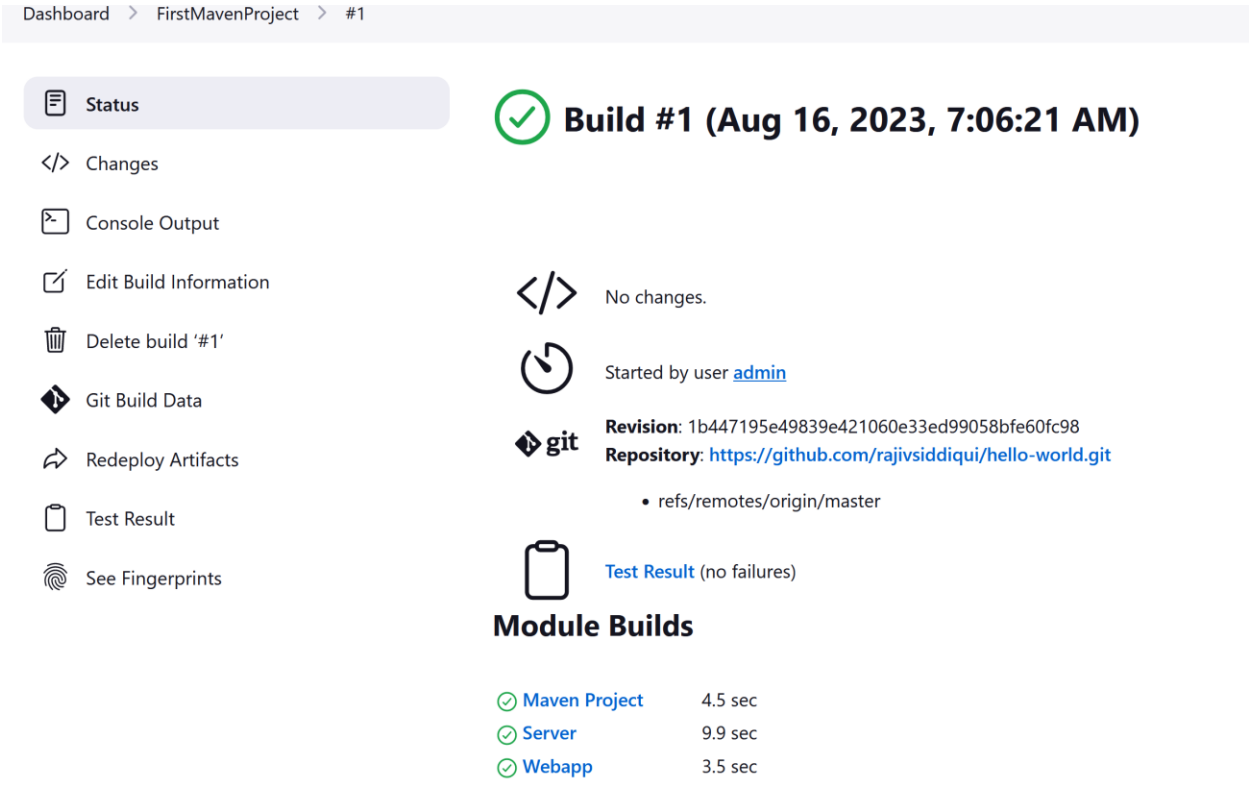
scroll down and add

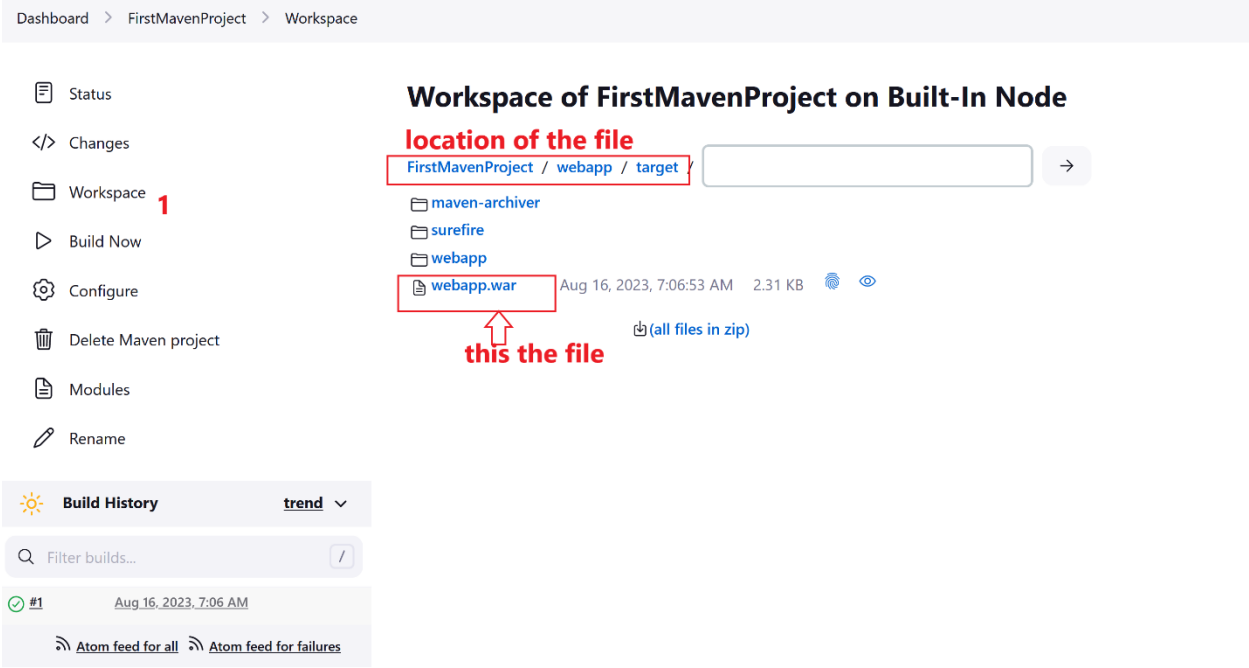


Maven Goals list

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


Status
Changes
Workspace
Build Now
1
Configure
Delete Maven project
Modules
Rename
Build History
trend
Filter builds...
/
#1
2
Aug 16, 2023, 7:06 AM
Atom feed for all
Atom feed for failures
Maven project FirstMavenProject
My first maven porject
Permalinks
it will take some time to get the build history after click the build now

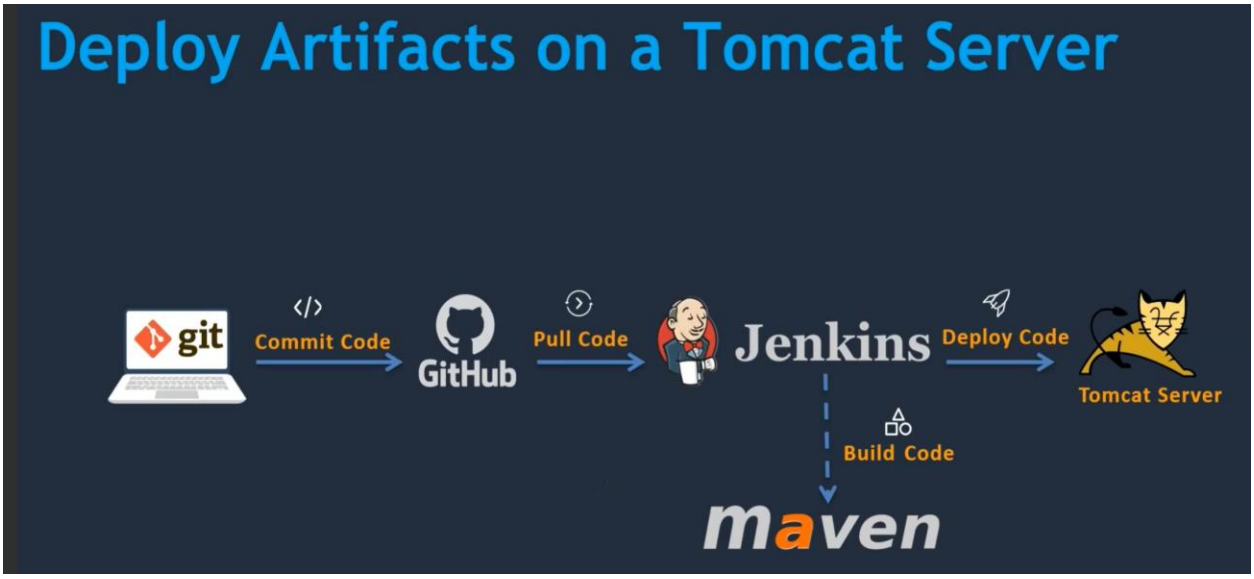




Location of the index file:

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

Phase 1: Deploy artifacts on Tomcat server



a) Setup a linxux 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

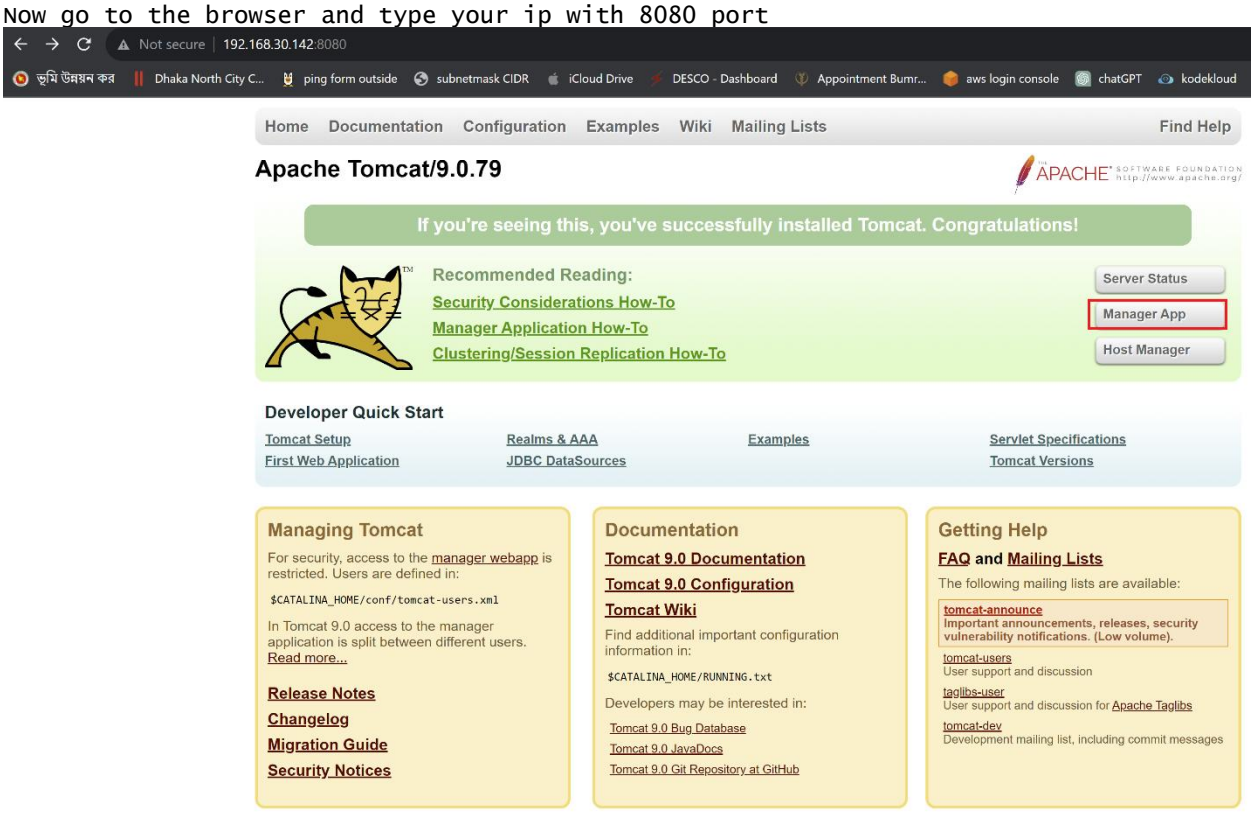
First Install Java -amazon linux-2 ios use

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



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

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

#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

<u>Create users for access the server</u>

#cd /opt/tomcat/conf

```
root@tomcat: /opt/tomcat/conf
root@tomcat:/opt/tomcat/conf# pwd
/opt/tomcat/conf
root@tomcat:/opt/tomcat/conf# ll
total 244
drwx------ 3 root root   4096 Aug 19 05:52 ./
drwxr-xr-x 9 root root   4096 Aug 19 05:44 ../
drwxr-x--- 3 root root   4096 Aug 19 05:52 Catalina/
-rw------- 1 root root  12953 Aug  8 20:41 catalina.policy
-rw------- 1 root root   7630 Aug  8 20:41 catalina.properties
-rw------- 1 root root   1400 Aug  8 20:41 context.xml
-rw------- 1 root root   1149 Aug  8 20:41 jaspic-providers.xml
-rw------- 1 root root   2313 Aug  8 20:41 jaspic-providers.xsd
-rw------- 1 root root   4144 Aug  8 20:41 logging.properties
-rw------- 1 root root   7856 Aug  8 20:41 server.xml
-rw------- 1 root root   2756 Aug  8 20:41 tomcat-users.xml
-rw------- 1 root root   2558 Aug  8 20:41 tomcat-users.xsd
-rw------- 1 root root 172576 Aug  8 20:41 web.xml
root@tomcat:/opt/tomcat/conf#
```

#vi tomcat-users.xml
add the following line at the bottom of the file

```
<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"/>
```

```
root@tomcat: /opt/tomcat/conf
<!--
  The sample user and role entries below are intended for use with the
  examples web application. They are wrapped in a comment and thus are ignored
  when reading this file. If you wish to configure these users for use with the
  examples web application, do not forget to remove the <!.. ..> that surrounds
  them. You will also need to set the passwords to something appropriate.
-->
<!--
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
  <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
  <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-status"/>
 <user username="deployer" password="deployer" roles="manager-script"/>
 <user username="tomcat" password="s3cret" roles="manager-gui"/>
</tomcat-users>
-- INSERT (paste) --                                                    64,16          Bot
```

Now restart the tomcat and access the server by using the user name and password

Start and stop tomcat server

#cd tomcat/bin
#ll
#./startup.sh

Create a link file for easy way to start and stop tomcat

ln -s /opt/tomcat/bin/startup.sh /usr/local/bin/tomcatup

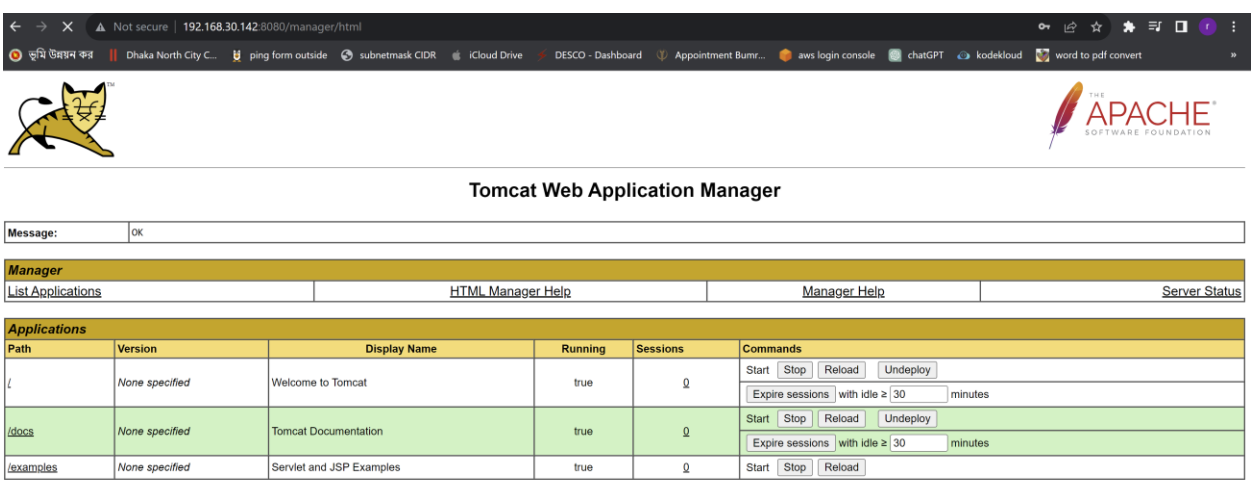
```
ln -s /opt/tomcat/bin/shutdown.sh /usr/local/bin/tomcatdown

now try to start tomcat
#tomcatup
#tomcatdown

if not work then do the following
go to /root
vi .bash_profile
and the following line
PATH=$PATH:$HOME/.local/bin:$HOME/bin:/usr/local/bin

Now use following command
source .bash_profile
echo $PATH
then try
#tomcatstart
#tomcatdown
```

```
root@ip-172-31-46-215:/home/ec2-user
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/.local/bin:$HOME/bin:/usr/local/bin

export PATH
~
~
~
```

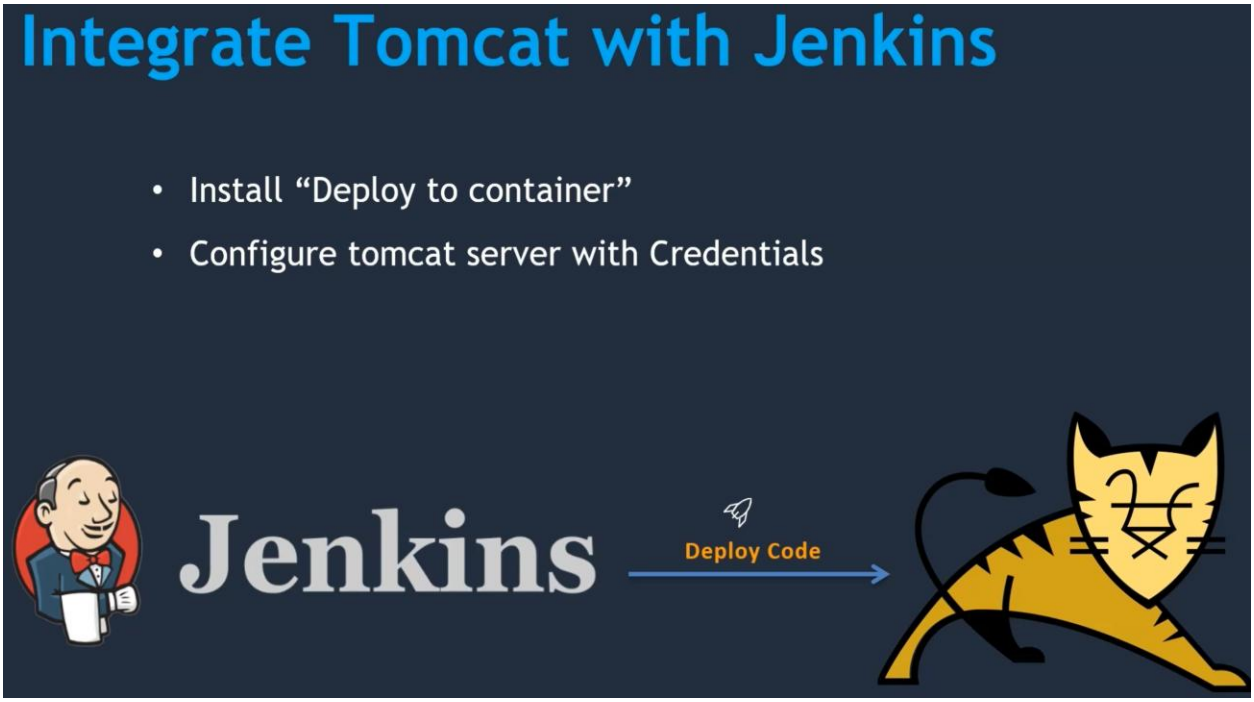
Now we can access the tomcat server by using the user name and password

username: tomcat

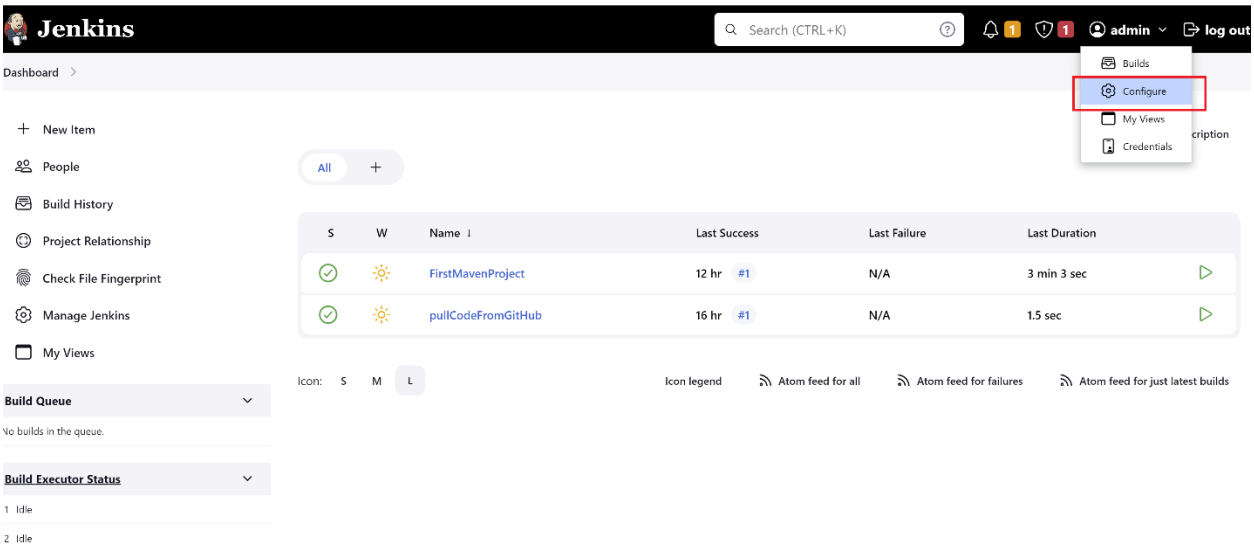
password: s3cret



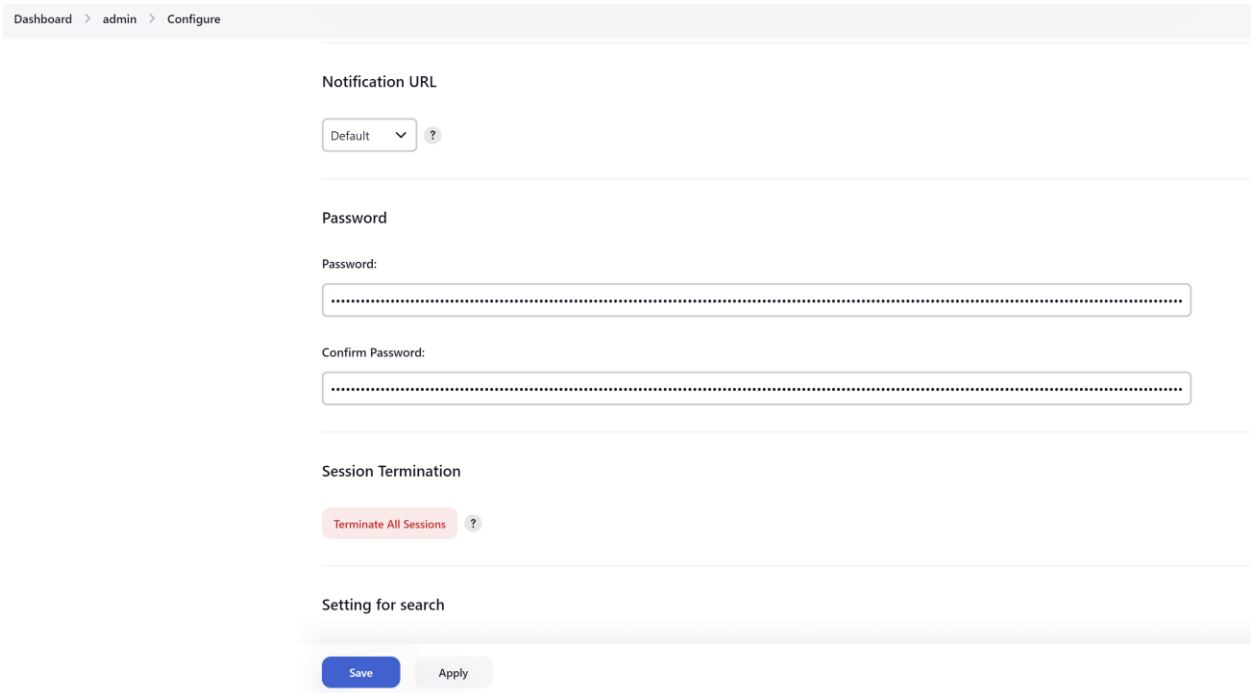
f)Integrate tomcat with jenkins



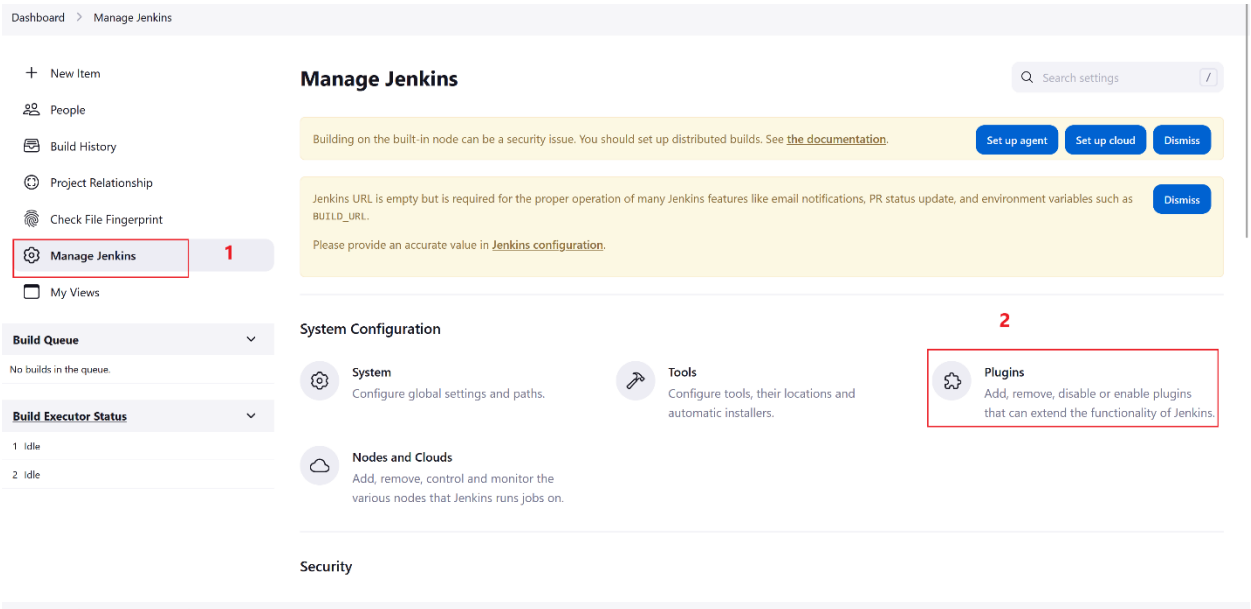
Edit the jenkins password.

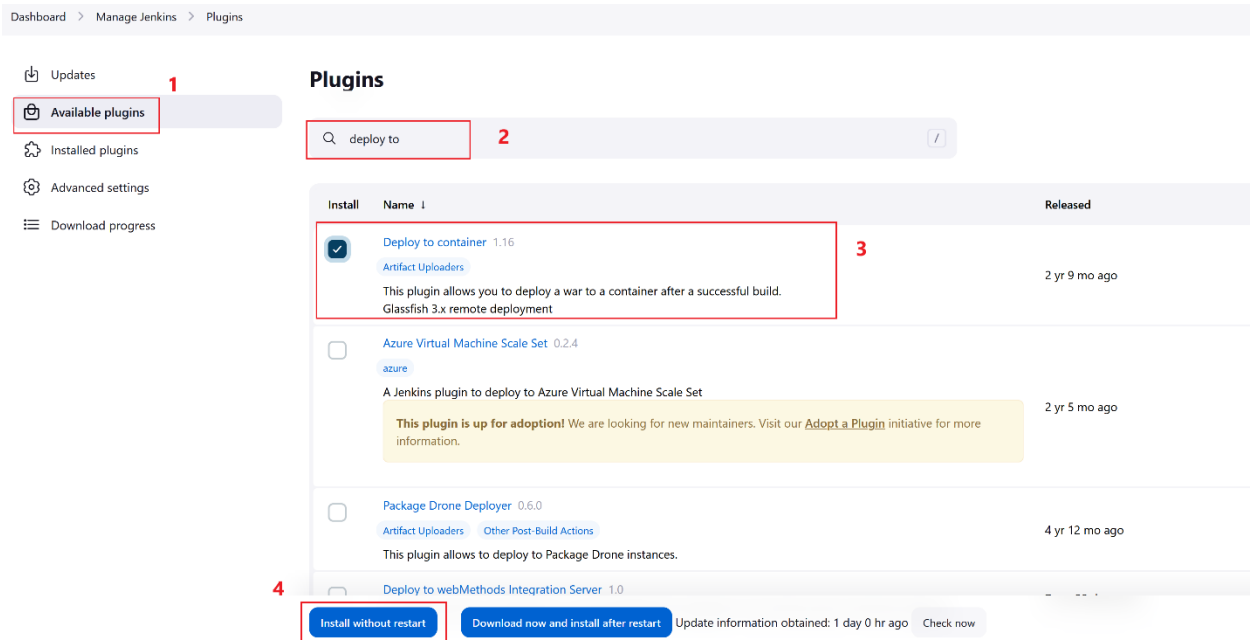


Scroll down and set your password



Install deploy to container plugin





Now set our username and password to Jenkins to access tomcat server

Dashboard >

- New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

Add description

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☀	FirstMavenProject	7 hr 32 min #1	N/A	33 sec	▷
✓	☀	helloworldJob	23 hr #1	N/A	0.13 sec	▷
✓	☀	pullcodeFromGitHub	21 hr #1	N/A	0.82 sec	▷

Icon: S M L Icon legend Atom feed for all Atom feed for failures Atom feed for just latest builds

Dashboard > Manage Jenkins

- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

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

System Configuration

System
Configure global settings and paths.

Tools
Configure tools, their locations and automatic installers.

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

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

Security

Security
Secure Jenkins; define who is allowed to access/use the system.

Credentials
Configure credentials

Credential Providers
Configure the credential providers and types

Users
Create/delete/modify users that can log in to this Jenkins.

Dashboard > Manage Jenkins > Credentials

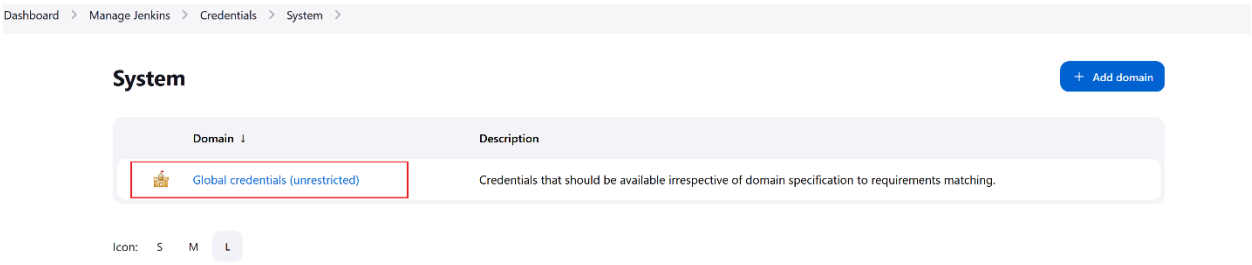
Credentials

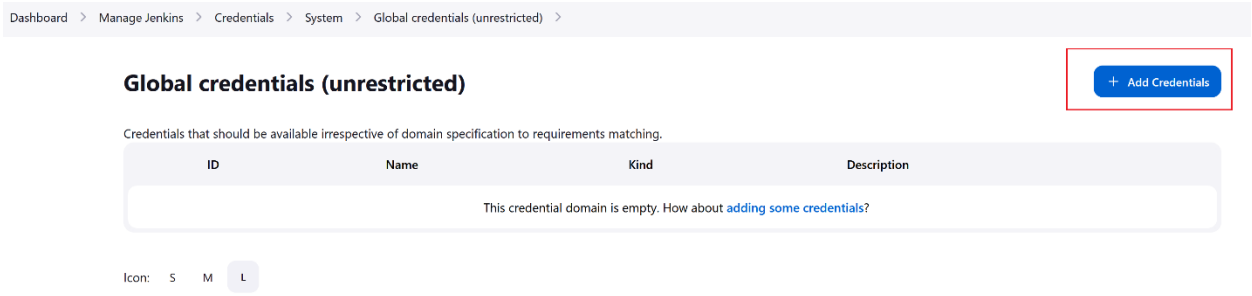
T	P	Store ↓	Domain	ID	Name

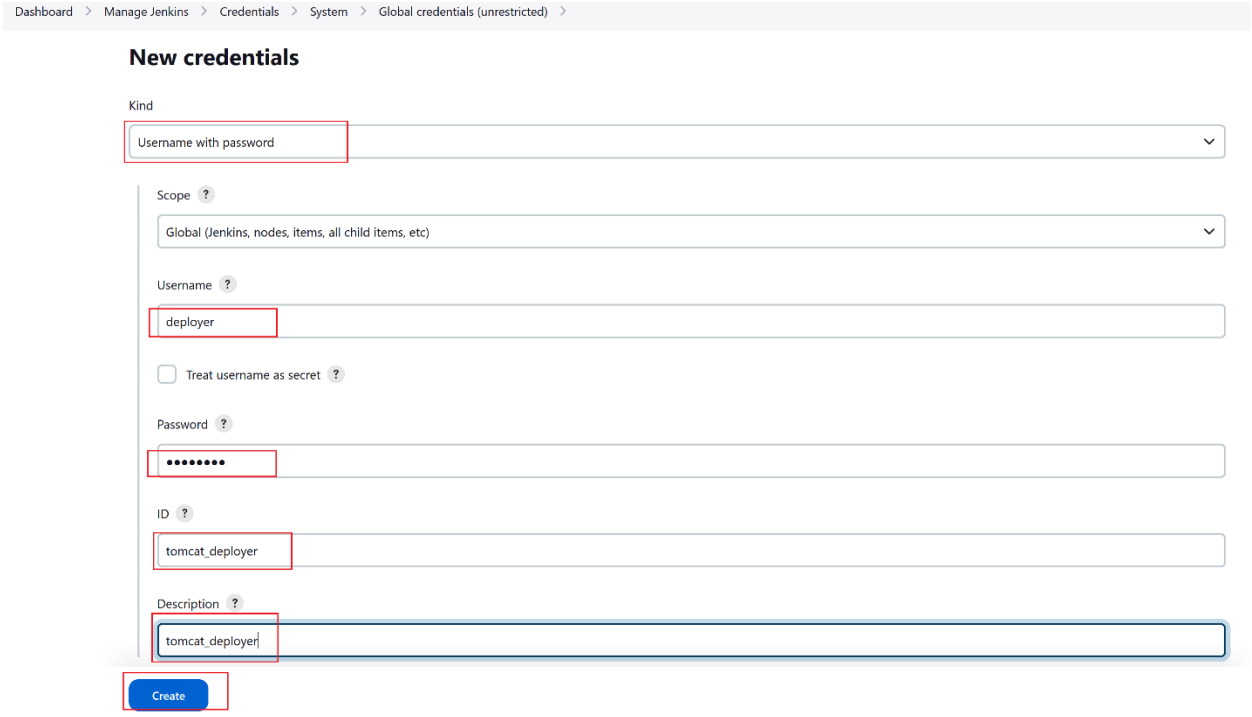
Stores scoped to Jenkins

P	Store ↓	Domains
	System	(global)

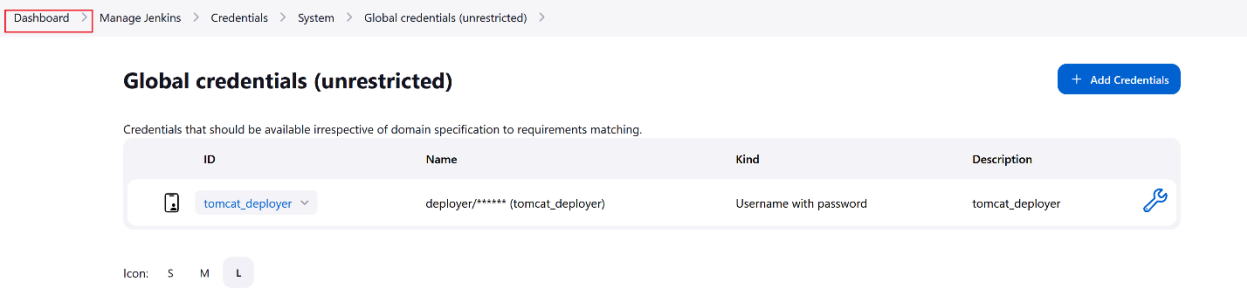
Icon: S M L



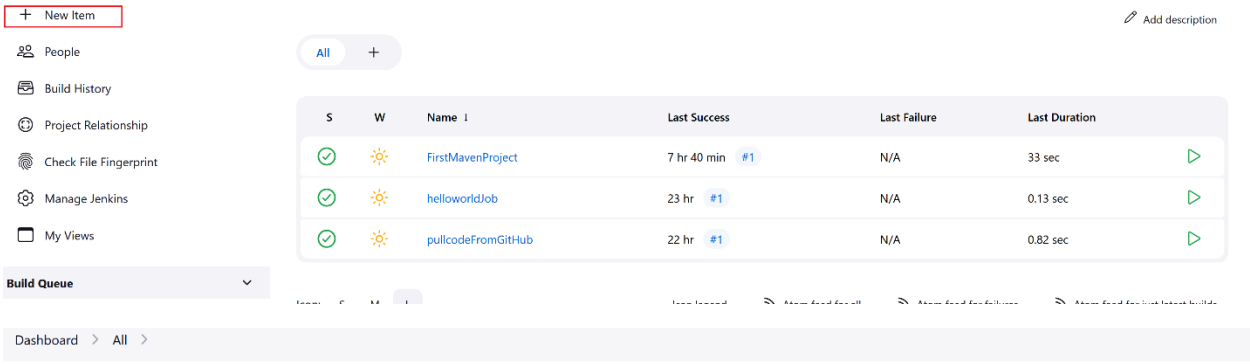


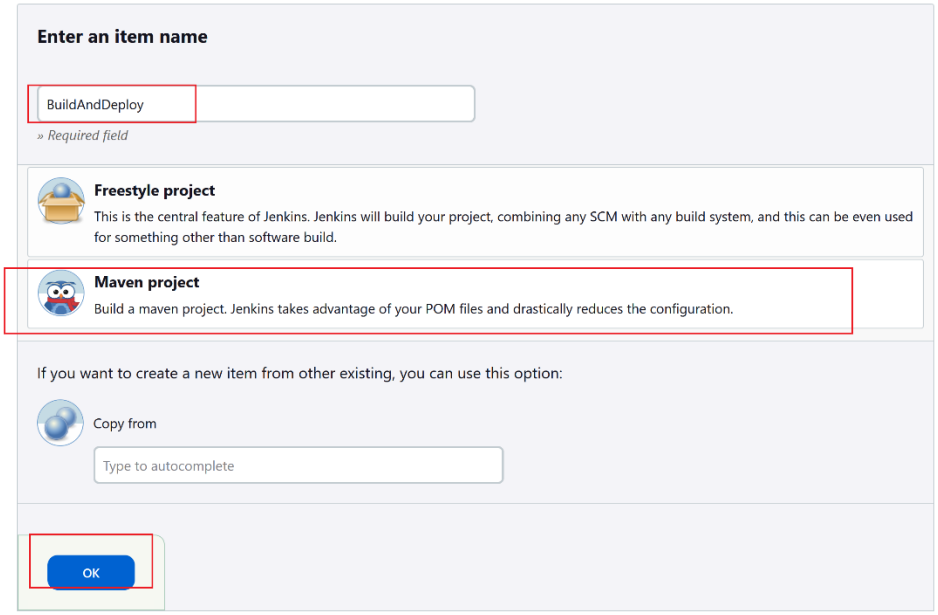


username: deployer
password: deployer



Build and deploy website to tomcat server





Dashboard > BuildAndDeploy > Configuration

Configure

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

General

Enabled

Description

Build code and deploy with the help of maven to tomcat server

[Plain text] Preview

- [ ] Discard old builds
- [ ] GitHub project
- [ ] This project is parameterized
- [ ] Execute concurrent builds if necessary

Advanced

Configure

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

Source Code Management

- ( ) None
- (•) Git

Repositories

Repository URL

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

Credentials

- none -

Add

Advanced

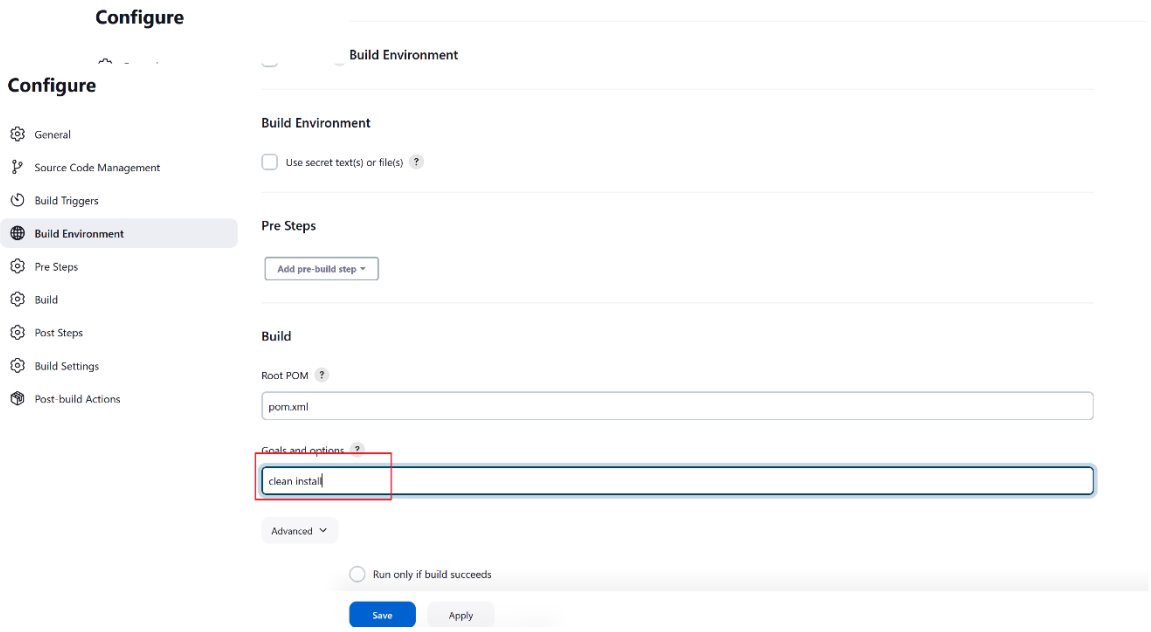
Add Repository

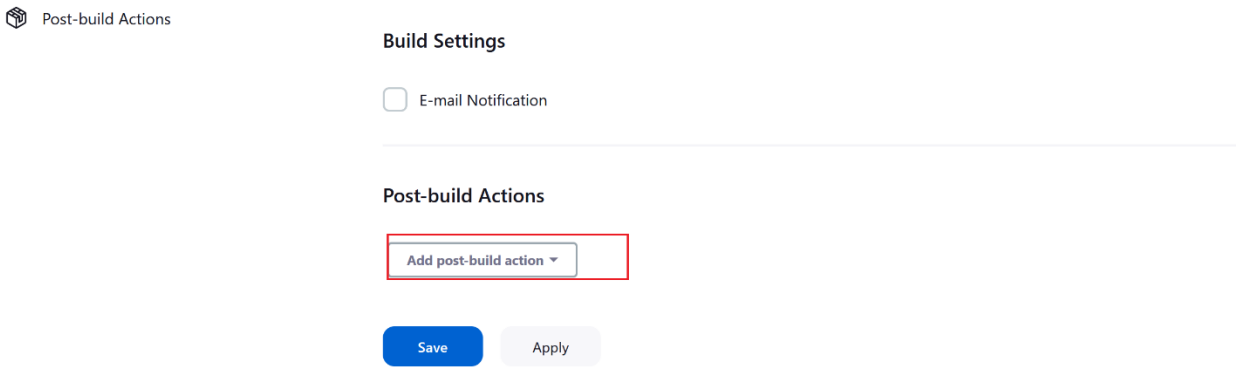
Branches to build

Branch Specifier (blank for 'any')

*/master

Configure


Build Environment
Configure
General
Source Code Management
Build Triggers
Build Environment
Pre Steps
Build
Post Steps
Build Settings
Post-build Actions
Build Environment
Use secret text(s) or file(s)
Pre Steps
Add pre-build step
Build
Root POM
pom.xml
Goals and options
clean install
Advanced
Run only if build succeeds
Save
Apply


Post-build Actions
Build Settings
E-mail Notification
Post-build Actions
Add post-build action
Save
Apply

Dashboard > BuildAndDeploy > Configuration

Configure

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

Advanced

Post Steps

- ( ) Run only if build succeeds
- ( ) Run only if build succeeds or is unstable
- (•) Run regardless of build result

Should the post-build steps run only for successful builds, etc.

Add post-build step

Filter

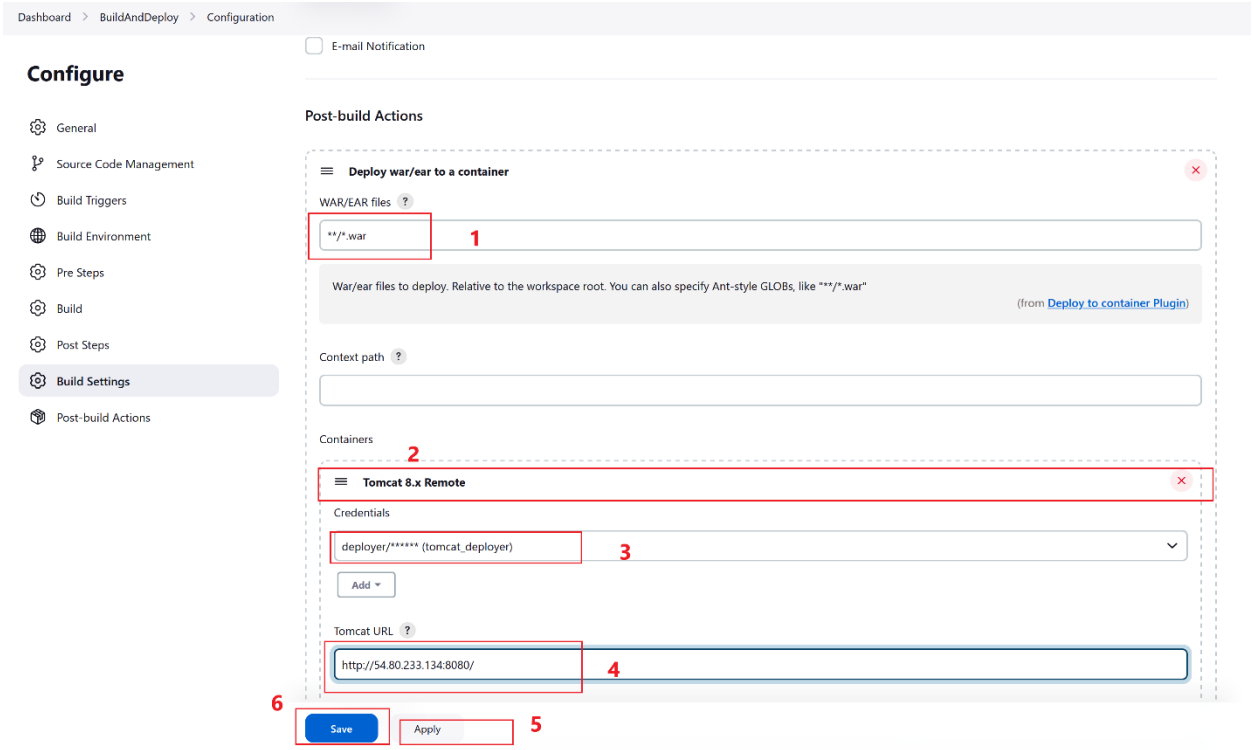
- Aggregate downstream test results
- Archive the artifacts
- Build other projects
- Deploy artifacts to Maven repository
- Record fingerprints of files to track usage
- Git Publisher
- Deploy war/ear to a container
- Set GitHub commit status (universal)
- Set build status on GitHub commit [deprecated]

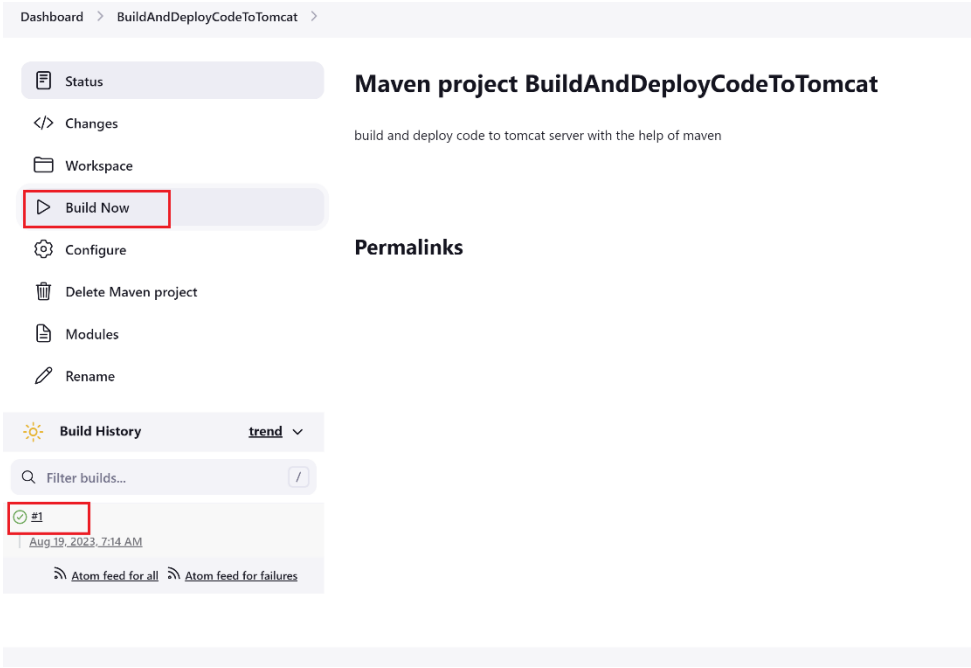
Add post-build action

Save Apply

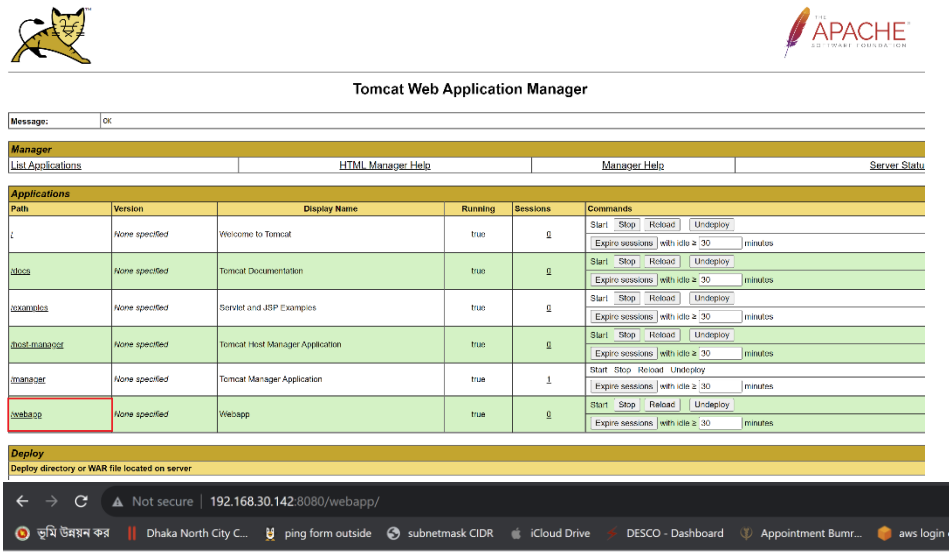
Get the WAR file location

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-32-75 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-32-75 ~]$ sudo su
[root@ip-172-31-32-75 ec2-user]# cd /var/lib/jenkins/workspace/
[root@ip-172-31-32-75 workspace]# ll
total 0
drwxr-xr-x 5 jenkins jenkins 147 Aug 16 07:06 FirstMavenProject
drwxr-xr-x 2 jenkins jenkins   6 Aug 15 15:28 helloworldJob
drwxr-xr-x 5 jenkins jenkins 147 Aug 15 16:41 pullcodeFromGitHub
[root@ip-172-31-32-75 workspace]# cd FirstMavenProject/
[root@ip-172-31-32-75 FirstMavenProject]# ll
total 24
-rw-r--r-- 1 jenkins jenkins  130 Aug 16 07:06 Dockerfile
-rw-r--r-- 1 jenkins jenkins 5970 Aug 16 07:06 pom.xml
-rw-r--r-- 1 jenkins jenkins  271 Aug 16 07:06 README.md
-rw-r--r-- 1 jenkins jenkins  479 Aug 16 07:06 regapp-deploy.yml
-rw-r--r-- 1 jenkins jenkins  195 Aug 16 07:06 regapp-service.yml
drwxr-xr-x 4 jenkins jenkins   46 Aug 16 07:06 server
drwxr-xr-x 4 jenkins jenkins   46 Aug 16 07:06 webapp
[root@ip-172-31-32-75 FirstMavenProject]# cd webapp/
[root@ip-172-31-32-75 webapp]# ll
total 4
-rw-r--r-- 1 jenkins jenkins 1141 Aug 16 07:06 pom.xml
drwxr-xr-x 3 jenkins jenkins   18 Aug 16 07:06 src
drwxr-xr-x 5 jenkins jenkins   76 Aug 16 07:06 target
[root@ip-172-31-32-75 webapp]# cd target/
[root@ip-172-31-32-75 target]# ll
total 4
drwxr-xr-x 2 jenkins jenkins   28 Aug 16 07:06 maven-archiver
drwxr-xr-x 2 jenkins jenkins    6 Aug 16 07:06 surefire
drwxr-xr-x 4 jenkins jenkins   54 Aug 16 07:06 webapp
-rw-r--r-- 1 jenkins jenkins 2367 Aug 16 07:06 webapp.war
[root@ip-172-31-32-75 target]# pwd
/var/lib/jenkins/workspace/FirstMavenProject/webapp/target
[root@ip-172-31-32-75 target]#
```





Now browse the tomcat ip and we can see the new website is launch



Tomcat Web Application Manager

Message:	OK

Manager

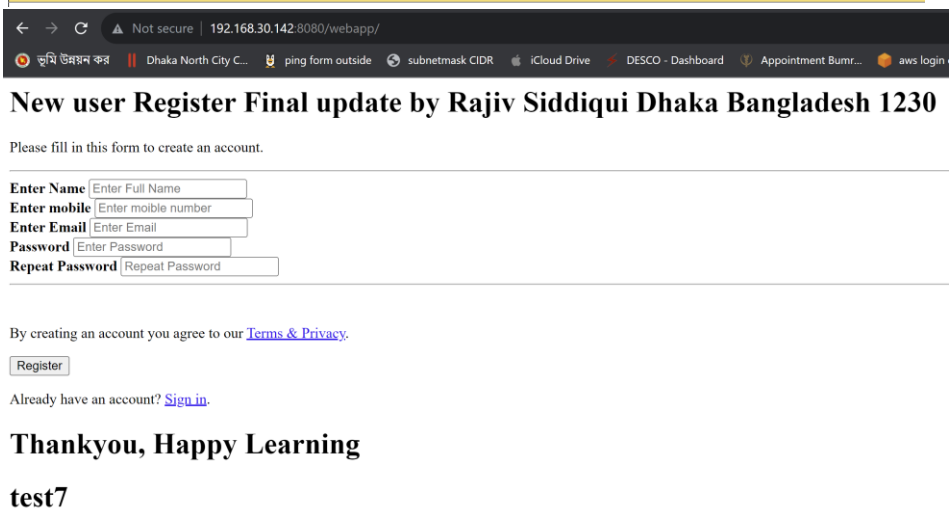
List Applications	HTML Manager Help	Manager Help	Server Statu

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	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/webapp	None specified	Webapp	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server



Not secure | 192.168.30.142:8080/webapp/

ভূমি উন্নয়ন কর | Dhaka North City C... | ping form outside | subnetmask CIDR | iCloud Drive | DESCO - Dashboard | Appointment Bumr... | aws login

New user Register Final update by Rajiv Siddiqui Dhaka Bangladesh 1230

Please fill in this form to create an account.

Enter Name Enter Full Name
Enter mobile Enter moible number
Enter Email Enter Email
Password Enter Password
Repeat Password Repeat Password

By creating an account you agree to our Terms & Privacy.

Register

Already have an account? Sign in.

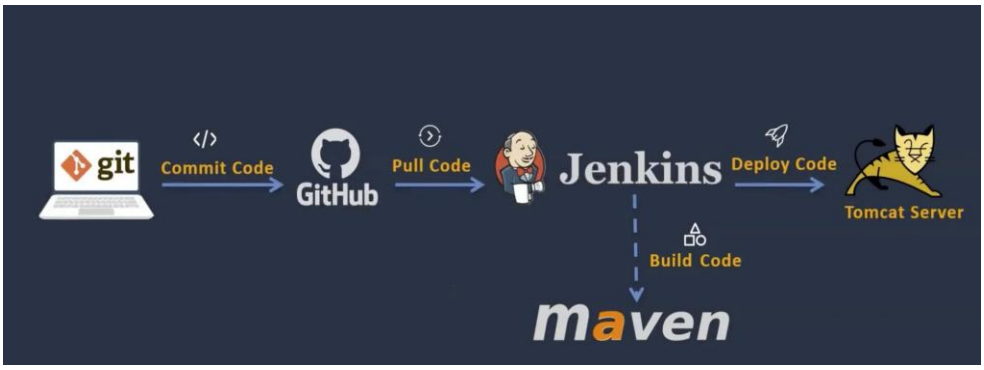
Thankyou, Happy Learning

test7

Tomcat file website file location
`/opt/tomcat/webapps/webapp`

Jenkins war file for website launch in tomcat
`/var/lib/jenkins/workspace/BuildAndDeploy/webapp/target`

Any change in git automatically update the website



Now clone the project from github

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

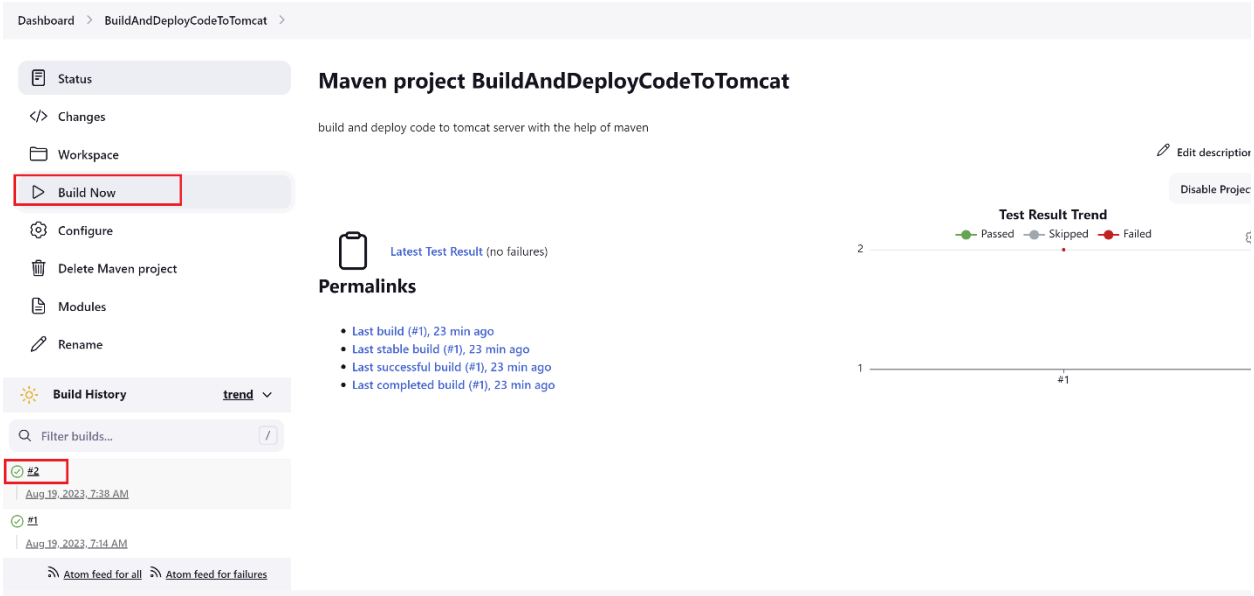
Now edit the file

```
MINGW64:/d/arena-devops-project/hello-world/webapp/src/main/webapp

aweso@RAJIV-XPS17 MINGW64 /d/arena-devops-project
$ git clone https://github.com/rajivsiddiqui/hello-world.git
Cloning into 'hello-world'...
remote: Enumerating objects: 525, done.
remote: Counting objects: 100% (51/51), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 525 (delta 13), reused 47 (delta 10), pack-reused 474
Receiving objects: 100% (525/525), 55.80 KiB | 627.00 KiB/s, done.
Resolving deltas: 100% (121/121), done.

aweso@RAJIV-XPS17 MINGW64 /d/arena-devops-project
$ ll
total 4
drwxr-xr-x 1 aweso 197609 0 Aug 19 13:29 hello-world/

aweso@RAJIV-XPS17 MINGW64 /d/arena-devops-project
$ cd hello-world/

aweso@RAJIV-XPS17 MINGW64 /d/arena-devops-project/hello-world (master)
$ ll
total 12
-rw-r--r-- 1 aweso 197609  134 Aug 19 13:29 Dockerfile
-rw-r--r-- 1 aweso 197609  274 Aug 19 13:29 README.md
-rw-r--r-- 1 aweso 197609 6190 Aug 19 13:29 pom.xml
-rw-r--r-- 1 aweso 197609  507 Aug 19 13:29 regapp-deploy.yml
-rw-r--r-- 1 aweso 197609  209 Aug 19 13:29 regapp-service.yml
drwxr-xr-x 1 aweso 197609    0 Aug 19 13:29 server/
drwxr-xr-x 1 aweso 197609    0 Aug 19 13:29 webapp/

aweso@RAJIV-XPS17 MINGW64 /d/arena-devops-project/hello-world (master)
$ cd webapp/
pom.xml  src/

aweso@RAJIV-XPS17 MINGW64 /d/arena-devops-project/hello-world (master)
$ cd webapp/src/main/webapp/

aweso@RAJIV-XPS17 MINGW64 /d/arena-devops-project/hello-world/webapp/src/main/webapp (master)
$ ll
total 4
drwxr-xr-x 1 aweso 197609    0 Aug 19 13:29 WEB-INF/
-rw-r--r-- 1 aweso 197609 1345 Aug 19 13:29 index.jsp

aweso@RAJIV-XPS17 MINGW64 /d/arena-devops-project/hello-world/webapp/src/main/webapp (master)
$ |
```

#git add .

```
#git commit -m “edit index file”

#git push origin master
```

Open the index page and edit it and push it git and then we can check it git and see page is updated and if we brows the page form tomcat we can see the page it not updated and for that we need to create another build



Not secure | 192.168.30.142:8080/webapp/

ভূমি উন্নয়ন কর | Dhaka North City C... | ping form outside | subnetmask CIDR | iClo

New user Register Final update by Rajiv

Please fill in this form to create an account.

Enter Name Enter Full Name
Enter mobile Enter moible number
Enter Email Enter Email
Password Enter Password
Repeat Password Repeat Password

By creating an account you agree to our Terms & Privacy.

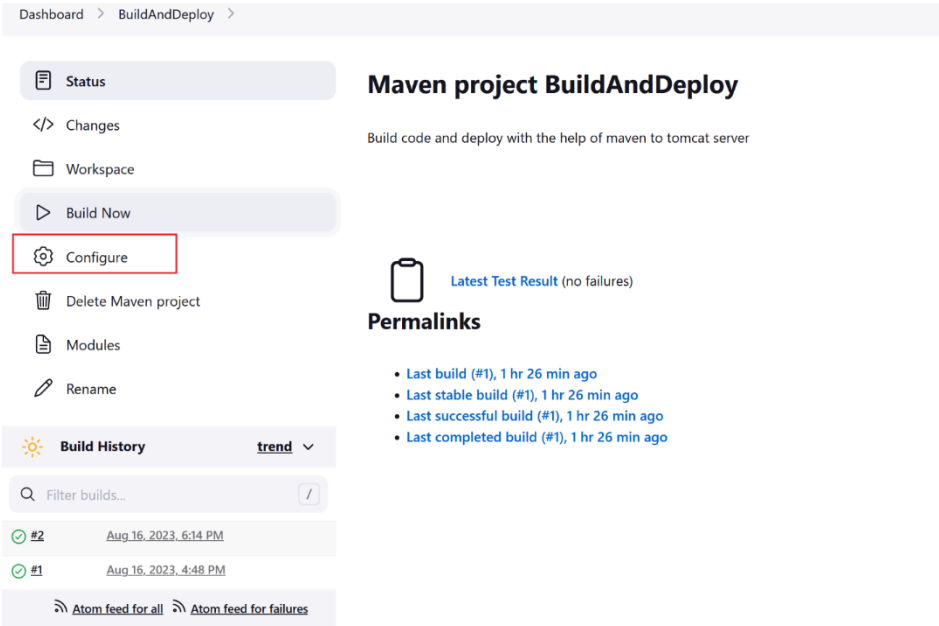
Register

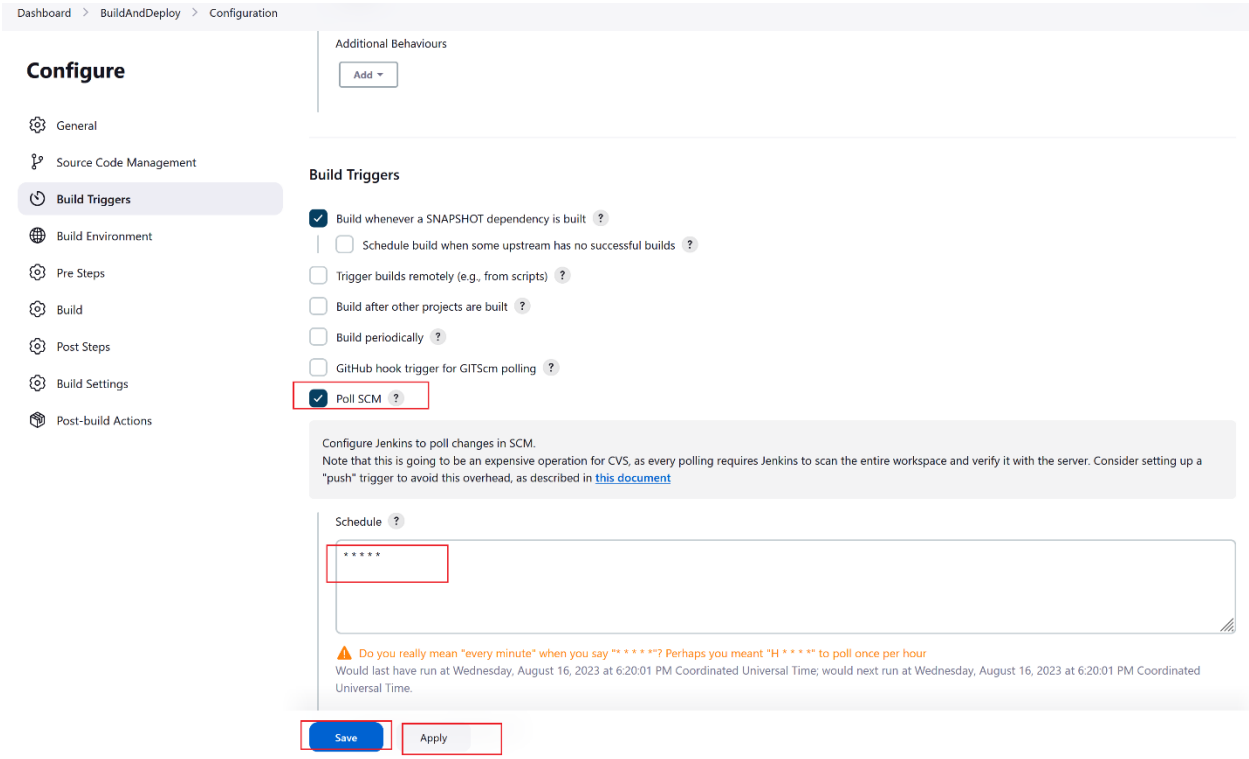
Already have an account? Sign in.

Thankyou, Happy Learning

test7

```
Now we want to do the things automatically so when we edit any code in our local pc
and push it to git hub change will be done automatically we don’t need to generate a
new build
```





Now edit the index file and push it to git hub and we can see the change in tomcat browser

```
aweso@RAJIV-XPS17 MINGW64 /d/arena-devops-project/hello-world/webapp/src/main/webapp (master)
$ vi index.jsp

aweso@RAJIV-XPS17 MINGW64 /d/arena-devops-project/hello-world/webapp/src/main/webapp (master)
$ git add .

aweso@RAJIV-XPS17 MINGW64 /d/arena-devops-project/hello-world/webapp/src/main/webapp (master)
$ git commit -m "edit index page"
[master 873066c] edit index page
 1 file changed, 1 insertion(+), 1 deletion(-)

aweso@RAJIV-XPS17 MINGW64 /d/arena-devops-project/hello-world/webapp/src/main/webapp (master)
$ git push origin master
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 534 bytes | 534.00 KiB/s, done.
Total 7 (delta 2), reused 4 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/rajivsiddiqui/hello-world.git
   17da60c..873066c  master -> master

aweso@RAJIV-XPS17 MINGW64 /d/arena-devops-project/hello-world/webapp/src/main/webapp (master)
$
```

Jenkins

Search (CTR

Dashboard > BuildAndDeployCodeToTomcat >

- Status
- Changes
- Workspace
- Build Now
- Configure
- Delete Maven project
- Modules
- Git Polling Log
- Rename

Build History trend

Filter builds... /

#3
(pending—In the quiet period. Expires in 4.4 sec)

#2
Aug 19, 2023, 7:38 AM

#1
Aug 19, 2023, 7:14 AM

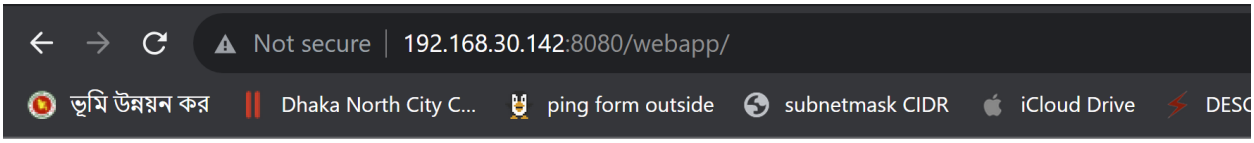
Maven project BuildAndDeployCodeToTomcat

build and deploy code to tomcat server with the help of maven

Latest Test Result (no failures)

Permalinks

- Last build (#2), 10 min ago
- Last stable build (#2), 10 min ago
- Last successful build (#2), 10 min ago
- Last completed build (#2), 10 min ago



New user Register Final update by Rajiv Siddiqui

Please fill in this form to create an account.

Enter Name Enter Full Name
Enter mobile Enter moible number
Enter Email Enter Email
Password Enter Password
Repeat Password Repeat Password

By creating an account you agree to our Terms & Privacy.

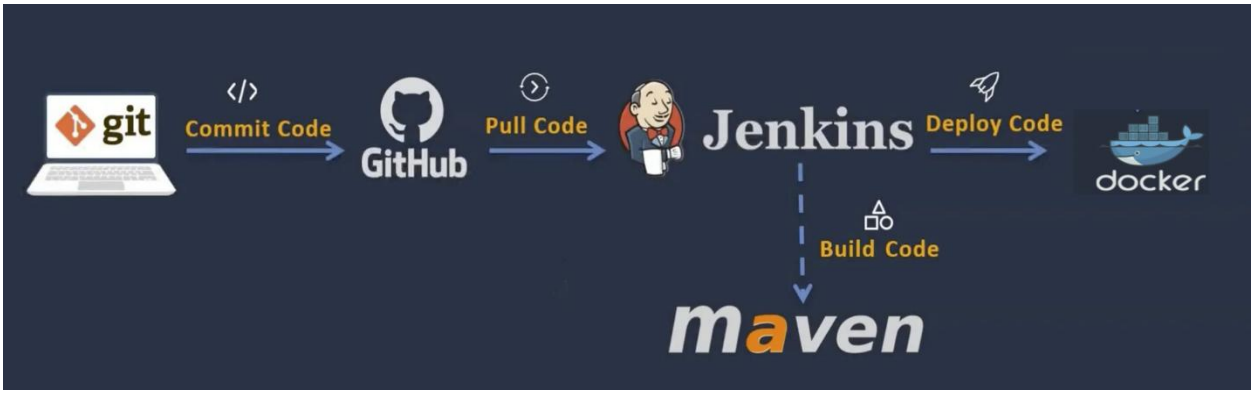
Register

Already have an account? Sign in.

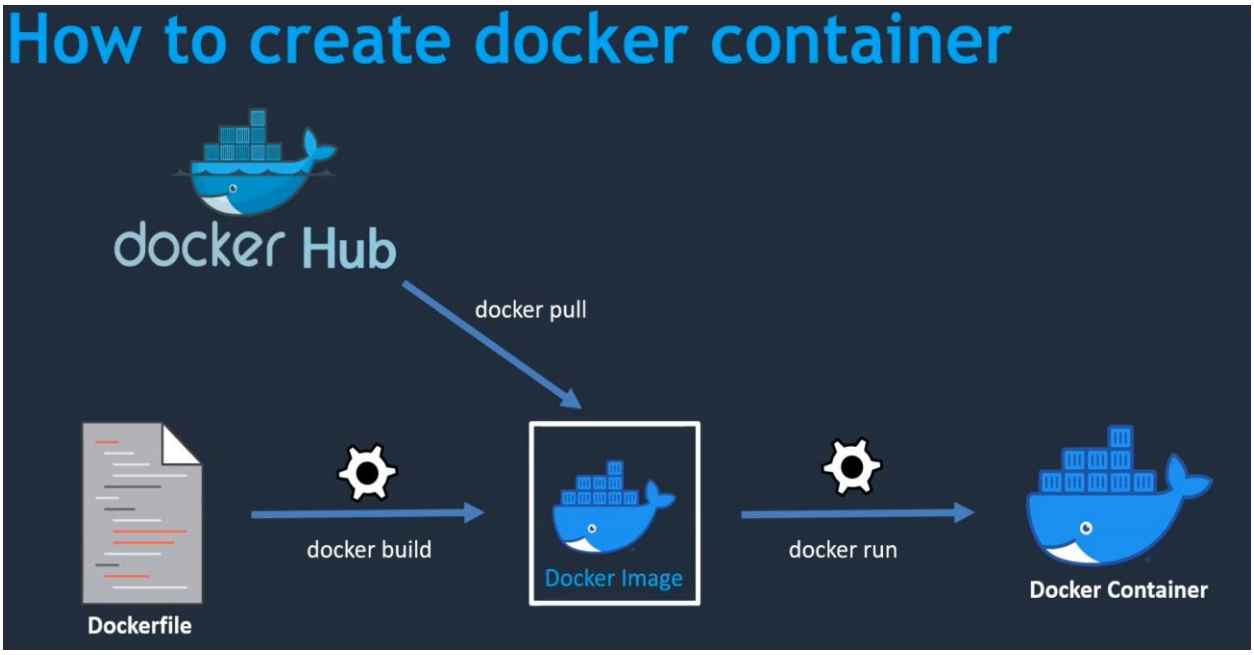
Thankyou, Happy Learning

test7

Now we deply the code to docker



1. Create a VM / EC2
2. Install docker host
3. Start docker service
4. Run tomcat container
5. Create a image by using docker file




```
#docker run -d --name tomcat1 -p 8081:8080 tomcat
```

We brows http://192.168.30.139:8081 and the page is not showing properly

```
#docker ps
#docker exec -it 5555 /bin/bash
```

```
After login the docker container run the command
# cp webapps.dist/* -R webapps
```

Now try to browse the page and the page is showing now

Now it we stop the container and run it again the copy will be gone so we need to create our own image my using Dockerfile

#vi Dockerfile

```
FROM tomcat:latest
RUN cp -R  /usr/local/tomcat/webapps.dist/*  /usr/local/tomcat/webapps
COPY ./*.war /usr/local/tomcat/webapps
```

```
#vi Dockerfile
#docker build -t demotomcat .
#docker image ls |grep -i demo
#docker container run -d --name demo-tomcat -p 8082:8080 demotomcat
```

We can also use docker file from centos base image

```
FROM centos:7
RUN yum install java -y
RUN mkdir /opt/tomcat/
WORKDIR /opt/tomcat
ADD https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.84/bin/apache-tomcat-9.0.84.tar.gz
.
RUN tar xvfz apache*.tar.gz
RUN mv apache-tomcat-9.0.84/* /opt/tomcat
EXPOSE 8080
CMD ["/opt/tomcat/bin/catalina.sh", "run"]
```

Integrate Docker with jenkins

- Create a dockeradmin user
- install "publish over ssh plugin"
- Add dockerhost to Jenkins "Configure system"

Create dockeradmin user and add it docker group

#adduser dockeradmin

#useradd dockeradmin = using this command, user directory will not be created don't use this

#passwd dockeradmin

#id dockeradmin

```
#sudo usermod -aG docker dockeradmin
#id dockeradmin
```

make sure dockeradmin user can run docker image if not then use the following command

Run docker command without sudo

$ sudo groupadd docker

$ sudo gpasswd -a $USER docker

OR

$ sudo usermod -aG docker dockeradmin

$ sudo service docker restart

$ su – dockeradmin

#docker ps - make sure dockeradmin user can run this command

Enable password based authenticaton on aws ec2

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

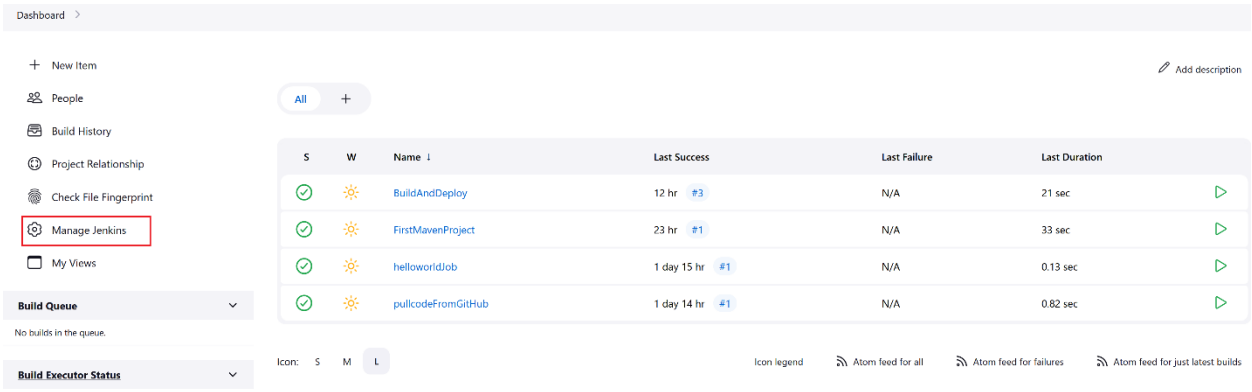
```
root@ip-172-31-46-196:/home/ec2-user/docker-tomcat
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to n
PasswordAuthentication yes        uncomment it
#PermitEmptyPasswords no
#PasswordAuthentication no        comment it by #

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

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

# GSSAPI options
GSSAPIAuthentication yes
```

```
# service sshd reload
```

Now try to login docker host by username and password
Username is dockeradmin and password is 1234

```
#ssh dockeradmin@54.158.215.235
```

Install "publish over Jenkins" in Jenkins



Dashboard > Manage Jenkins

- New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

Manage Jenkins

Search settings /

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`. [Dismiss]

Please provide an accurate value in Jenkins configuration.

System Configuration

System
Configure global settings and paths.

Tools
Configure tools, their locations and automatic installers.

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

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

Security

Dashboard > Manage Jenkins > Plugins

- Updates
- Available plugins 1
- Installed plugins
- Advanced settings
- Download progress

Plugins

publish over ssh 2

Install	Name ↓	Released
☑	Publish Over SSH 1.25 Artifact Uploaders Build Tools Send build artifacts over SSH 3	1 mo 11 days ago

4 [Install without restart] [Download now and install after restart] Update information obtained: 1 day 17 hr ago [Check now]

Dashboard >

- New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- My Views

Build Queue

No builds in the queue.

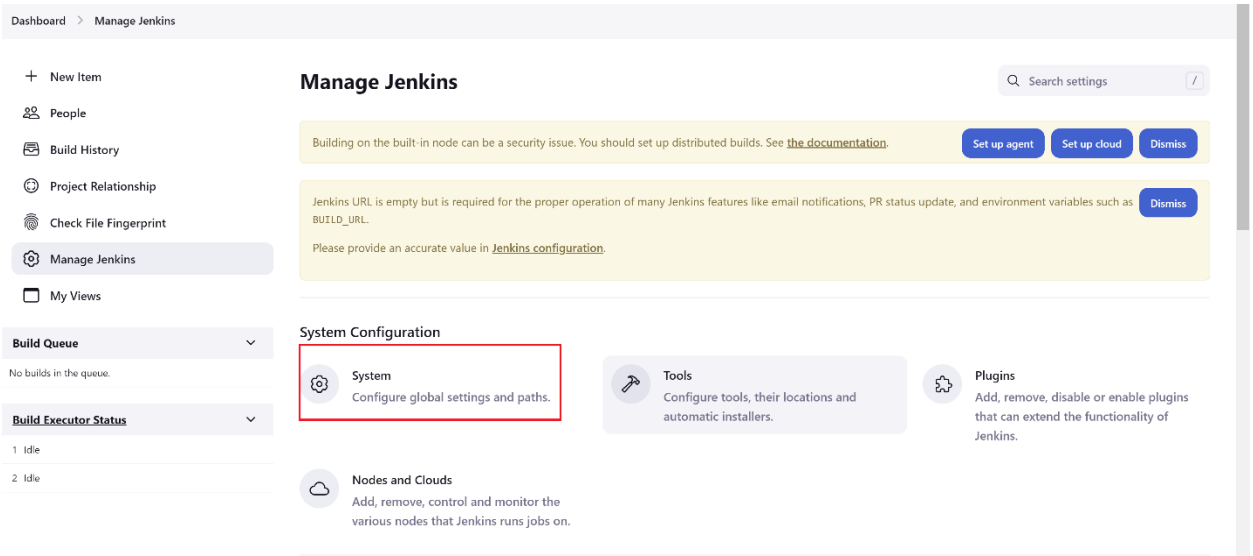
Build Executor Status

Add description

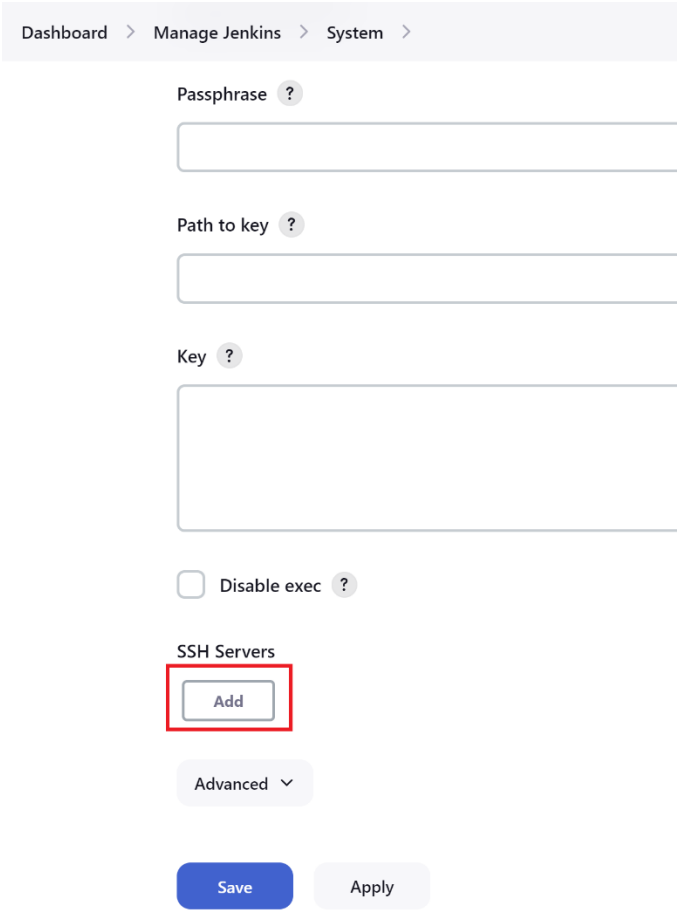
All +

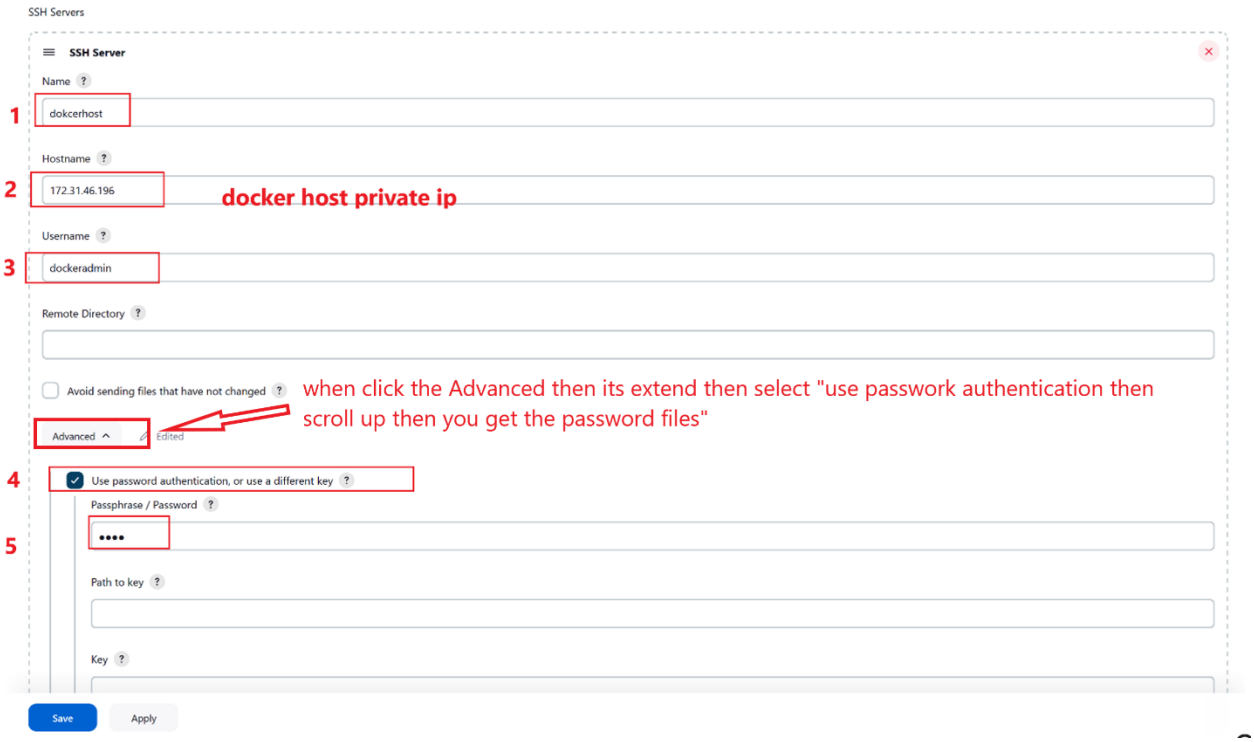
S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☼	BuildAndDeploy	12 hr #3	N/A	21 sec
✓	☼	FirstMavenProject	23 hr #1	N/A	33 sec
✓	☼	helloworldJob	1 day 15 hr #1	N/A	0.13 sec
✓	☼	pullcodeFromGitHub	1 day 14 hr #1	N/A	0.82 sec

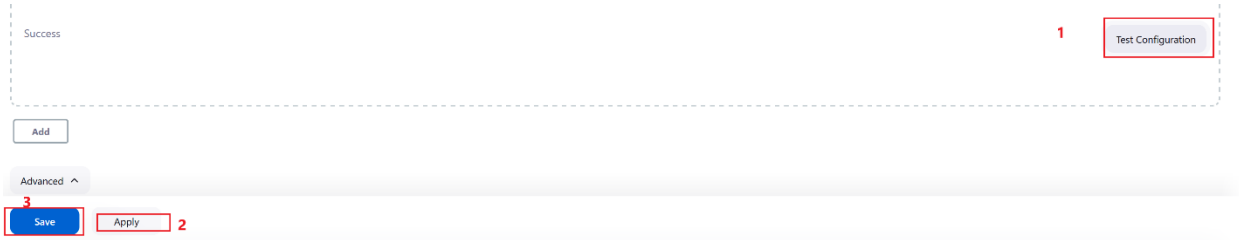
Icon: S M L Icon legend Atom feed for all Atom feed for failures Atom feed for just latest builds


Dashboard > Manage Jenkins
New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Manage Jenkins
Search settings
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.
Dismiss
Please provide an accurate value in Jenkins configuration.
System Configuration
System
Configure global settings and paths.
Tools
Configure tools, their locations and automatic installers.
Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

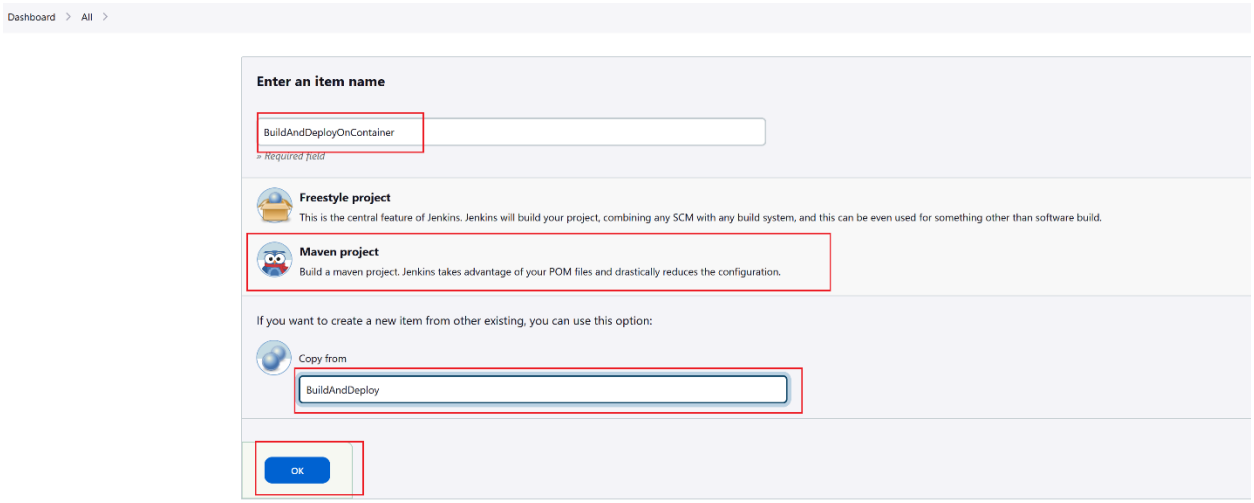
Scroll down

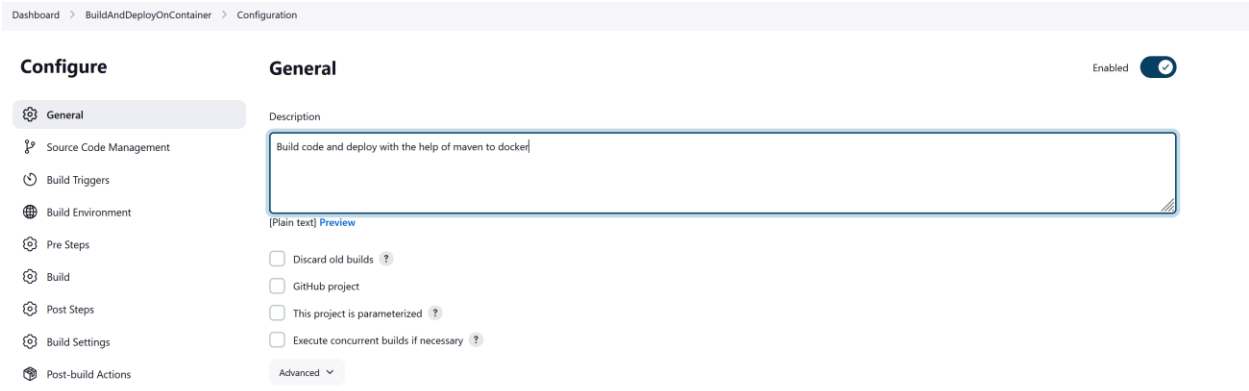

Dashboard > Manage Jenkins > System >
Passphrase
Path to key
Key
Disable exec
SSH Servers
Add
Advanced
Save
Apply





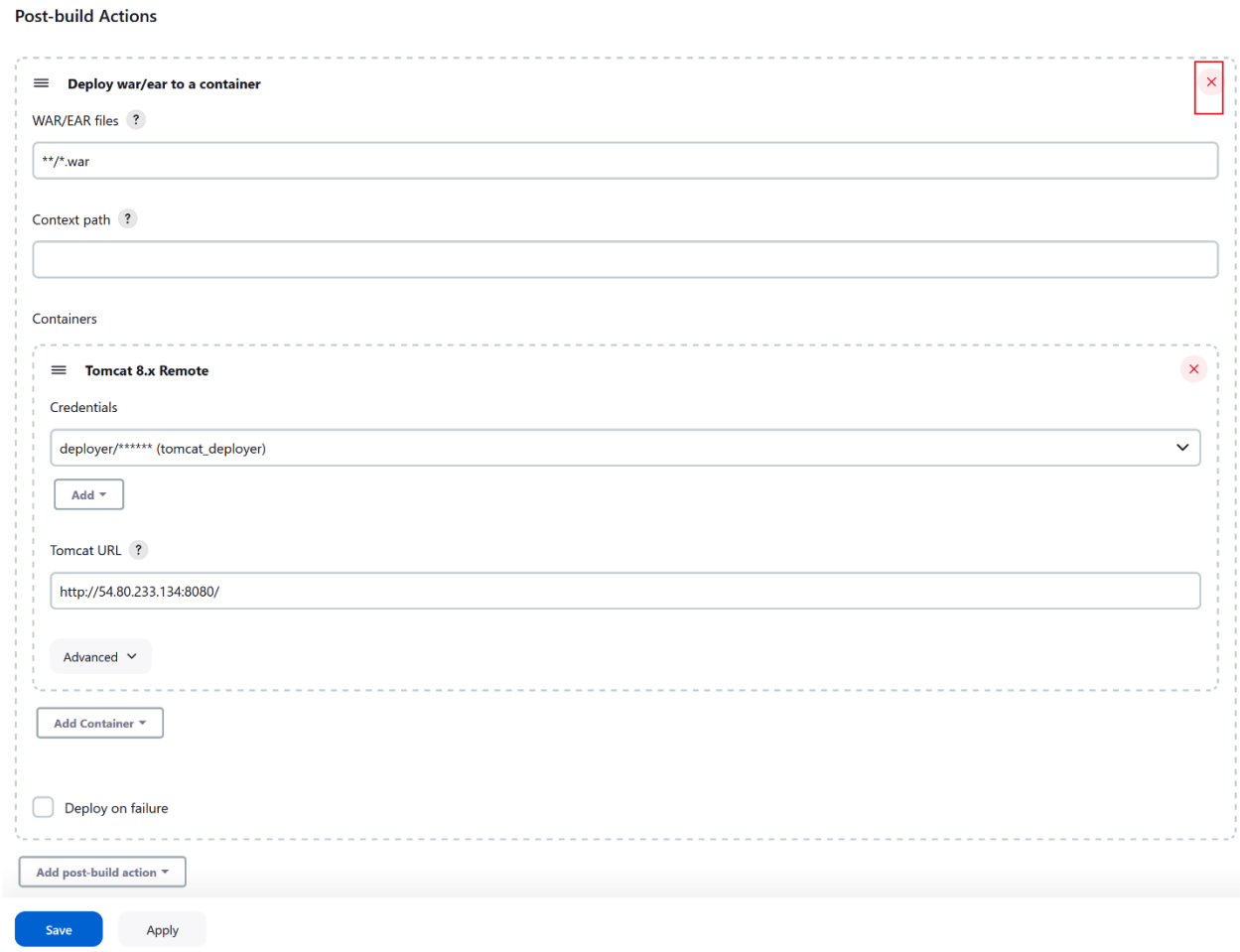
Now deploy code to docker host





Scroll down

Now we don't need to build and deploy to tomcat so we need to cross this one



Dashboard > BuildAndDeployOnContainer > Configuration

Configure

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

Goals and options ?

clean install

Advanced

Post Steps

- ○ Run only if build succeeds
- ○ Run only if build succeeds or is unstable
- ● Run regardless of build result

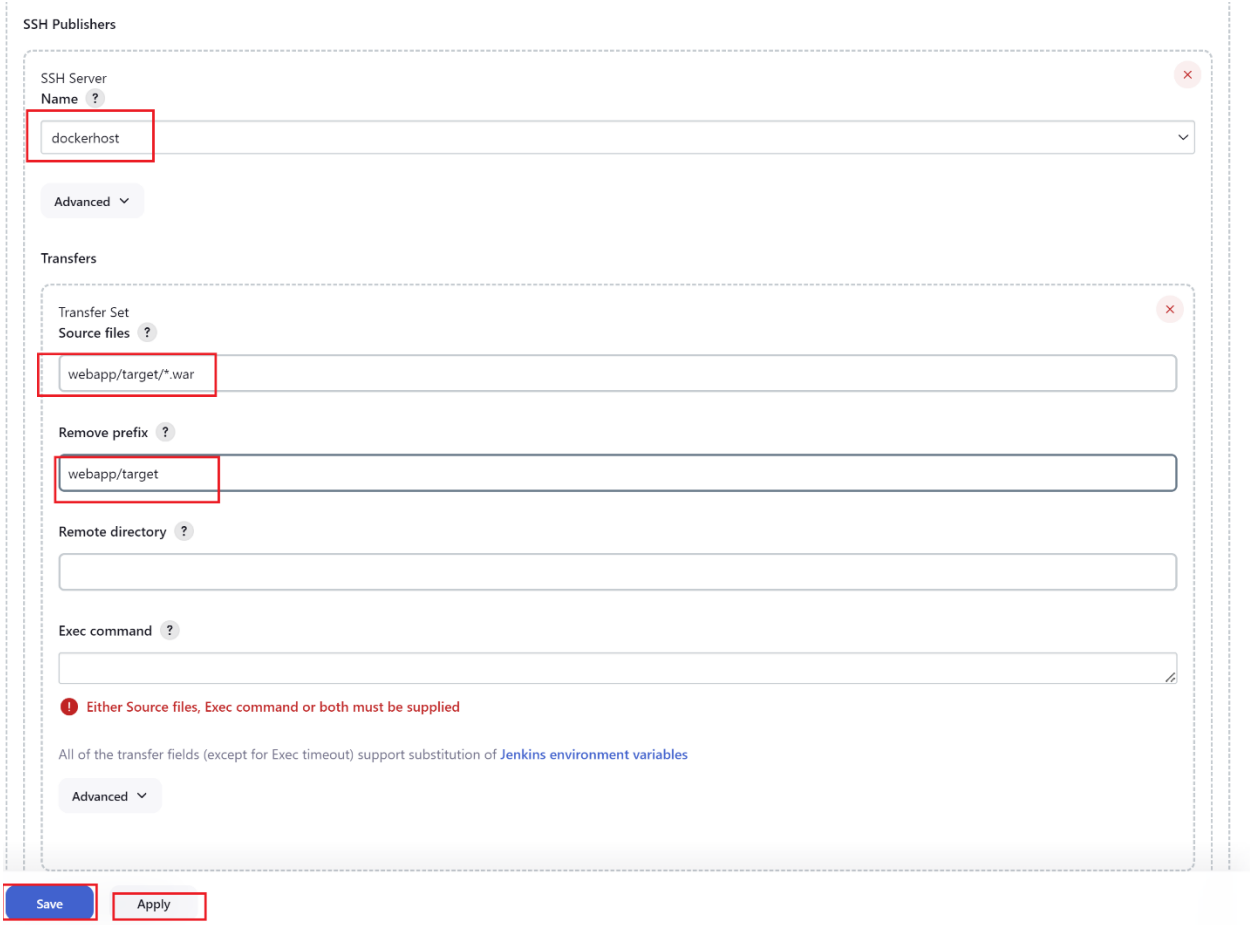
Should the post-build steps run only for successful builds, etc.

Filter

- Aggregate downstream test results
- Archive the artifacts
- Build other projects
- Deploy artifacts to Maven repository
- Record fingerprints of files to track usage
- Git Publisher
- Deploy war/ear to a container
- 2 Send build artifacts over SSH
- Set GitHub commit status (universal)
- Set build status on GitHub commit [deprecated]

1 Add post-build action

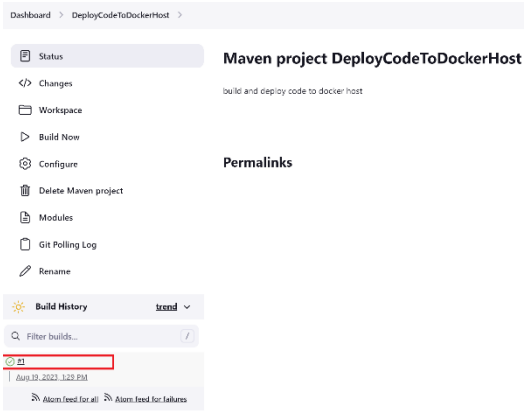
Save Apply



Now automatically generate the build after click the save button and we can also see it from dockerhost and it will take sometimes so need to wait.

#cd /home/dockeradmin

#ls |grep -i web*



<u>Create a directory on dockerhost server(192.168.30.142)</u>

#mkdir /opt/docker

Give the directory ownership to dockeradmin

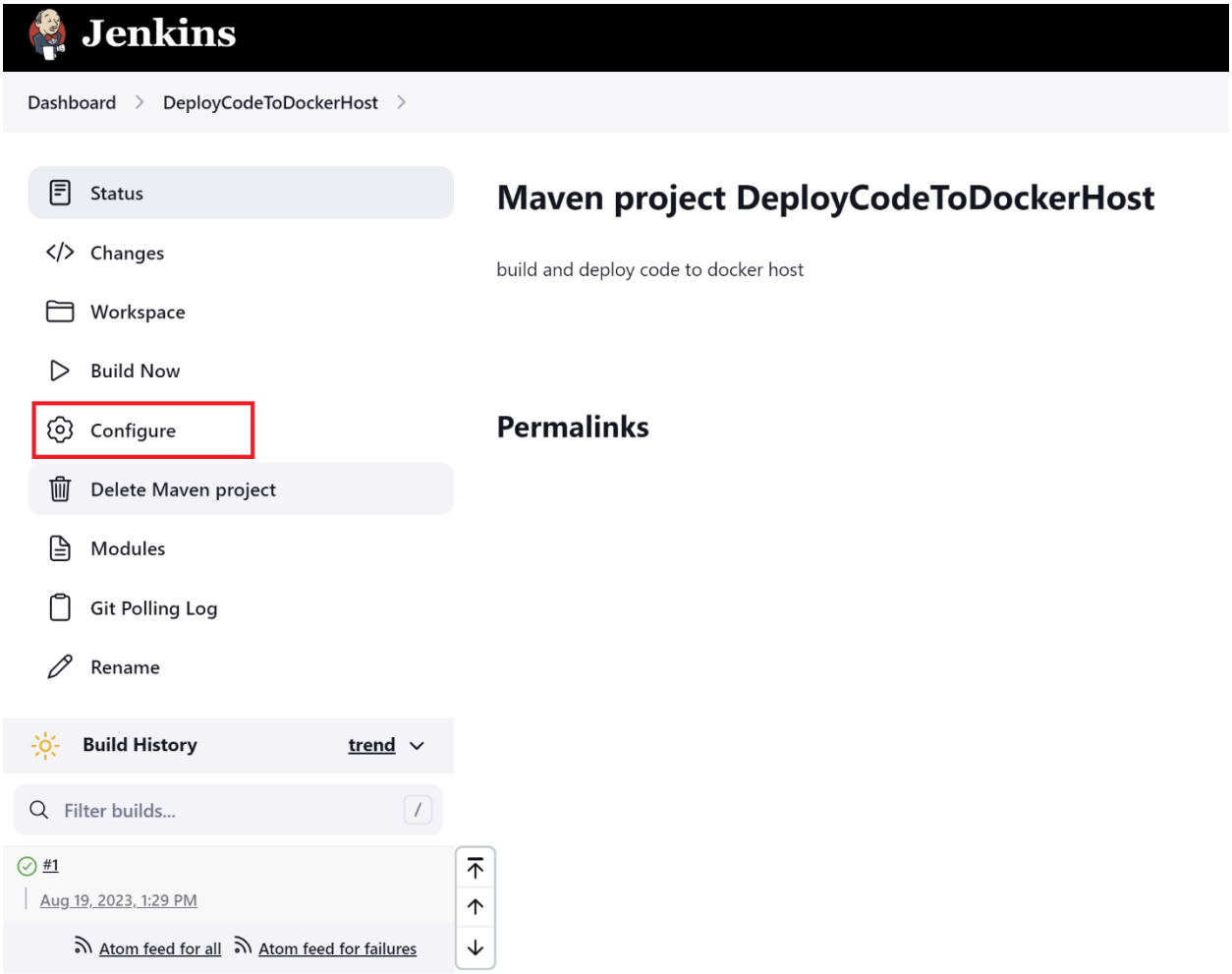
```
#chown -R rajiv:rajiv docker
```

Now create a docker file on this directory and give the ownership to rajiv
#vi Dockerfile

```
FROM tomcat:latest
RUN cp -R /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps
```

```
# chown -R rajiv:rajiv Dockerfile
#ls
#ls -ld
```

Now set artifact file location in Jenkins and set it to **//opt//rajiv**



Dashboard > BuildAndDeployCodeToDocker > Configuration

Configure

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

SSH Server

Name ?

dockerhost

Advanced

Transfers

Transfer Set

Source files ?

webapp/target/*.war

Remove prefix ?

webapp/target

Remote directory ?

//opt//docker

Exec command ?

All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables

Advanced

Add Transfer Set

Save Apply

Now build now again

Jenkins

Dashboard > DeployCodeToDockerHost >

- Status
- Changes
- Workspace
- Build Now
- Configure
- Delete Maven project
- Modules
- Git Polling Log
- Rename

Build History trend

Filter builds... /

#2

Aug 19, 2023, 1:43 PM

#1

Aug 19, 2023, 1:29 PM

Atom feed for all Atom feed for failures

Maven project DeployCodeToDock

build and deploy code to docker host

Latest Test Result (no failures)

Permalinks

- Last build (#1), 14 min ago
- Last stable build (#1), 14 min ago
- Last successful build (#1), 14 min ago
- Last completed build (#1), 14 min ago

Now go to the docker host and check the location /opt/docker and we see the file

```
root@server-A: /opt/docker
root@server-A:/opt/docker# ll
total 16
drwxr-xr-x 2 rajiv rajiv 4096 আগস্ট     19 19:43 ./
drwxr-xr-x 4 root  root  4096 আগস্ট     19 19:37 ../
-rw-r--r-- 1 rajiv rajiv   88 আগস্ট     19 19:37 Dockerfile
-rw-rw-r-- 1 rajiv rajiv 2374 আগস্ট     19 19:43 webapp.war
root@server-A:/opt/docker#
```

Now edit the docker file and add a line to copy this war file

#vi Dockerfile

```
FROM tomcat:latest
RUN cp -R /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps
COPY ./*.war /usr/local/tomcat/webapps
```

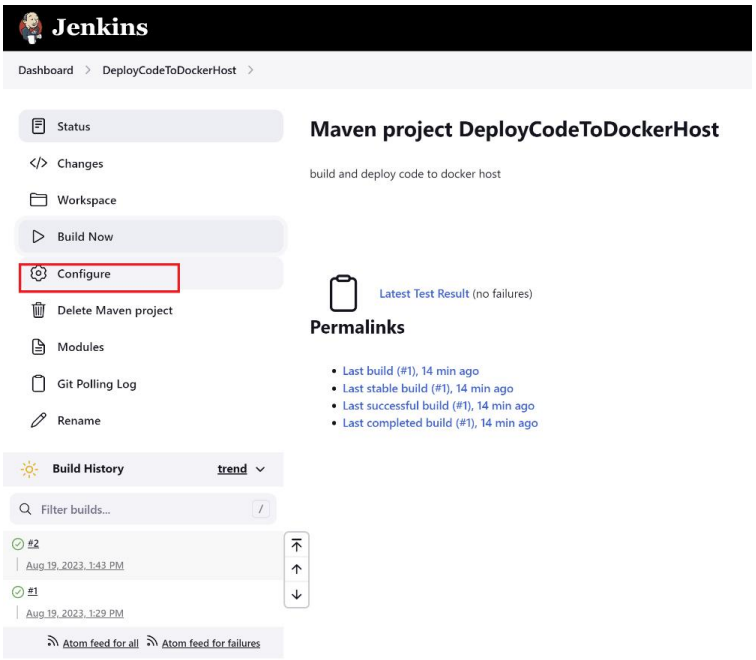
Now create a image by using the docker file

#cd /opt/docker

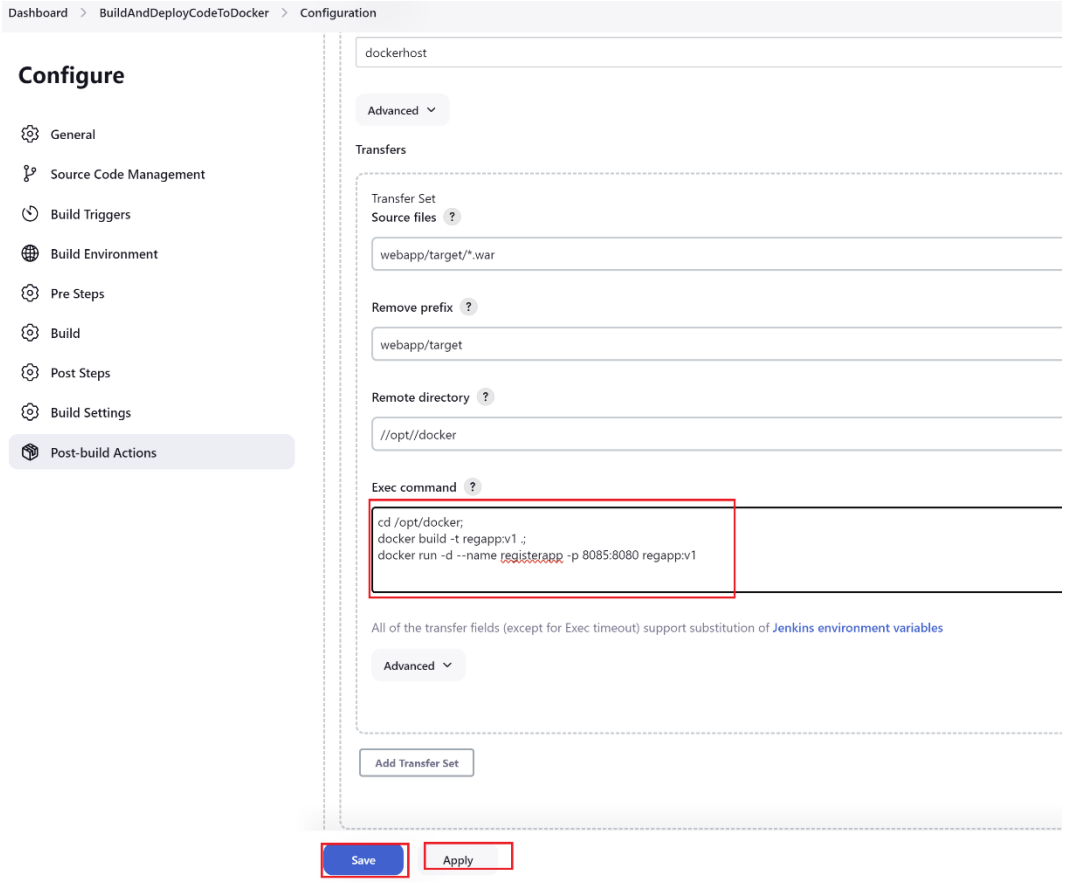
```
#docker build -t tomcat:v1 .
#docker image ls | grep -i tom*
#docker container run -d --name tomv1 -p 8083:8080 tomcat:v1
#docker ps
```

Now we can see the page http://192.168.30.139:8083

Now run create this container automatically by Jenkins



Now edit configure and add some code




```
cd /opt/docker;
docker build -t regapp:v1 .;
docker run -d --name registerapp -p 8085:8080 regapp:v1
```

Dashboard > DeployCodeToDockerHost > Configuration

Configure

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

SSH Server
Name ?

dockerhost

Advanced

Transfers

Transfer Set
Source files ?

webapp/target/*.war

Remove prefix ?

webapp/target

Remote directory ?

//opt//docker

Exec command ?

```
cd /opt/docker;
docker build -t regapp:v1 .;
docker run -d --name registerapp -p 8085:8080 regapp:v1
```

All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables

Save Apply

Now stop and remove all running container
now build the job

Jenkins

Dashboard > DeployCodeToDockerHost >

- Status
- Changes
- Workspace
- Build Now
- Configure
- Delete Maven project
- Modules
- Git Polling Log
- Rename

Build History trend

Filter builds... /

#2
Aug 19, 2023, 1:43 PM

#1
Aug 19, 2023, 1:29 PM

Atom feed for all Atom feed for failures

Maven project DeployCodeToDockerHost

build and deploy code to docker host

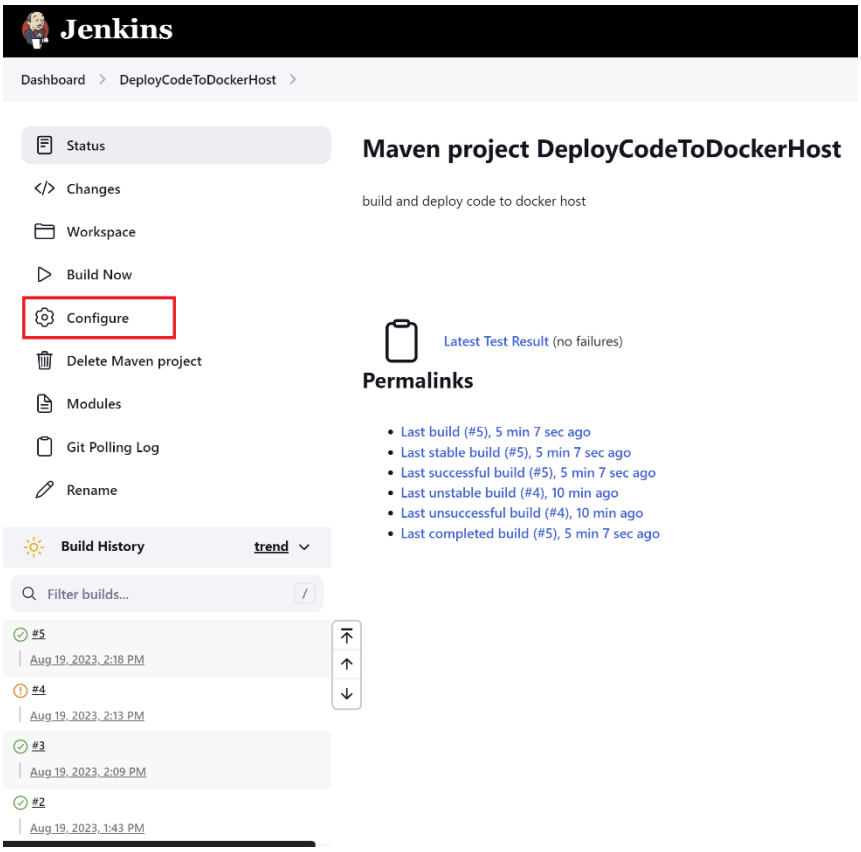
Latest Test Result (no failures)

Permalinks

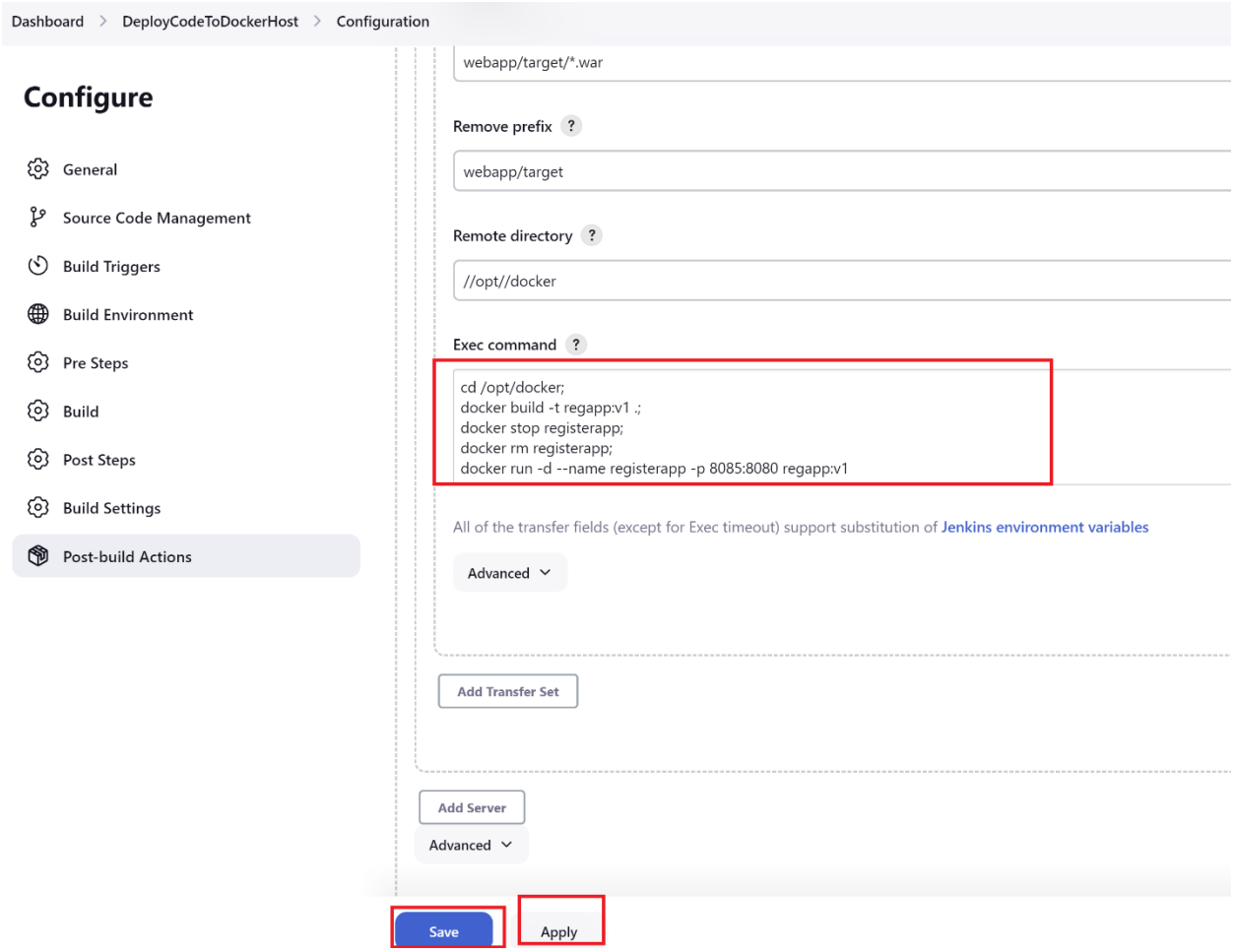
- Last build (#2), 24 min ago
- Last stable build (#2), 24 min ago
- Last successful build (#2), 24 min ago
- Last completed build (#2), 24 min ago

Now we can see the page

Now go to your local pc and edit the index file and push it
now new build will generate but it will be failed because the same name container already available so for that we need to edit the configuration and 2 more lines

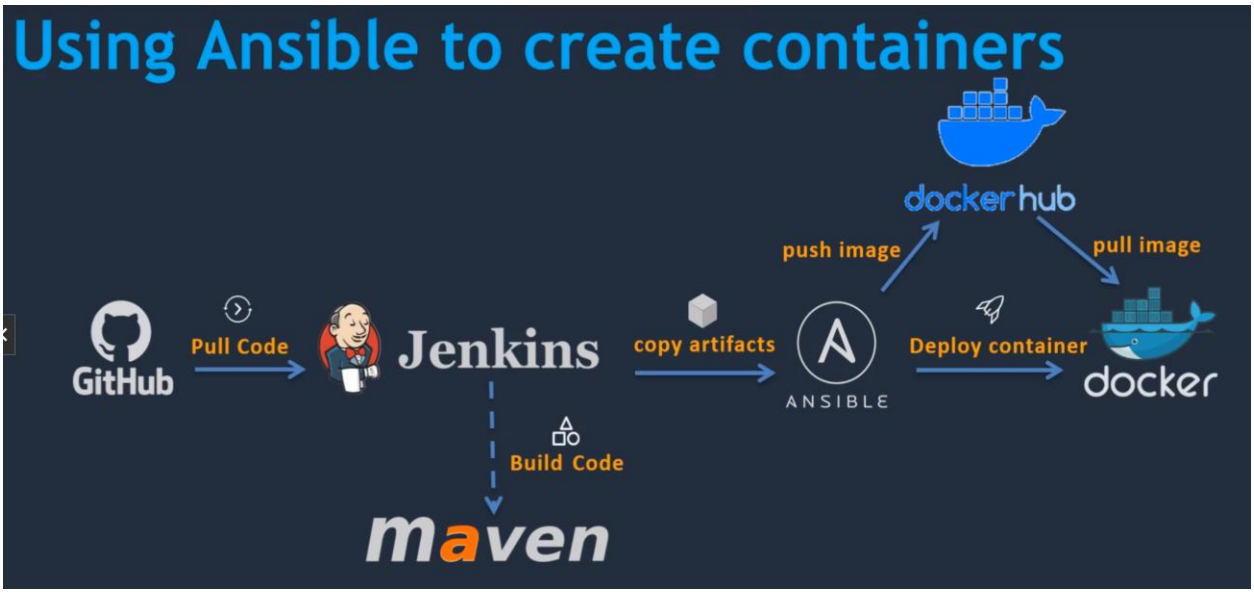



```
cd /opt/docker;
docker build -t regapp:v1 .;
docker container stop registerapp;
docker container rm registerapp;
docker run -d --name registerapp -p 8085:8080 regapp:v1
```



Now we build will be generate successfully and it will delete old container and then run new container so we can see the web page

Integrate Ansible with CI/CD



setup ec2

setup hostname

create ansamdin user

add user to sudoers file

generate ssh keys

enbale password based login

install ansible

on ansible node

- Create a user
- Enable sudo privileges for this user
- Enable password-based authentication.
- Login to ansadmin user
- Create ssh-key
- add to host file
  - copy ssh key
  - Test the connection

on docker host node need to do

- create a user ansadmin
- add this user to sudoers fiels
- enable password based login if ec2

in ansible server

#sudo su
#adduser ansadmin
#passwd ansadmin
#visudo
#vi /etc/sudoers
add the line in this file

```
ansadmin ALL=(ALL:ALL) NOPASSWD:ALL
```

change the host name

#sudo su
#vi /etc/hostname
give the hostname ansible-server
#init 6

Create a user

#sudo su
#useradd ansadmin
#passwd ansadmin

give sudo previlidge

#sudo su
#visduo
or
#sudo vi /etc/sudoers
add the following line

```
ansadmin ALL=(ALL:ALL) NOPASSWD: ALL
```

```
# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
ansadmin ALL=(ALL:ALL) NOPASSWD: ALL
# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
"/etc/sudoers" [readonly] 54L, 1707B
```

if user nano then for edit the file we need the following command.

#ctrl +x

#y

#enter

<u>Enable password based authentication if it is AWS Ec2</u>

#sudo su

```
#vi /etc/ssh/sshd_config
uncomment PasswordAuthentication yes
Comment PasswordAuthentication no

# service sshd reload
```

```
root@ip-172-31-46-196:/home/ec2-user/docker-tomcat
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to n
PasswordAuthentication yes      uncomment it
#PermitEmptyPasswords no
#PasswordAuthentication no      comment it by #

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

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

# GSSAPI options
GSSAPIAuthentication yes
```

Create private and public key for ansadmin user

Open a new terminal and login to user ansadmin then run the following command

```
# ssh-keygen
```

Install Ansible

```
#sudo su
#sudo apt install ansible        install ansible core
check the version
#ansible --version
#python3 --version
```

on docker host node

- create a user ansadmin
- add this user to sudoers fiels
- enable password based login if ec2

create host file in ansibe server

ansible host/inventory file

vi /etc/ansible/hosts

ansible config file
vi /etc/ansible/ansible.cfg

Now in docker host need to do the following things

change the host name
#sudo su
#vi /etc/hostname
give the hostname docker
#init 6
Create a user
#sudo su
#useradd ansadmin
#passwd ansadmin

give sudo previlidge
#sudo su
#visduo
or
#sudo vi /etc/sudoers
add the following line

```
ansadmin ALL=(ALL:ALL) NOPASSWD: ALL
```

#ctrl +x
#y
#enter

Enable password based authentication if it is AWS Ec2
#sudo su

```
#vi /etc/ssh/ssh_config
uncomment PasswordAuthentication yes
Comment PasswordAuthentication no

# service sshd reload
```

Check password based authentication
```
#grep Password /etc/ssh/ssh_config
```

```
root@ip-172-31-46-196:/home/ec2-user/docker-tomcat
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to n
PasswordAuthentication yes        uncomment it
#PermitEmptyPasswords no
#PasswordAuthentication no        comment it by #

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

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

# GSSAPI options
GSSAPIAuthentication yes
```

add the docker ip in ansible inventory (host) file

```
#sudo su
#vi /etc/ansible/hosts
192.168.30.139
```

Copy the ansible serber public key (ansadmin user) to docker host

```
Switch to ansadmin user from root
#su - ansadmin
#sudo ansadmin
#cd /.ssh
#ll
#ssh-copy-id <docker host ip>
#ssh-copy-id 192.168.30.144
```

Check the connection from ansible to docker

Login to ansible server and make user you are login with right user which is ansadmin

#ansible all -m ping

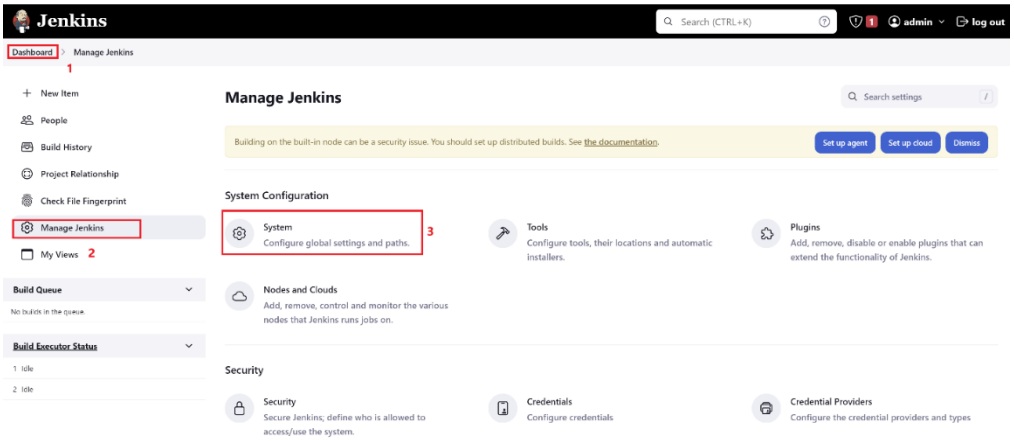
```
rajiv@ansible-master1: ~
rajiv@ansible-master1:~$ ansible all -m ping
192.168.30.139 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
rajiv@ansible-master1:~$
```

#ansible all -m command -a uptime

```
rajiv@ansible-master1:~$ ansible all -m command -a uptime
192.168.30.139 | CHANGED | rc=0 >>
 11:47:22 up 14:28,  7 users,  load average: 0.25, 0.20, 0.18
rajiv@ansible-master1:~$
```

Now we sure that ansible server is communicate with docker server

Intigrte ansible with jenkins
Install ansible plugin to Jenkins



Dashboard > Manage Jenkins > System >

dockerhost

Hostname ?

192.168.30.139

Username ?

rajiv

Remote Directory ?

Avoid sending files that have not changed ?

Advanced Edited

Add

Advanced

Save Apply

Dashboard > Manage Jenkins > System >

SSH Server

Name ?

ansible-server 1

Hostname ?

192.168.30.144 2

Username ?

rajiv 3

Remote Directory ?

Avoid sending files that have not changed ?

4 Advanced Edited

5 Use password authentication, or use a different key ?

Passphrase / Password ?

6 ••

Scroll down and test the connection

Proxy password

Success 2

1 Test Configuration

Add

Advanced

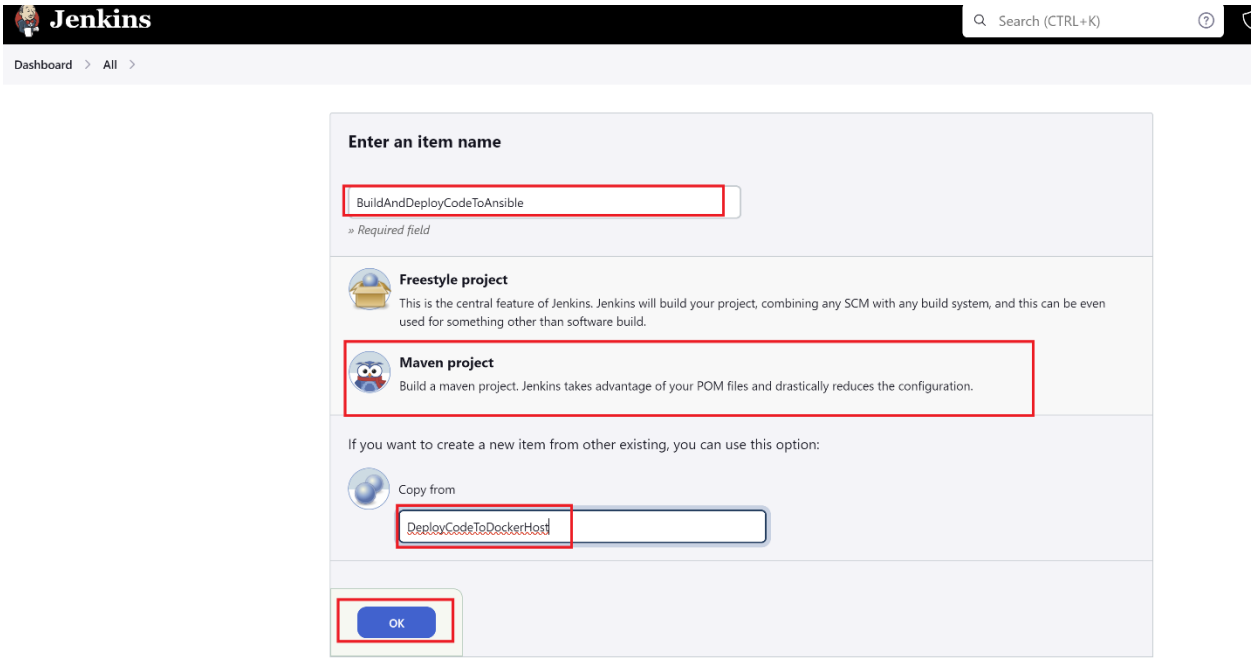
4 Save Apply 3

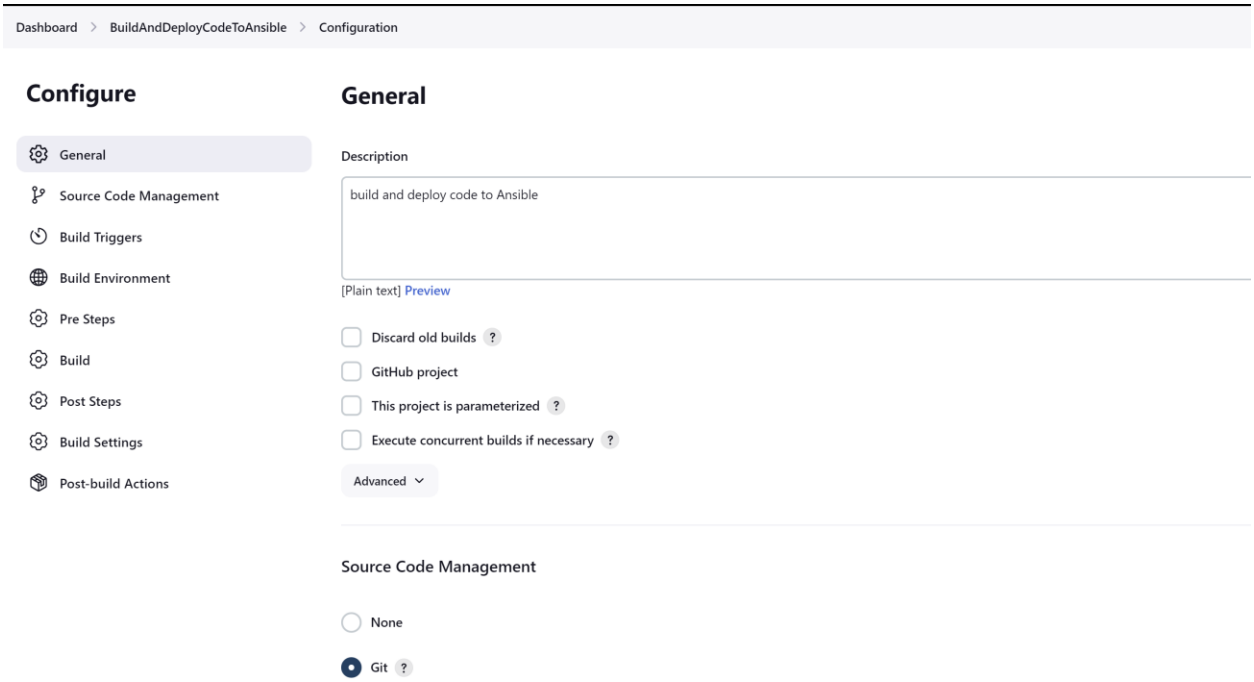
Now create a build to Ansible

first create a directory in ansible server

#cd /opt

```
# mkdir /opt/docker
#ll
change the ownership of docker folder
# sudo chown rajiv:rajiv docker
```





Dashboard > BuildAndDeployCodeToAnsible > Configuration

Configure

- General
- Source Code Management
- 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 ?

uncehck this one for disable it and we enable it when need again

Configure

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

Post-build Actions

Send build artifacts over SSH ?

SSH Publishers

SSH Server

Name ?

1 ansible-server

Advanced

Transfers

Transfer Set

Source files ?

2 webapp/target/*.war

Remove prefix ?

3 webapp/target

Remote directory ?

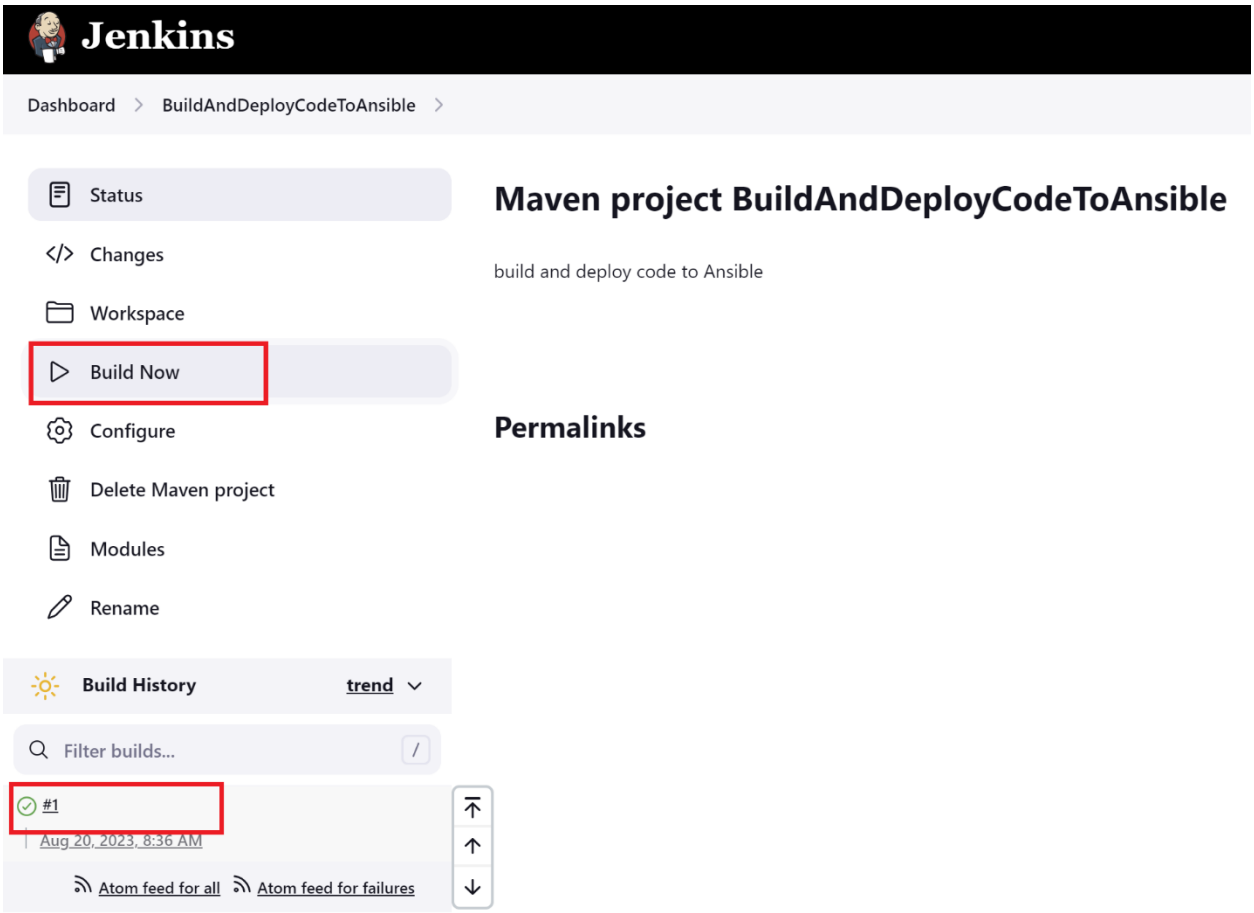
4 //opt//docker

Exec command ?

keep it blank

All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables

6 Save Apply 5



Now if we go to the ansible server /opt/docker location then we can see the artifact is created

```
root@ansible-master1:/opt# cd docker/
root@ansible-master1:/opt/docker# ll
total 8
drwxr-xr-x 2 rajiv rajiv 4096 Aug 20 08:19 ./
drwxr-xr-x 3 root  root  4096 Aug 20 08:19 ../
root@ansible-master1:/opt/docker# ll
total 12
drwxr-xr-x 2 rajiv rajiv 4096 Aug 20 08:36 ./
drwxr-xr-x 3 root  root  4096 Aug 20 08:19 ../
-rw-rw-r-- 1 rajiv rajiv 2383 Aug 20 08:36 webapp.war
root@ansible-master1:/opt/docker#
```

Now need to create a image to ansible server

First install docker on ansible server

#sudo su
install docker

#docker --version

now add the user to docker group
sudo usermod -aG docker ansadmin
#id ansadmin

Now check the docker server if not running then run ad enable it

in ansible server create same docker file which we create in docker host

#cd /opt/docker
#vi Dockerfile

```
FROM tomcat:latest
RUN cp -R /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps
COPY ./*.war /usr/local/tomcat/webapps
```

now chage the ownership of Dockerfile

```
#sudo chown ansadmin:ansadmin Dockerfile
#ll
```

```
#docker ps
if any permission issue then use
# sudo chmod 777 /var/run/docker.sock
```

Now build docker image

```
#cd /opt/docker
#docker build -t regapp:v1 .
```

Now run docker container in ansible server

#docker container run -d --name regappv1 -p 8081:8080 regapp:v1
#docker ps
now we can see the page from browser 192.168.30.144:8081

Edit the inventory file in ansible

#vi /etc/ansible/hosts

```
[dockerhost]
192.168.30.139

[ansible]
192.168.30.144
```

Now check
make user you are loged in as rajiv user

#ansible all -a uptime

```
rajiv@ansible-master1:~$ ansible all -m command -a uptime
192.168.30.144 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: rajiv@192.168.30.144: Permission denied (publickey,password).",
    "unreachable": true
}
192.168.30.139 | CHANGED | rc=0 >>
 15:39:13 up 18:20,  7 users,  load average: 0.15, 0.16, 0.17
rajiv@ansible-master1:~$
```

Now copy ssh key in ansible and copy its own ssh key

```
# ssh-copy-id 192.168.30.144
# ssh-copy-id localhost
```

#ansible all -m command -a uptime

```
rajiv@ansible-master1:~$ ansible all -m command -a uptime
192.168.30.139 | CHANGED | rc=0 >>
 15:41:32 up 18:22,  7 users,  load average: 0.19, 0.17, 0.18
192.168.30.144 | CHANGED | rc=0 >>
 09:41:32 up  5:25,  3 users,  load average: 0.27, 0.26, 0.25
rajiv@ansible-master1:~$ ^C
rajiv@ansible-master1:~$
```

Now create a playbook in ansible server in /opt/docker/regapp.yaml

```
---
- hosts: ansible

  tasks:
   - name: create docker image
     command: docker build -t regapp:latest .
     args:
      chdir: /opt/docker
```

Check the file

#ansible-playbook regapp.yaml --check

```
rajiv@ansible-master1:/opt/docker$ ll
total 16
drwxr-xr-x 2 rajiv rajiv 4096 Aug 20 08:54 ./
drwxr-xr-x 4 root  root  4096 Aug 20 08:45 ../
-rw-r--r-- 1 rajiv rajiv  127 Aug 20 08:54 Dockerfile
-rw-rw-r-- 1 rajiv rajiv 2383 Aug 20 08:36 webapp.war
rajiv@ansible-master1:/opt/docker$ vi regapp.yaml
rajiv@ansible-master1:/opt/docker$ ansible-playbook regapp.yaml --check

PLAY [ansible] ***************************************************************************************************

TASK [Gathering Facts] *******************************************************************************************
ok: [192.168.30.144]

TASK [create docker image] ***************************************************************************************
skipping: [192.168.30.144]

PLAY RECAP *******************************************************************************************************
192.168.30.144             : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Now before run the paybook check the available image we have

#docker image ls

Run the paybook

#ansible-playbook regapp.yaml

#sudo docker image ls

```
rajiv@ansible-master1:/opt/docker$ sudo docker image ls
[sudo] password for rajiv:
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
regapp       v1        8e5fd62573e9   54 minutes ago   430MB
rajiv@ansible-master1:/opt/docker$ ansible-playbook regapp.yaml

PLAY [ansible] ***********************************************************************************************************************

TASK [Gathering Facts] ***************************************************************************************************************
ok: [192.168.30.144]

TASK [create docker image] ***********************************************************************************************************
changed: [192.168.30.144]

PLAY RECAP ***************************************************************************************************************************
192.168.30.144             : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

rajiv@ansible-master1:/opt/docker$ docker image ls
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocke
/json": dial unix /var/run/docker.sock: connect: permission denied
rajiv@ansible-master1:/opt/docker$ sudo docker image ls
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
regapp       latest    8e5fd62573e9   57 minutes ago   430MB
regapp       v1        8e5fd62573e9   57 minutes ago   430MB
rajiv@ansible-master1:/opt/docker$
```

Copy this docker image to docker hub

#docker image ls

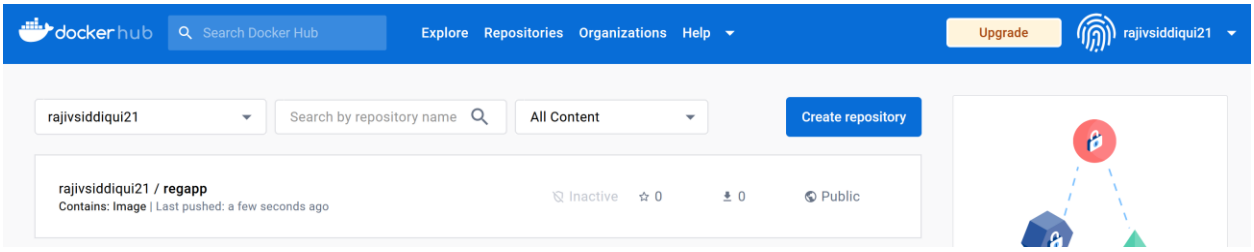
#docker login

#docker image tag <image_id> rajivsiddiqui21/regapp:latest

#docker image ls

#docker image push <image name>

now go to docker hub account and we can see our image



Now we do this thing automatically

Edit the file and add the line

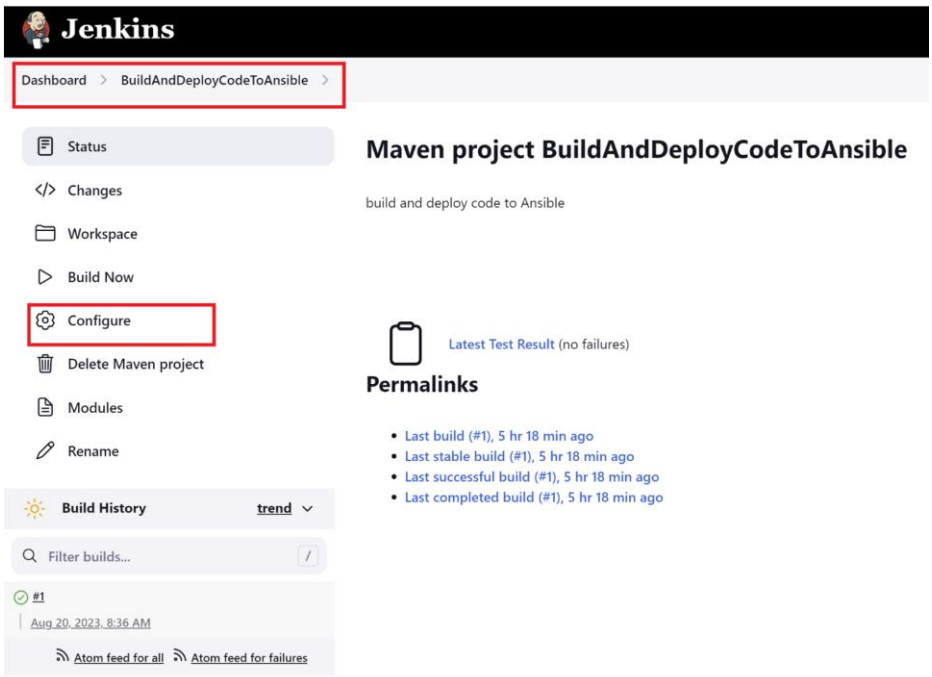
```
---
- hosts: ansible
```

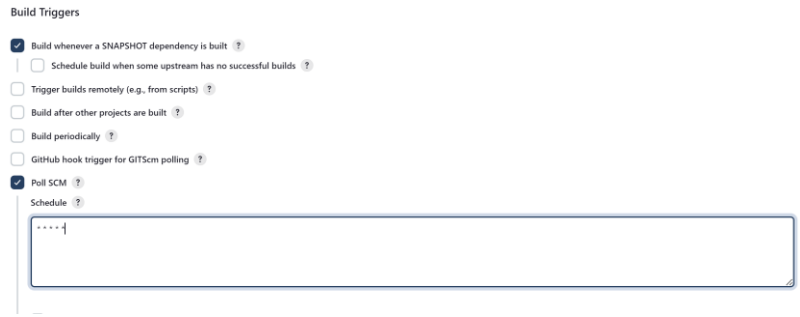
```yaml
tasks:
 - name: create docker image
   command: docker build -t regapp1:latest .
   args:
    chdir: /opt/docker

 - name: create tag on the image
   command: docker tag regapp1:latest rajivsiddiqui21/regapp1:latest

 - name: push image to docker hub
   command: docker push rajivsiddiqui21/regapp1:latest
```

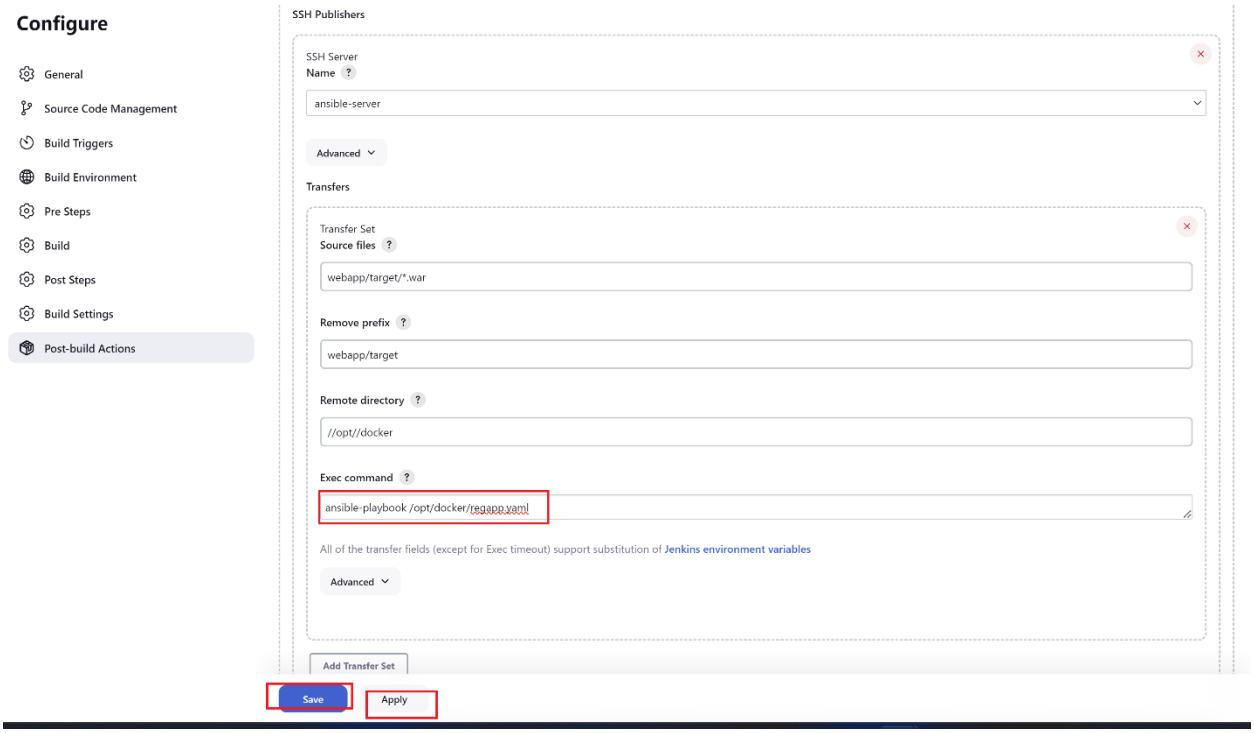
#ansible-playbook regapp.yml --check

ansible-playbook regapp.yml

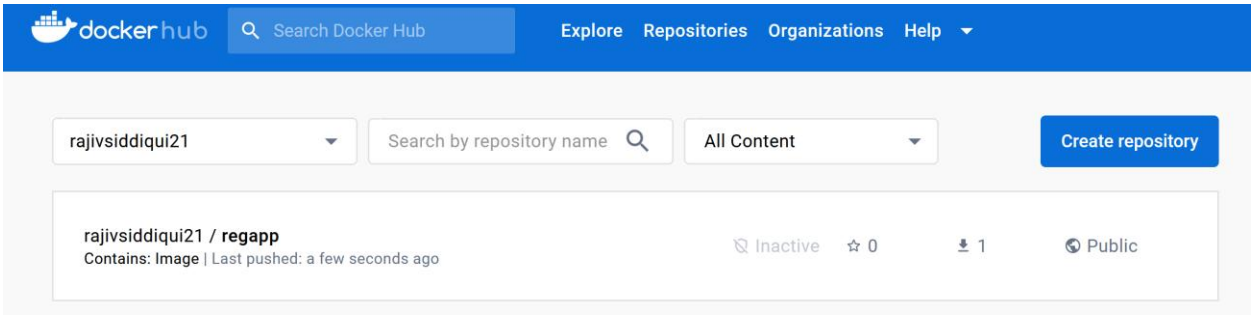
Integrate with Jenkins



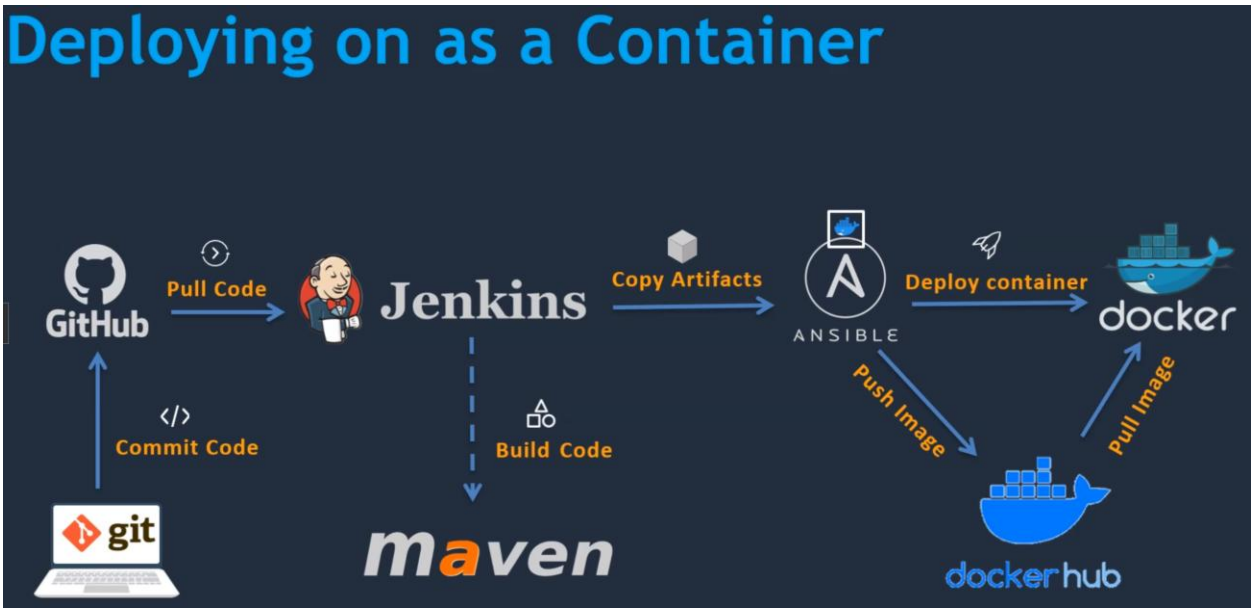




Now docker image tag and push to docker hub done by ansible and we can check form docker hub



Now pull this image from dockerhub and create a container to docker host



Create a ansible paly book file for creating a container to docker host

```
---
- hosts: docker

  tasks:
    - name: create container to docker host
      command: docker run -d --name regapp-server -p 8082:8080 rajivsiddiqui21/regapp:latest
```

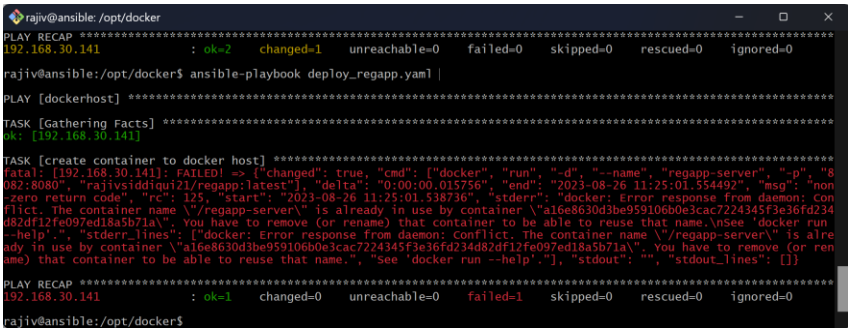
Check the ansible host/inventory file and make sure docker host ip is added

```
#vi /etc/ansible/hosts
```

Now run the ansible playbook file

```
#ansible-playbook deploy_regapp.yaml –check
#ansible-playbook deploy_regapp.yaml
```

Now if we run the same asnsible playbook file then get and error because the same name container already created



so we need to do the following things to overcome this problem.

- remove existing container
- remove existing image
- Create new container

so we need to edit the deploy_regapp.yaml

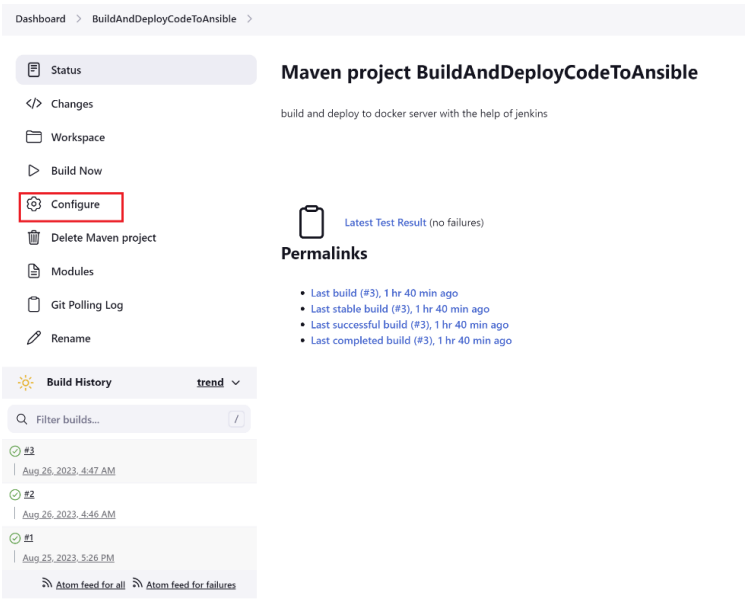
```
---
- hosts: dockerhost


  tasks:

    - name: stop existing container
      command: docker container stop regapp-server
      ignore_errors: yes

    - name: remove the container
      command: docker container rm regapp-server
      ignore_errors: yes

    - name: remove the image
      command: docker image rmi rajivsiddiqui21/regapp:latest
      ignore_errors: yes

    - name: create container to docker host
      command: docker run -d --name regapp-server -p 8082:8080
rajivsiddiqui21/regapp:latest
```

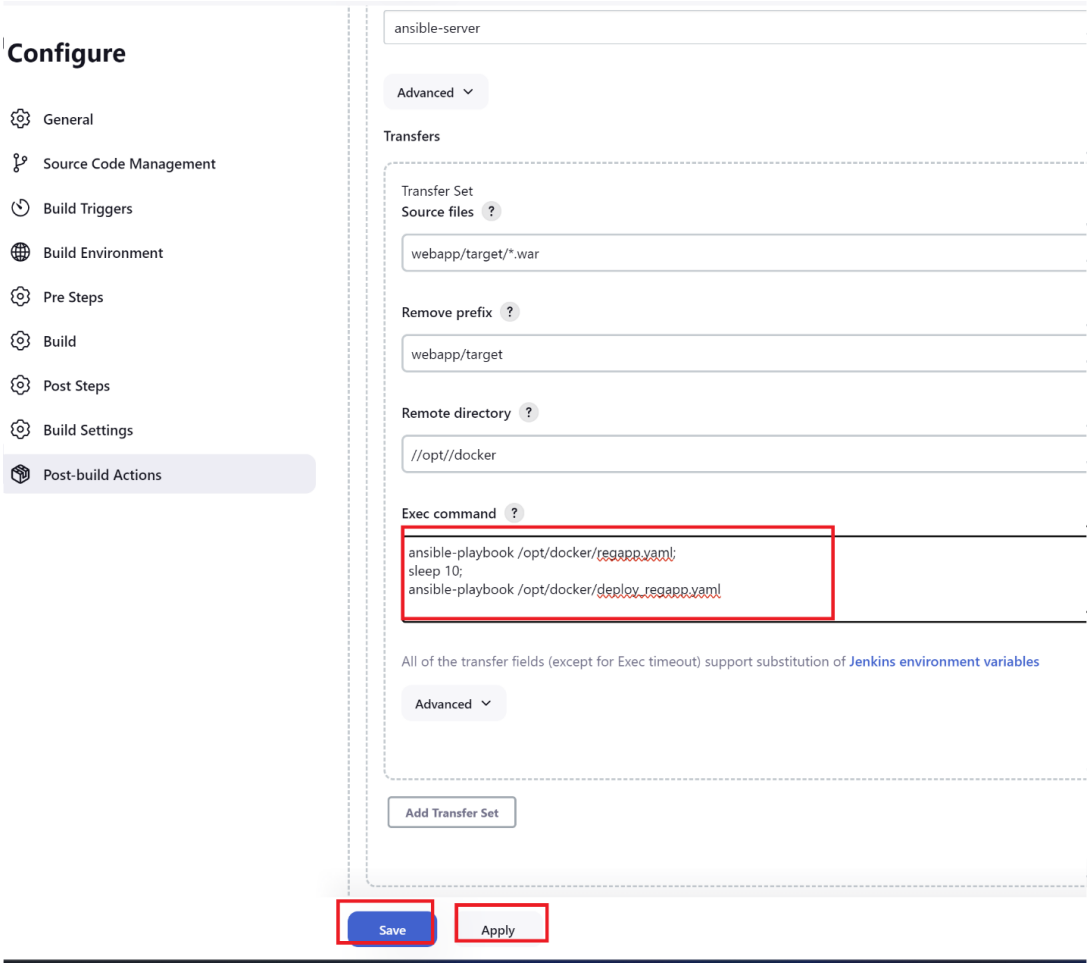
Docker image use in ansible command URL

https://docs.ansible.com/archive/ansible/2.3/docker_image_module.html

https://docs.ansible.com/ansible/latest/collections/community/docker/docker_image_module.html

Now run the ansible playbook file from jenkins





Command use in jenkins

```
ansible-playbook /opt/docker/regapp.yaml;
sleep 10;
ansible-playbook /opt/docker/deploy_regapp.yaml
```

Dashboard > BuildAndDeployCodeToAnsible >

- Status
- Changes
- Workspace
- Build Now
- Configure
- Delete Maven project
- Modules
- Git Polling Log
- Rename

Build History trend

Filter builds... /

#2
Aug 20, 2023, 2:16 PM

#1
Aug 20, 2023, 8:36 AM

Atom feed for all Atom feed for failures

Maven project BuildAndDeployCodeToAnsible

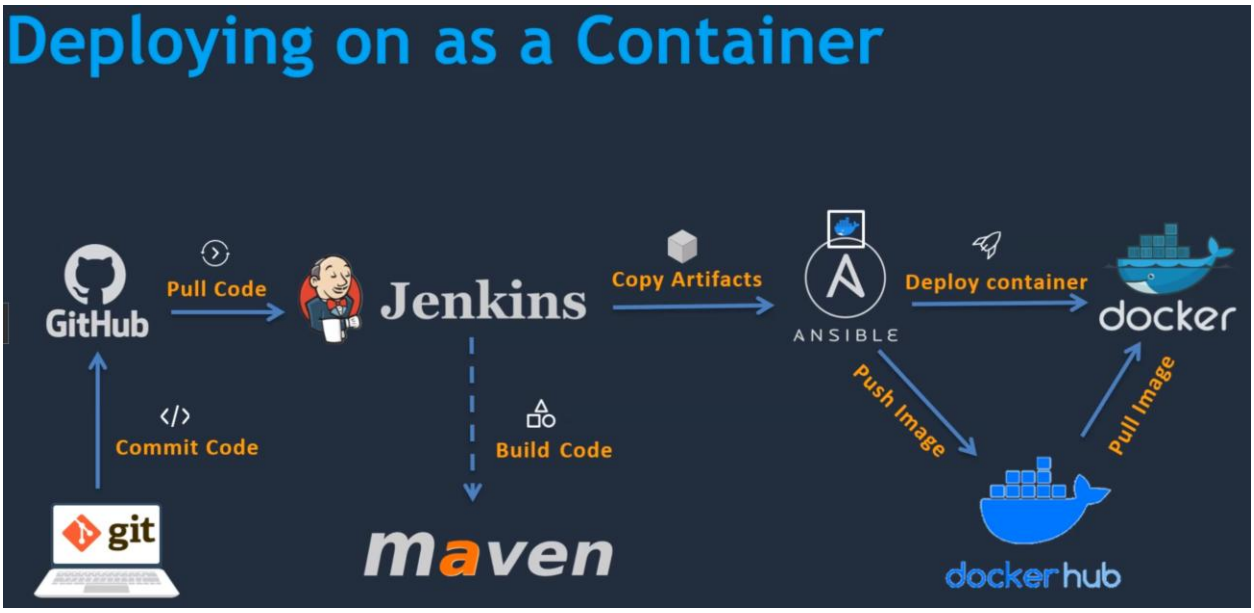
build and deploy code to Ansible

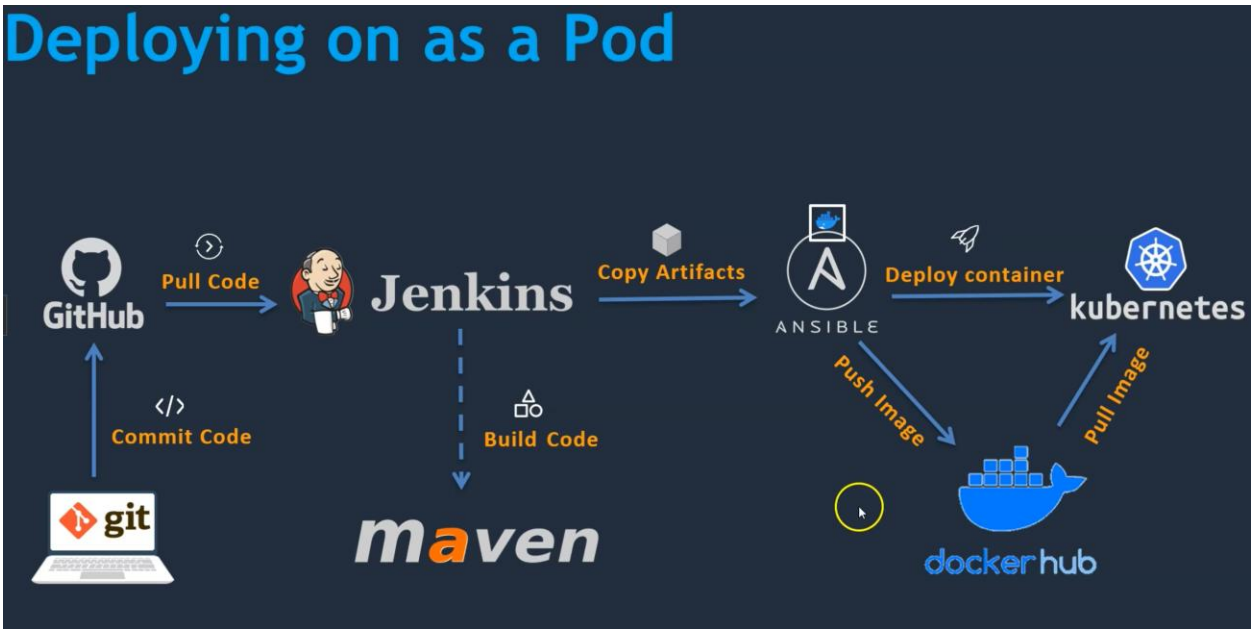
Latest Test Result (no failures)

Permalinks

- Last build (#1), 5 hr 37 min ago
- Last stable build (#1), 5 hr 37 min ago
- Last successful build (#1), 5 hr 37 min ago
- Last completed build (#1), 5 hr 37 min ago

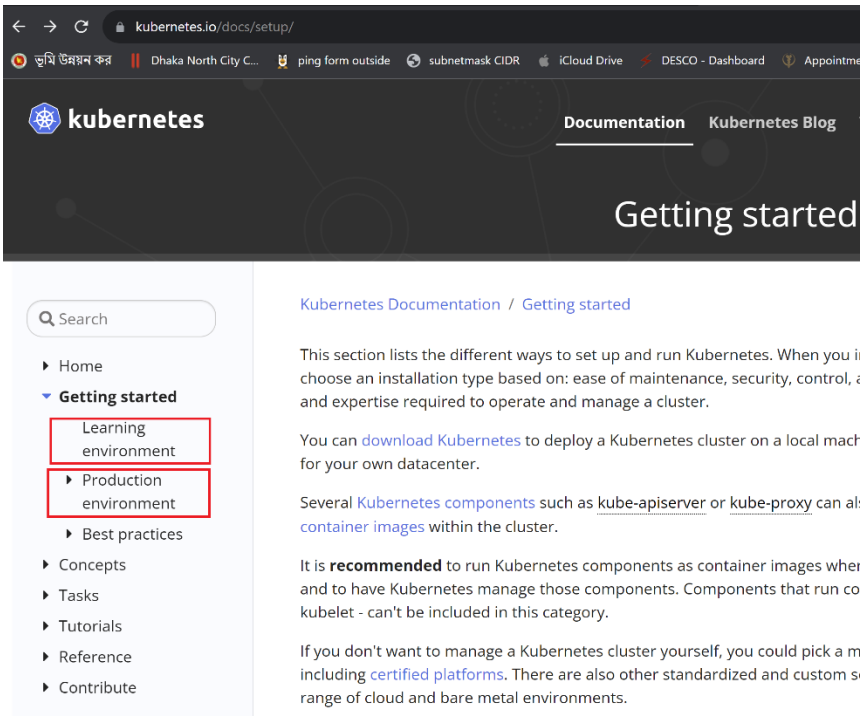
Deploy to Kubernetes

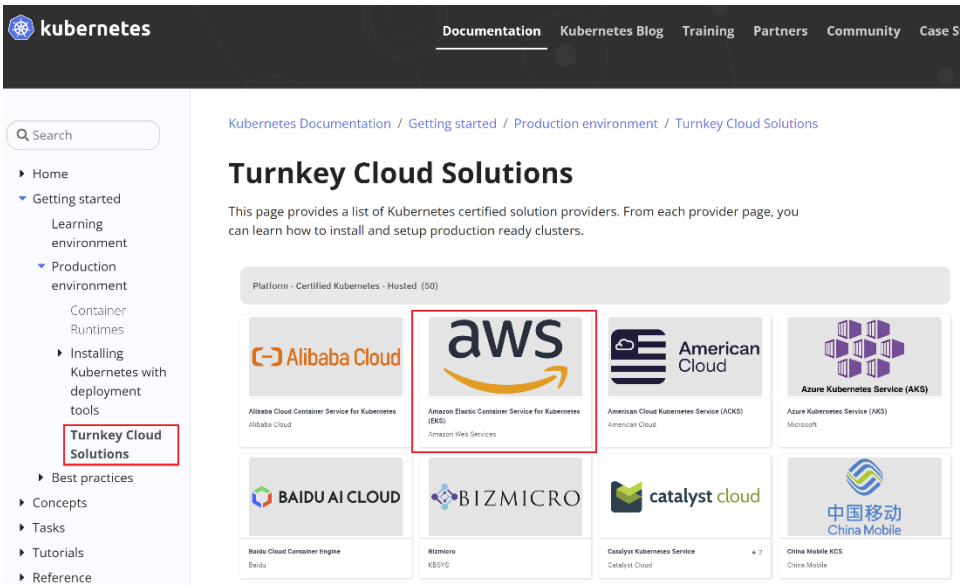




Setup Kubernetes environment

https://kubernetes.io/docs/setup/





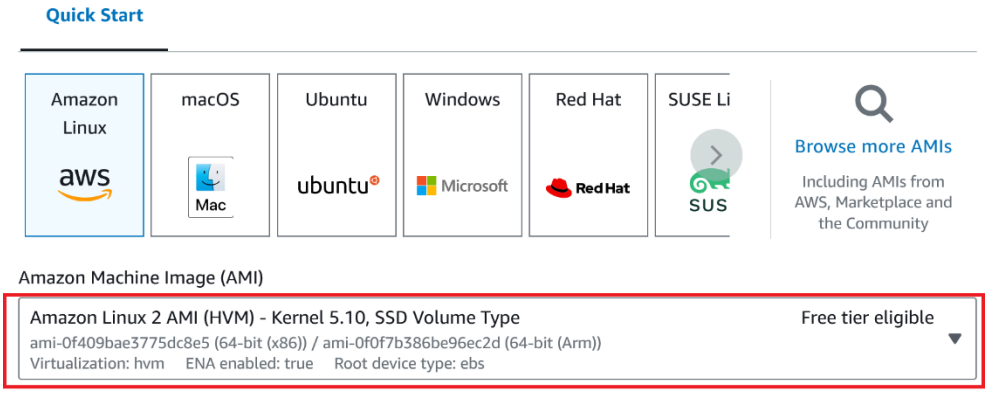
https://github.com/rajivsiddiqui/Simple-DevOps-Project

Kubernetes setup using eksctl

https://github.com/rajivsiddiqui/Simple-DevOps-Project/blob/master/Kubernetes/kubernetes_setup_using_eksctl.md

Step 1: launch a EC2 instance

when launch the instance make sure use the following amazon image



step 2: check AWS cli version if not latest version then install the latest version

#aws --version

```
ec2-user@ip-172-31-34-166:~
[ec2-user@ip-172-31-34-166 ~]$ aws --version
aws-cli/1.18.147 Python/2.7.18 Linux/5.10.186-179.751.amzn2.x86_64 botocore/1.18.6
[ec2-user@ip-172-31-34-166 ~]$
```

Aws cli latest version update

https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

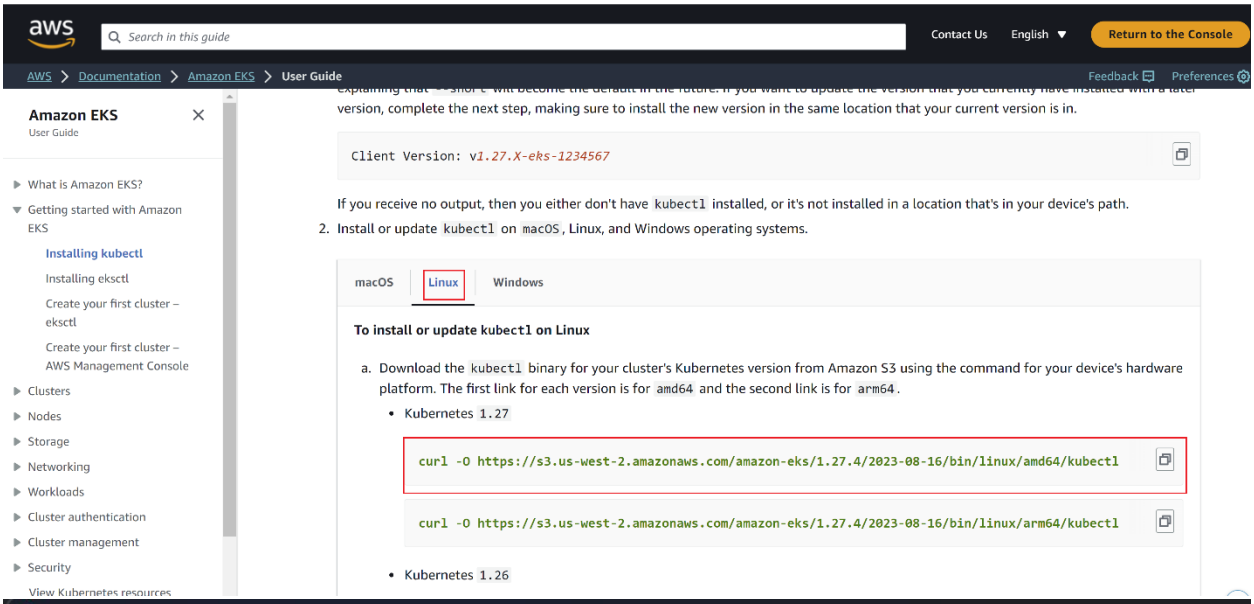
For get the latest version logout and login again

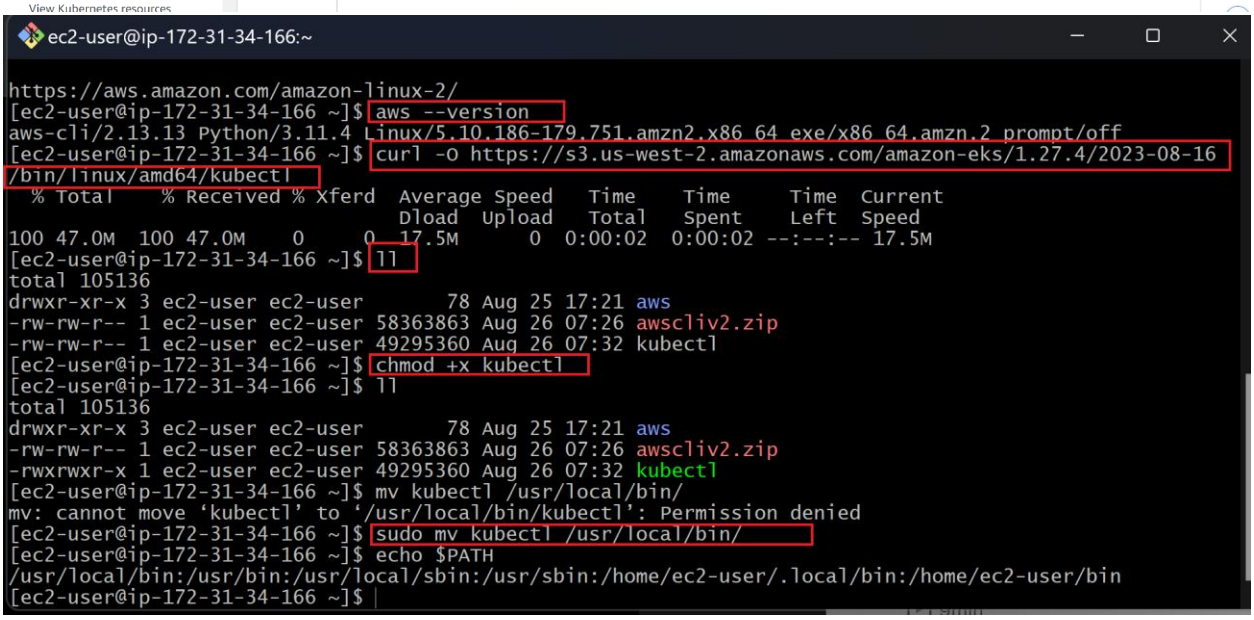
#aws --version

```
ec2-user@ip-172-31-34-166:~
[ec2-user@ip-172-31-34-166 ~]$ aws --version
aws-cli/1.18.147 Python/2.7.18 Linux/5.10.186-179.751.amzn2.x86_64 botocore/1.18.6
[ec2-user@ip-172-31-34-166 ~]$ exit
logout
Connection to ec2-3-94-122-241.compute-1.amazonaws.com closed.

aweso@RAJIV-XPS17 MINGW64 /d/aws
$ ssh -i "arina-n-ver.pem" ec2-user@ec2-3-94-122-241.compute-1.amazonaws.com
Last login: Sat Aug 26 07:18:27 2023 from 59.152.105.190

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-34-166 ~]$ aws --version
aws-cli/2.13.13 Python/3.11.4 Linux/5.10.186-179.751.amzn2.x86_64 exe/x86_64.amzn.2 prompt/off
[ec2-user@ip-172-31-34-166 ~]$
```

Step 3: Download and install the kubectl latest version

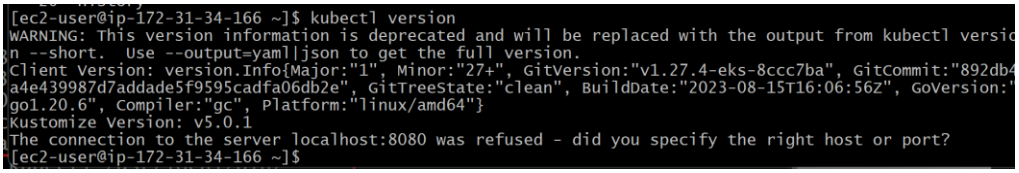
https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html






```
#aws --version
#curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.4/2023-08-16/bin/linux/amd64/kubectl
#ll
#chmod +x kubectl
#ll
#mv kubectl /usr/local/bin/
#sudo mv kubectl /usr/local/bin/
#echo $PATH
```

#kubectl version



Step 4: install eksctl

https://docs.aws.amazon.com/eks/latest/userguide/eksctl.html

https://eksctl.io/introduction/#installation

for installing run the following command

```
# for ARM systems, set ARCH to: `arm64`, `armv6` or `armv7`
ARCH=amd64
PLATFORM=$(uname -s)_$ARCH

curl -sLO "https://github.com/eksctl-io/eksctl/releases/latest/download/eksctl_$PLATFORM.tar.gz"

# (Optional) Verify checksum
curl -sL "https://github.com/eksctl-io/eksctl/releases/latest/download/eksctl_checksums.txt" | grep
$PLATFORM | sha256sum --check

tar -xzf eksctl_$PLATFORM.tar.gz -C /tmp && rm eksctl_$PLATFORM.tar.gz

sudo mv /tmp/eksctl /usr/local/bin
```

Now check the version
#eksctl version

```
[ec2-user@ip-172-31-34-166 ~]$ eksctl version
0.154.0
```

Step 5: create a IAM role

Ec2 full access
aws cloudformation full access
Iam full access

For making simple we create the IAM role with admin full access but its not right a good practice

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

IAM > Dashboard

IAM Dashboard

IAM resources

Resources in this AWS Account

User groups	Users
2	2

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

Policies

Identity providers

IAM > Roles

Roles (11) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Delete | Create role

Search

Role name	Trusted entities	Last activity
AWSServiceRoleForAmazonEKS	AWS Service: eks (Service-Linked Role)	16 days ago
AWSServiceRoleForAmazonEKSNodegroup	AWS Service: eks-nodegroup (Service-Linked Role)	16 days ago
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Linked Role)	16 days ago
AWSServiceRoleForCloudWatchEvents	AWS Service: events (Service-Linked Role)	44 days ago

IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity Info

Trusted entity type

- AWS service — Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account — Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity — Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation — Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy — Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

- EC2 — Allows EC2 instances to call AWS services on your behalf.
- Lambda — Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

Choose a service to view use case

Cancel | Next

IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Add permissions Info

Permissions policies (Selected 1/874) Info
Choose one or more policies to attach to your new role.

Create policy

Filter policies by property or policy name and press enter.

1 2 3 4 5 6 7 ... 44

Policy name	Type	Description
AdministratorAccess	AWS m...	Provides full access to AWS services and resources.
PowerUserAccess	AWS m...	Provides full access to AWS services and resources, but does not allow management of Users and groups.
ReadOnlyAccess	AWS m...	Provides read-only access to AWS services and resources.
AWSCloudFormationReadOnlyAccess	AWS m...	Provides access to AWS CloudFormation via the AWS Management Console.
CloudFrontFullAccess	AWS m...	Provides full access to the CloudFront console plus the ability to list Amazon S3 buckets via the AWS Mana...
AWSCloudHSMFullAccess	AWS m...	Provides full access to all CloudHSM resources.
AWSCloudHSMReadOnlyAccess	AWS m...	Provides read only access to all CloudHSM resources.
AmazonAppStreamReadOnlyAccess	AWS m...	Provides read only access to Amazon AppStream via the AWS Management Console.

Set permissions boundary - *optional* Info

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel Previous Next

IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

eksctl_role_for_regapp

Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Step 1: Select trusted entities

Edit

```
1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Effect": "Allow",
6              "Action": [
7                  "sts:AssumeRole"
8              ],
9              "Principal": {
10             "Service": [
11                 "ec2.amazonaws.com"
12             ]
13             }
14         }
15     ]
16 }
```

Step 2: Add permissions

Edit

Permissions policy summary

Policy name	Type	Attached as
AdministratorAccess	AWS managed - job function	Permissions policy

Tags

Add tags - *optional* Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

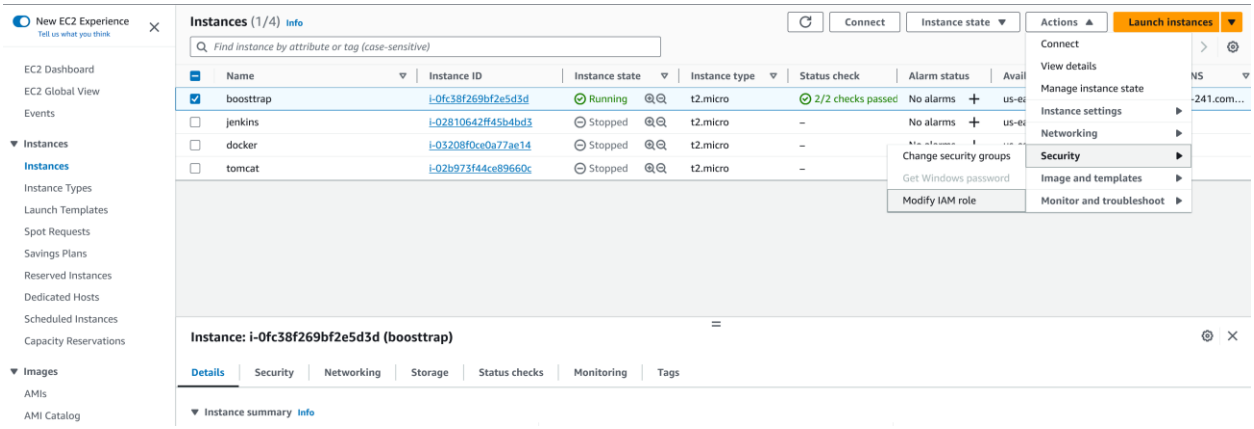
No tags associated with the resource.

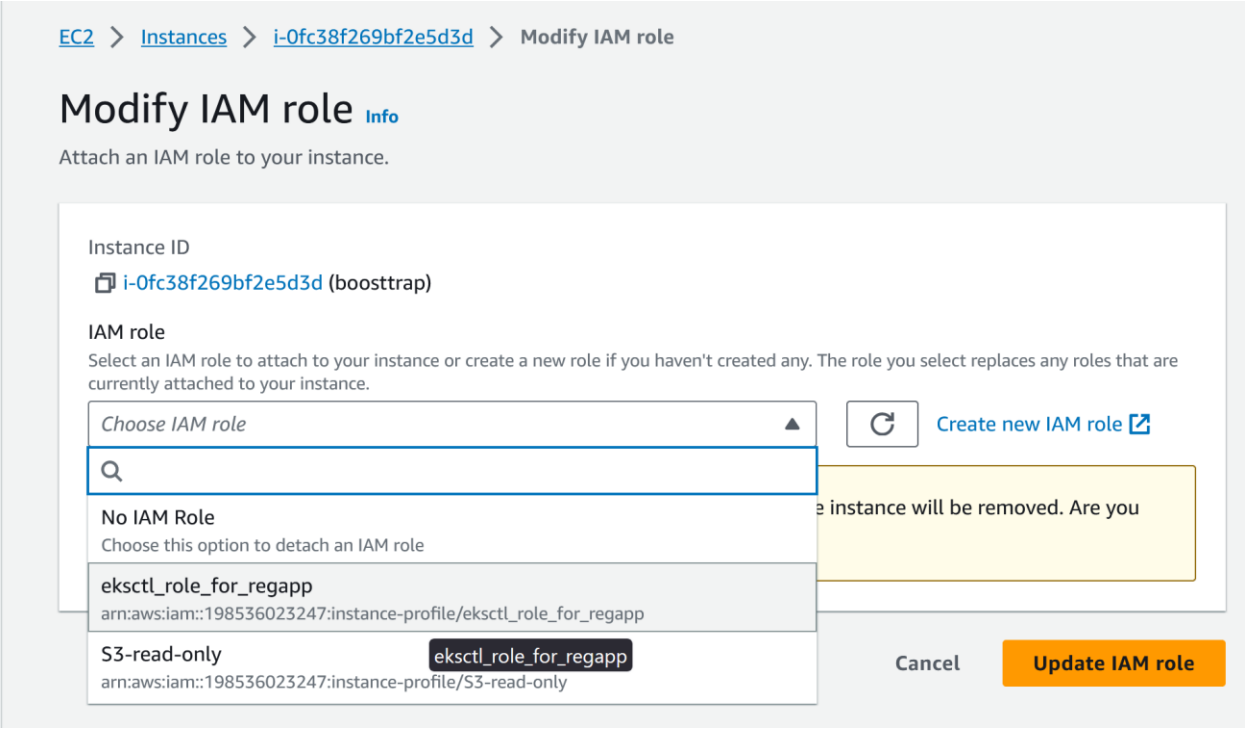
Add tag

You can add up to 50 more tags.

Cancel Previous Create role

Step 6: Attach this role to bootstarp Ec2





Step 7: Now create Cluster

eksctl create cluster --name cluster-name \

--region region-name \

--node-type instance-type \

--nodes-min 2 \

--nodes-max 2 \

--zones <AZ-1>,<AZ-2>

example:

eksctl create cluster --name regapp5-cluster \

--region us-east-1 \

--node-type t2.small

Step 8: check the cluster and run a httpd container to test it

Main config file location
#cat /root/.kube/config

#kubectl get nodes
#kubectl get all
#kubectl run webapp --image=httpd

Deploy an nginx deploymet for test

kubectl create deployment demo-nginx --image=nginx --replicas=2 --port=80

kubectl deployment regapp --image=rajivsiddiqui21/regapp --replicas=2 --port=8080

kubectl get all

kubectl get pod

Expose the deployment as service. This will create an ELB in front of those 2 containers and allow us to publicly access them

kubectl expose deployment demo-nginx --port=80 --type=LoadBalancer

kubectl expose deployment regapp --port=8080 --type=LoadBalancer

kubectl get services -o wide

Now delete the deployment

```
#kubectl delete deployment demo-nginx
#kubectl get svc
#kubectl delete svc demo-nginx
```

Kubernetes pod creation by file

pod manifest file documentation
https://kubernetes.io/docs/concepts/workloads/pods/

service manifest file documentation
https://kubernetes.io/docs/concepts/services-networking/service/

Step:8 deploy this app to kubernetes

Create regadd-deploy.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: rajivsiddiqui21-regapp
```

```
  labels:
    app: regapp

spec:
  replicas: 3
  selector:
    matchLabels:
      app: regapp

  template:
    metadata:
      labels:
        app: regapp
    spec:
      containers:
      - name: regapp
        image: rajivsiddiqui21/regapp
        imagePullPolicy: Always
        ports:
        - containerPort: 8080
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
```

Create regapp-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: rajivsiddiqui21-service
  labels:
    app: regapp
spec:
  selector:
    app: regapp

  ports:
    - port: 8080
      targetPort: 8080

  type: LoadBalancer
```

```
#kubectl apply -f regapp-deploy.yaml
#kubectl apply -f regapp-service.yaml
#kubectl get all
#kubectl get deployments
```

#kubectl describe deployment <deployment_name>
#kubectl delete deployment <deployment_name>

Now we can browse the URL and get the page
http://a1a5b85045ceb4475b8d70478cf9f769-500062273.us-east-1.elb.amazonaws.com:8080/webapp/

Step 9: Integrate Kubernetes with ansible

In bootstrap server need to do the following things
- add a user
    #sudo su
    #useradd rajiv
    #passwd rajiv
-add sudo previlidge
    #vi /etc/sudoers
    add the following line
    rajiv ALL=(ALL) NOPASSWD: ALL

```
root@ip-172-31-34-166:/home/ec2-user
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
rajiv   ALL=(ALL)       NOPASSWD: ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL
```

- Enable password based authentication
    - #vi /etc/ssh/sshd_config
    -
      ```
      # To disable tunneled clear text passwords, change to no here!
      PasswordAuthentication yes
      #PermitEmptyPasswords no
      #PasswordAuthentication no
      ```
    - #systemctl reload sshd
- Now go to ansible server
    - Login as a ansadmin user
    - #cd /opt/docker
      #ll
      #mv regapp.yaml create_image_regapp.yaml
      #mv deploy_regapp.yaml docker_deployment.yaml

- Add bootstrap server ip in ansible host file
  #vi /etc/ansible/hosts

```
rajiv@ansible: /opt/docker
[dockerhost]
192.168.30.141

[ansible]
192.168.30.145

[kubernetes]
3.94.122.241
~
```

- Ansible Ssh key copy to bootstrap server
  #ssh-copy-id rajiv@3.94.122.241

- Now check everything is working
  #ansible -i /etc/ansible/hosts all -a uptime

```
rajiv@ansible:/etc$ ansible -i /etc/ansible/hosts all -a uptime
192.168.30.145 | CHANGED | rc=0 >>
 10:41:04 up  8:42,  3 users,  load average: 0.20, 0.27, 0.24
192.168.30.141 | CHANGED | rc=0 >>
 16:41:03 up  8:58,  3 users,  load average: 0.35, 0.22, 0.19
[WARNING]: Platform linux on host 3.94.122.241 is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
3.94.122.241 | CHANGED | rc=0 >>
 10:41:16 up  3:25,  2 users,  load average: 0.00, 0.00, 0.00
rajiv@ansible:/etc$
```

-
- Now create a mainfast file in ansible
  #vi kube_deploy.yaml

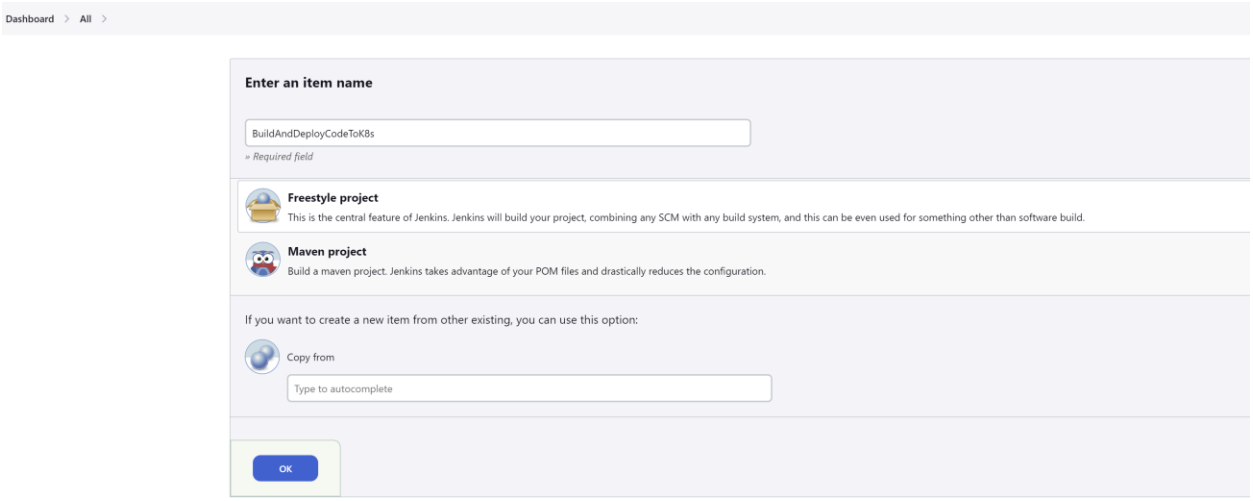
```
---
- hosts: kubernetes
#  become: ture
  user: ec2-user

  tasks:
    - name: deploy regapp on kubernetes
      command: kubectl apply -f /home/ec2-user/regapp-deploy.yaml

    - name: create service for regapp
      command: kubectl apply -f /home/ec2-user/regapp-service.yaml

    - name: update deployment with new pods if image updated
docker hub
      command: kubectl rollout restart
deployment.apps/rajivsiddiqui21-regapp
```

-
- Now check bootstrap server and we can see deployment is running
- Now delete the deployment from bootstrap server
  #kubectl delete deployment rajivsiddiqui21-regapp

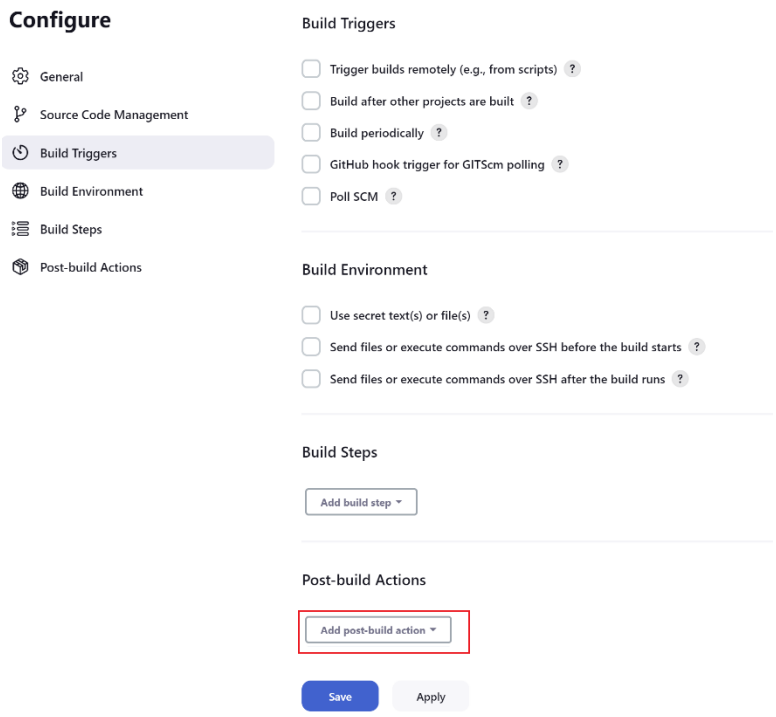
NB: Under which user we installed kubectl and eksctl on ec2 instance (bootsserver) make sure that user run the file when connecting from ansible to bootstrap server also use the same user in manifest file kube_deply.yaml

Step 10: now do this from Jenkins server

Open Jenkins from browser


Dashboard > All >
Enter an item name
BuildAndDeployCodeToK8s
» 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.
If you want to create a new item from other existing, you can use this option:
Copy from
Type to autocomplete
OK

scroll down


Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
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
Use secret text(s) or file(s)
Send files or execute commands over SSH before the build starts
Send files or execute commands over SSH after the build runs
Build Steps
Add build step
Post-build Actions
Add post-build action
Save
Apply

Dashboard > BuildAndDeployCodeToK8s > Configuration

Configure

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

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

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

Filter

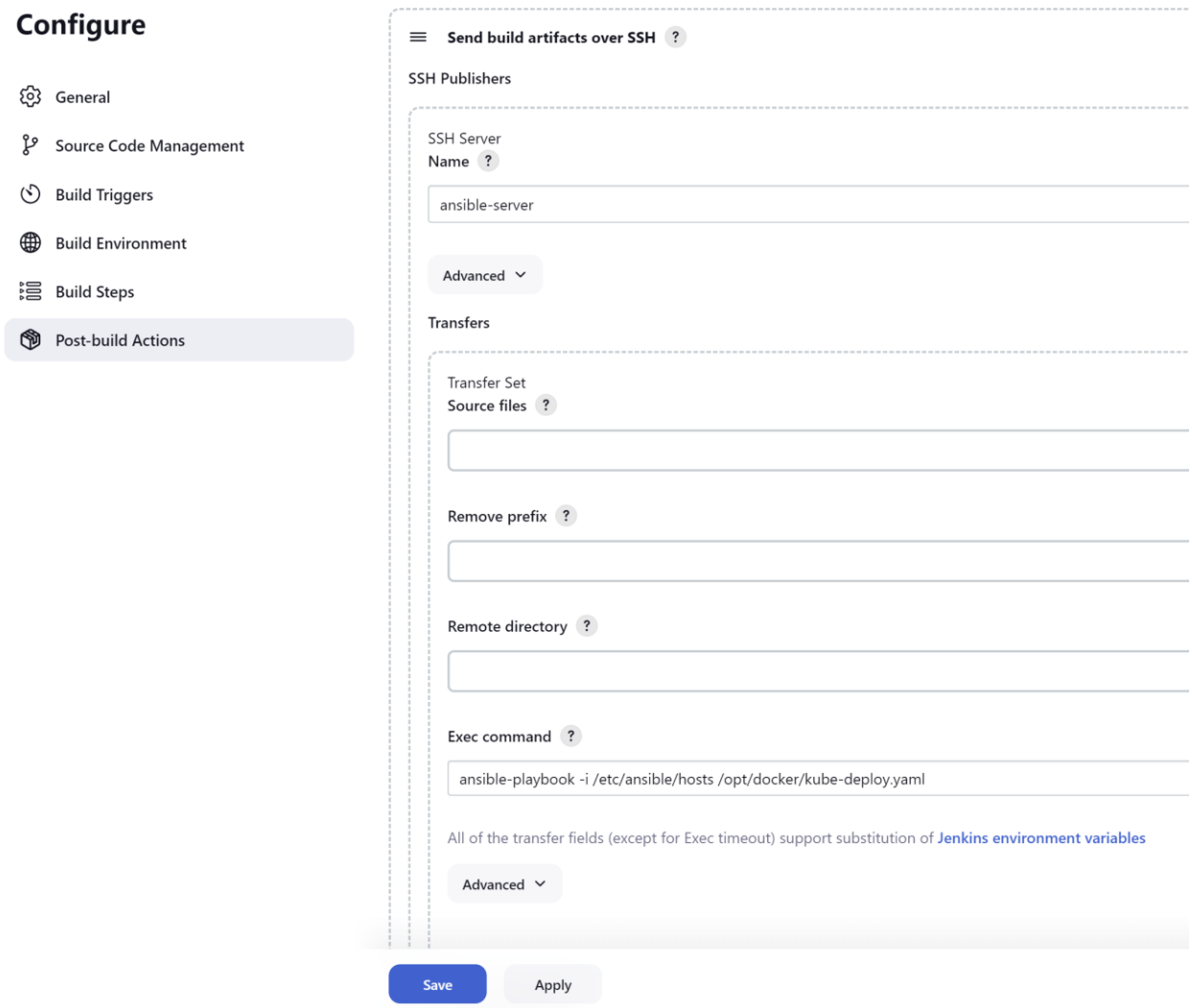
before the build starts ?

after the build runs ?

- Aggregate downstream test results
- Archive the artifacts
- Build other projects
- Publish JUnit test result report
- Publish Javadoc
- Record fingerprints of files to track usage
- Git Publisher
- Deploy war/ear to a container
- E-mail Notification
- Send build artifacts over SSH
- Set GitHub commit status (universal)
- Set build status on GitHub commit [deprecated]

Add post-build action

Save Apply

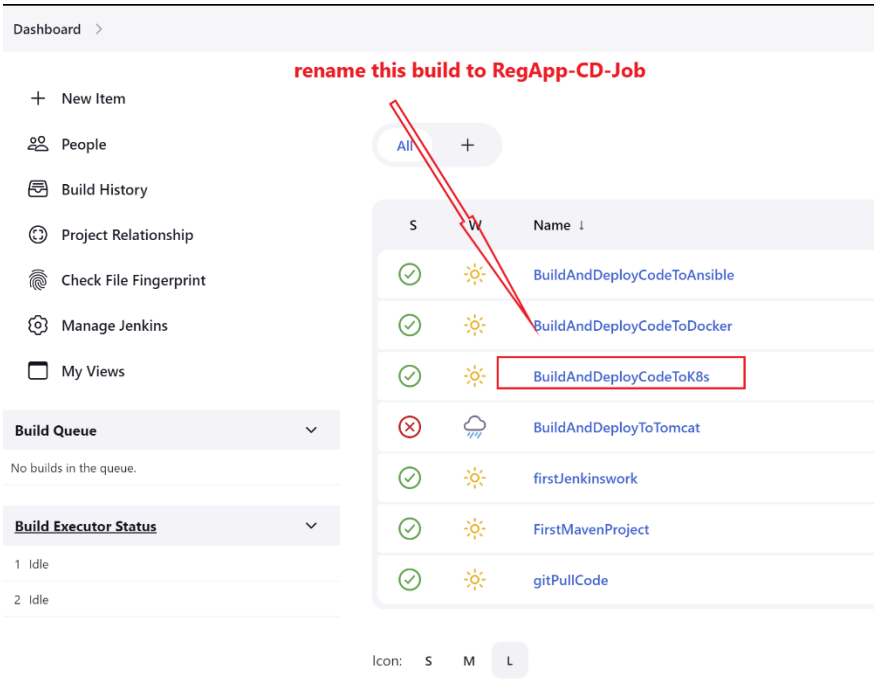


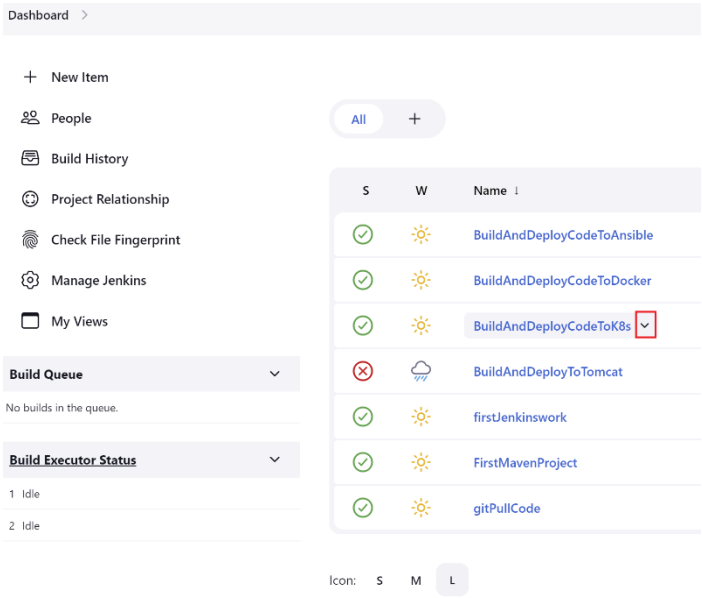

```
#ansible-playbook -i /etc/ansible/hosts /opt/docker/kube-deploy.yaml
```

Step 11:

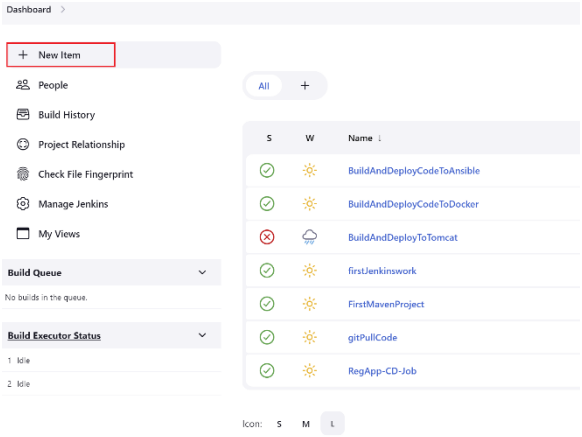
Create CD jobs:

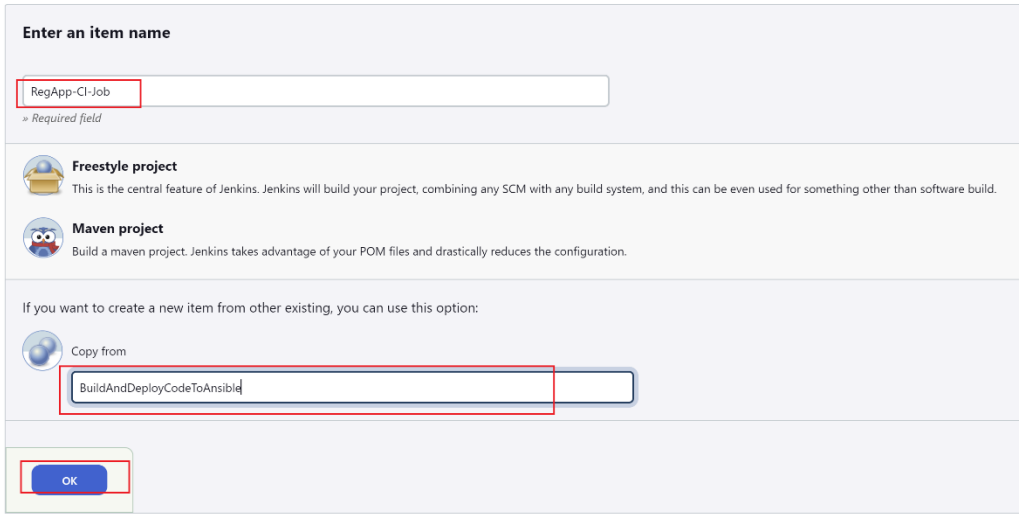
Open Jenkins from browser
edit the build name RegApp-CD-job

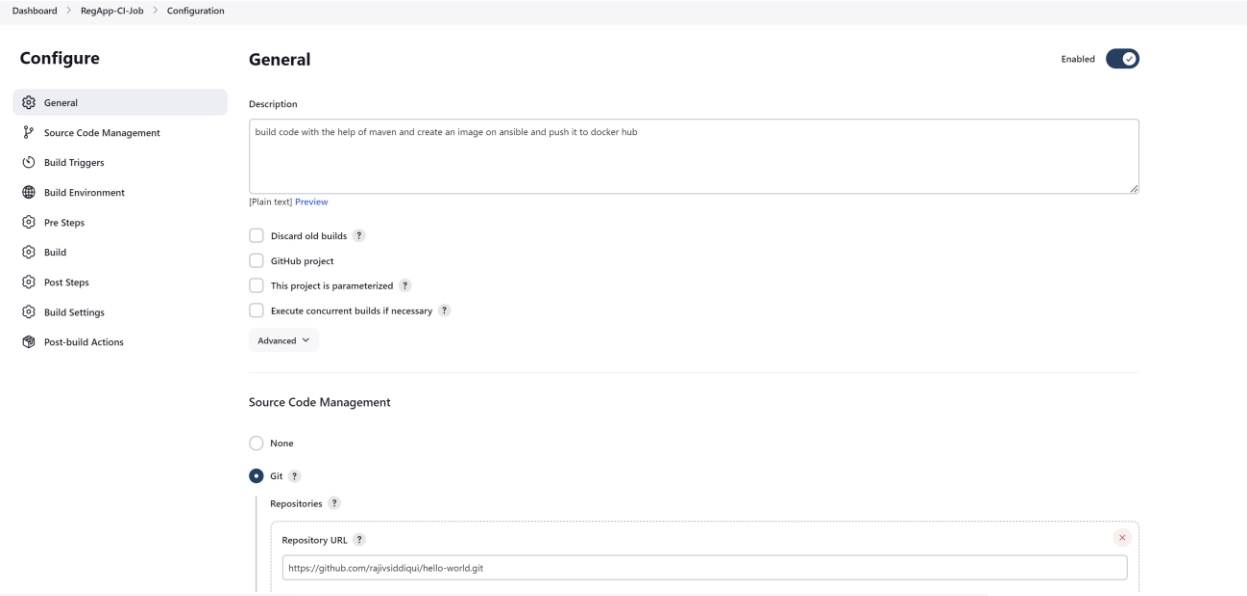

Dashboard
rename this build to RegApp-CD-Job
New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
All
S
W
Name
BuildAndDeployCodeToAnsible
BuildAndDeployCodeToDocker
BuildAndDeployCodeToK8s
BuildAndDeployToTomcat
firstJenkinswork
FirstMavenProject
gitPullCode
Icon: S M L

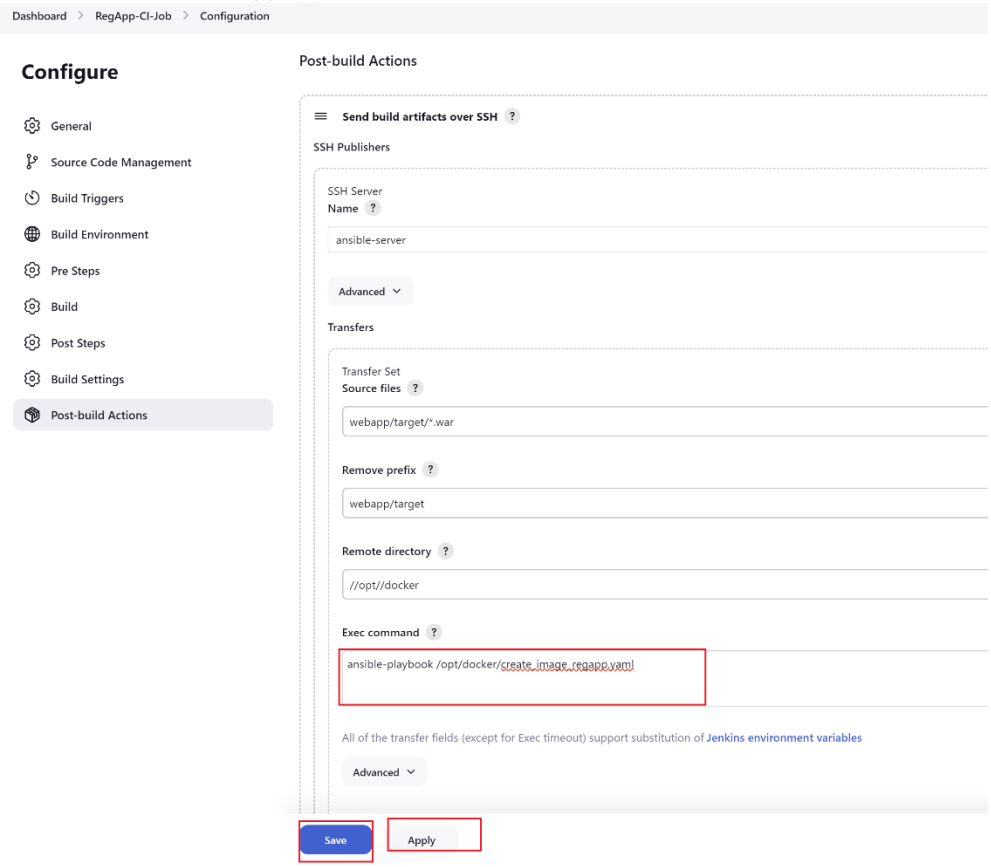

Dashboard
New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
All
S
W
Name
BuildAndDeployCodeToAnsible
BuildAndDeployCodeToDocker
BuildAndDeployCodeToK8s
BuildAndDeployToTomcat
firstJenkinswork
FirstMavenProject
gitPullCode
Icon: S M L

Now Create CI job





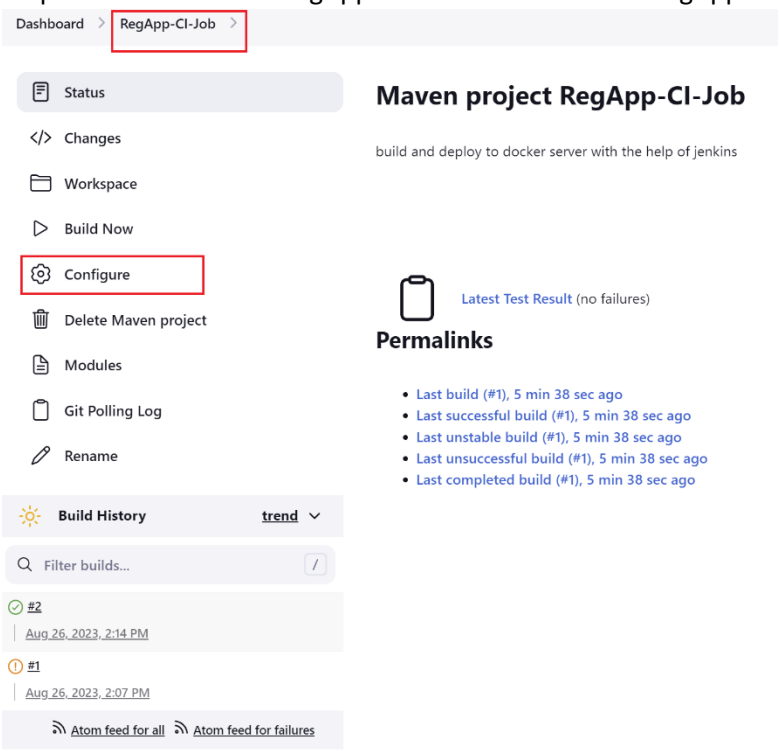


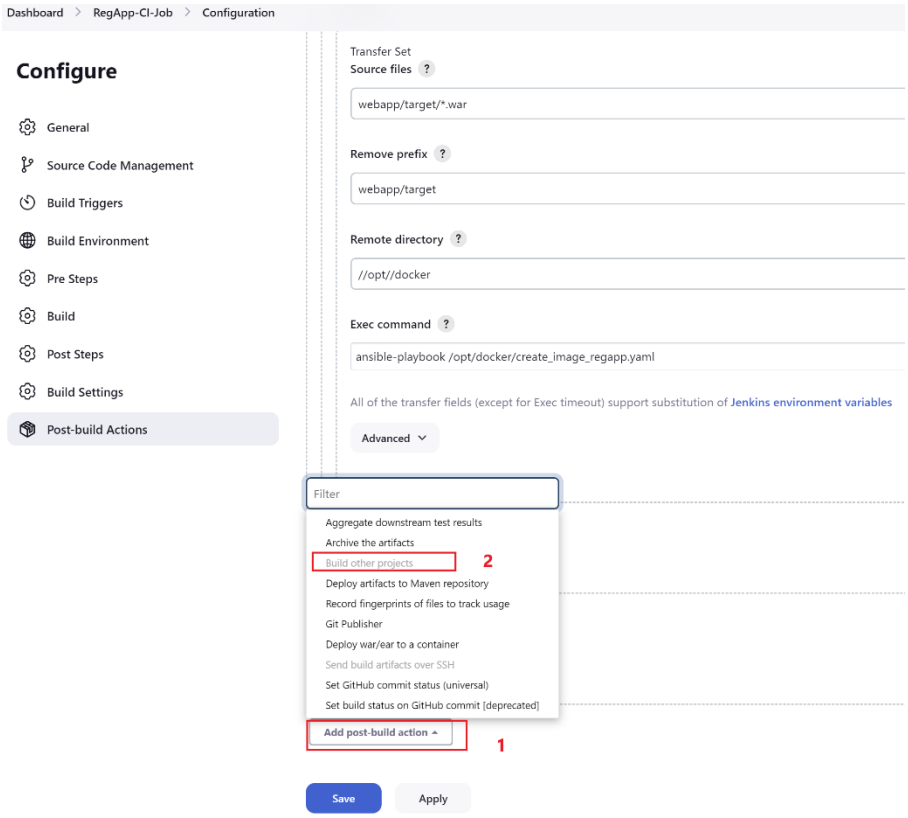



```
#ansible-playbook /opt/docker/create_image_regapp.yaml
```

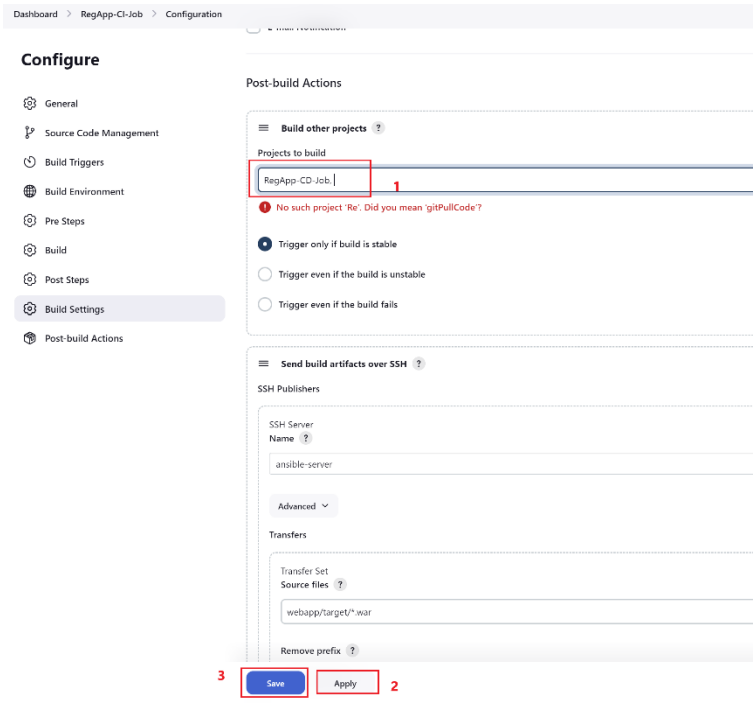
Now go to your local pc and edit the code and push it and we can see the new image is created.

Step 12: add in Jenkins RegApp-CI-Job done then start RegApp-CD-Job





Scroll up



Now edit kube_deploy.yaml

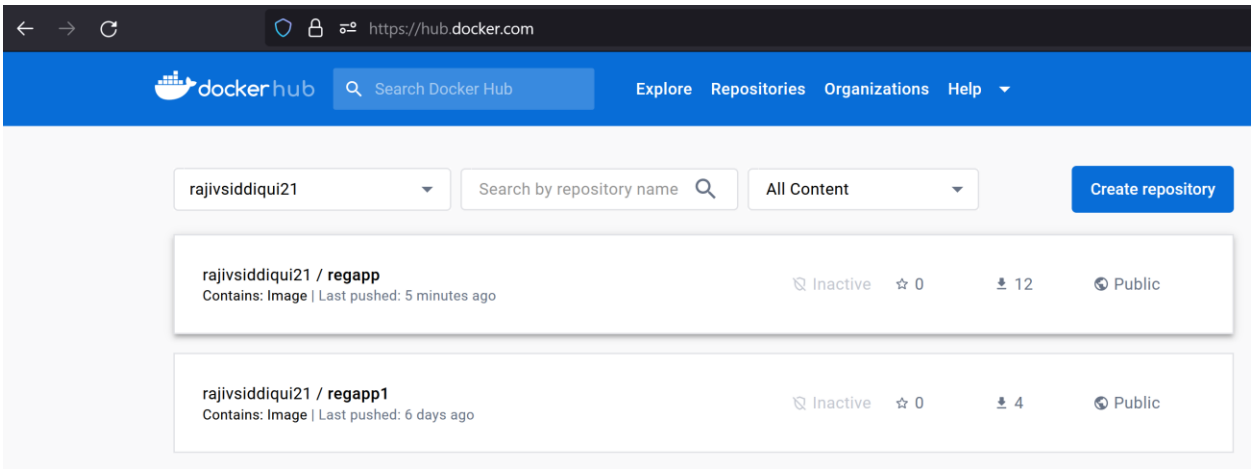
```
---
- hosts: kubernetes
#  become: ture
  user: ec2-user

  tasks:
    - name: deploy regapp on kubernetes
      command: kubectl apply -f /home/ec2-user/regapp-deploy.yaml

    - name: create service for regapp on kubernetes
      command: kubectl apply -f /home/ec2-user/regapp-service.yaml

    - name: update deployment with new pods if image updated in docker hub
      command: kubectl rollout restart deployment.apps/rajivsiddiqui21-regapp
```

Now edit the index from our local pc and push to git hub and we can see the latest update

And also we can see the artifact is created
new image is update to docker hub
and Kubernetes pods is created just now
and the webpage is showing new data

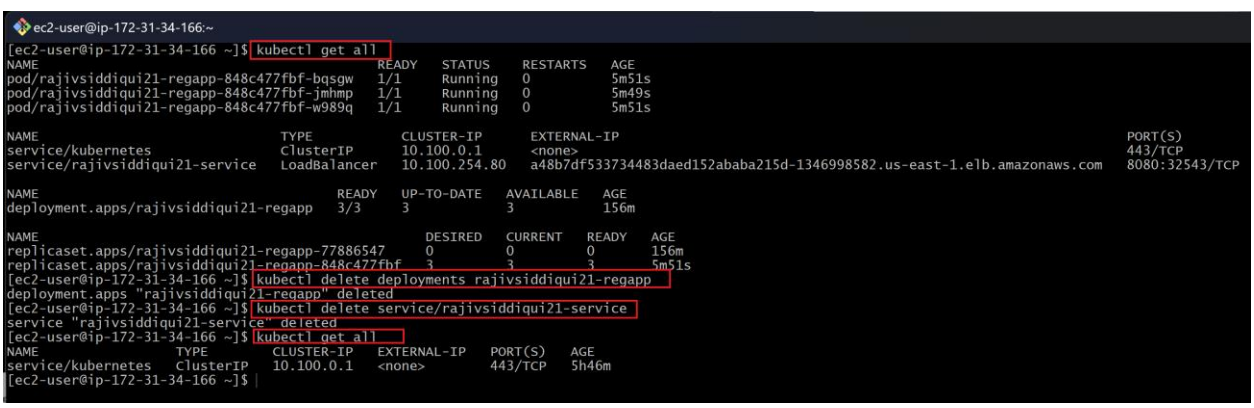
```
rajiv@ansible:/opt/docker$ ll
total 28
drwxr-xr-x 2 rajiv rajiv 4096 Aug 26 14:33 ./
drwxr-xr-x 4 root  root  4096 Aug 25 17:58 ../
-rw-rw-r-- 1 rajiv rajiv  349 Aug 26 03:51 create_image_regapp.yaml
-rw-rw-r-- 1 rajiv rajiv  520 Aug 26 05:34 docker_deployment.yaml
-rw-rw-r-- 1 rajiv rajiv  128 Aug 25 18:00 Dockerfile
-rw-rw-r-- 1 rajiv rajiv  455 Aug 26 14:33 kube-deploy.yaml
-rw-rw-r-- 1 rajiv rajiv 2395 Aug 26 14:37 webapp.war
rajiv@ansible:/opt/docker$ date
Sat 26 Aug 2023 02:38:31 PM UTC
rajiv@ansible:/opt/docker$
```

```
[ec2-user@ip-172-31-34-166 ~]$ kubectl get all
NAME                                          READY   STATUS    RESTARTS   AGE
pod/rajivsiddiqui21-regapp-77886547-k28wf     1/1     Running   0          27s
pod/rajivsiddiqui21-regapp-77886547-rqbmv     1/1     Running   0          27s
pod/rajivsiddiqui21-regapp-77886547-wj7nx     1/1     Running   0          27s

NAME                                  TYPE           CLUSTER-IP      EXTERNAL-IP
 PORT(S)          AGE
service/kubernetes                    ClusterIP      10.100.0.1      <none>
 443/TCP          3h9m
service/rajivsiddiqui21-service       LoadBalancer   10.100.254.80   a48b7df533734483daed152ababa215d-1346998582.us-east-1.elb.amazonaws.com
 8080:32543/TCP   17s

NAME                                        READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/rajivsiddiqui21-regapp      3/3     3            3           27s

NAME                                                   DESIRED   CURRENT   READY   AGE
replicaset.apps/rajivsiddiqui21-regapp-77886547        3         3         3       27s
[ec2-user@ip-172-31-34-166 ~]$ kubectl get all
NAME                                          READY   STATUS    RESTARTS   AGE
pod/rajivsiddiqui21-regapp-848c477fbf-bqsgw   1/1     Running   0          5s
pod/rajivsiddiqui21-regapp-848c477fbf-jmhmp   1/1     Running   0          3s
pod/rajivsiddiqui21-regapp-848c477fbf-w989q   1/1     Running   0          5s

NAME                                  TYPE           CLUSTER-IP      EXTERNAL-IP                                                                   PORT(S)
service/kubernetes                    ClusterIP      10.100.0.1      <none>                                                                        443/TCP
service/rajivsiddiqui21-service       LoadBalancer   10.100.254.80   a48b7df533734483daed152ababa215d-1346998582.us-east-1.elb.amazonaws.com   8080:32543/TCP

NAME                                        READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/rajivsiddiqui21-regapp      3/3     3            3           150m

NAME                                                   DESIRED   CURRENT   READY   AGE
replicaset.apps/rajivsiddiqui21-regapp-77886547        0         0         0       150m
replicaset.apps/rajivsiddiqui21-regapp-848c477fbf      3         3         3       5s
[ec2-user@ip-172-31-34-166 ~]$
```



Step 13: delete deployments and delete the aws cluster

Delete the deployments



Delete the cluster

```
eksctl delete cluster <cluster name> --region us-east-1
# eksctl delete cluster regapp-cluster --region us-east-1
```