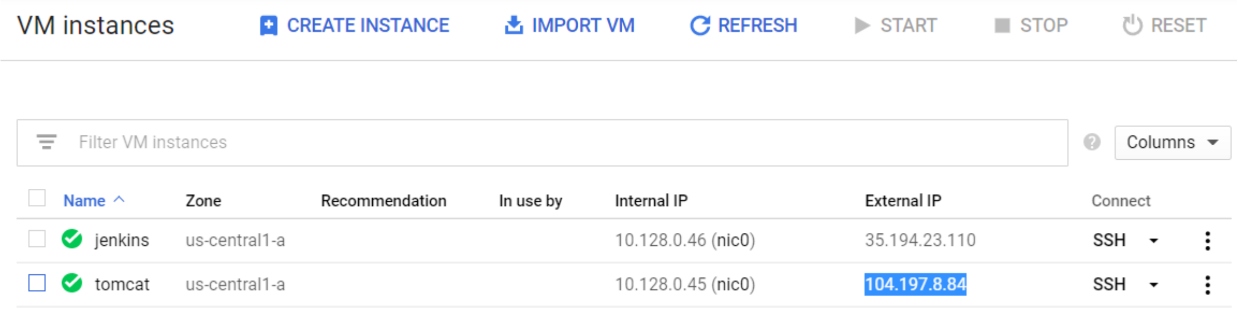
Scenario 1:- CI/CD (Jenkins + Git hub + Maven + Tomcat)

With the help of Jenkins orchestration tool, Maven will build the code from GIT repo and the artifact will be deployed to the bare metal tomcat server.

Server 1: - Jenkins (Make sure to install Java, Jenkins, Git, maven)
Server 2: - tomcat (Make sure to install Java, Tomcat)

Step1:- Create a two instance in GCP with 8080 port enabled on firewall rule.



Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
jenkins	us-central1-a			10.128.0.46 (nic0)	35.194.23.110	SSH
tomcat	us-central1-a			10.128.0.45 (nic0)	104.197.8.84	SSH

```
#yum update -y && yum install wget -y && yum install mlocate -y && updated
#setenforce 0
#vim /etc/selinux/config
#reboot
```

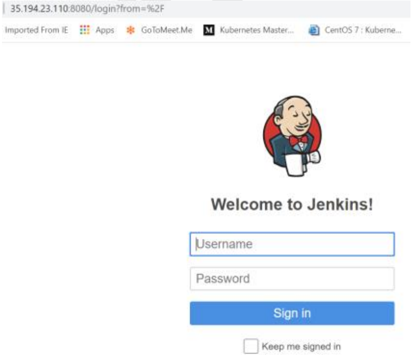
Step 2:- Install Java on both servers (jenkins and tomcat)

```
#yum -y install java-1.8.0-openjdk java-1.8.0-openjdk-devel
#cat > /etc/profile.d/java8.sh <<EOF
export JAVA_HOME=$(dirname $(dirname $(readlink $(readlink $(which javac)))))
export PATH=\$PATH:\$JAVA_HOME/bin
export CLASSPATH=.:\$JAVA_HOME/jre/lib:\$JAVA_HOME/lib:\$JAVA_HOME/lib/tools.jar
EOF
#source /etc/profile.d/java8.sh
```

Step3:- Install jenkins by following steps.

```
#cd /etc/yum.repos.d && curl -O https://pkg.jenkins.io/redhat-stable/jenkins.repo
#rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
```

@@Try to access jenkin via 8080 port example :- (http://35.194.23.110:8080/)@@
Note: - enter admin password and cancel the plug-in options for custom selection.



Welcome to Jenkins!

Username

Password

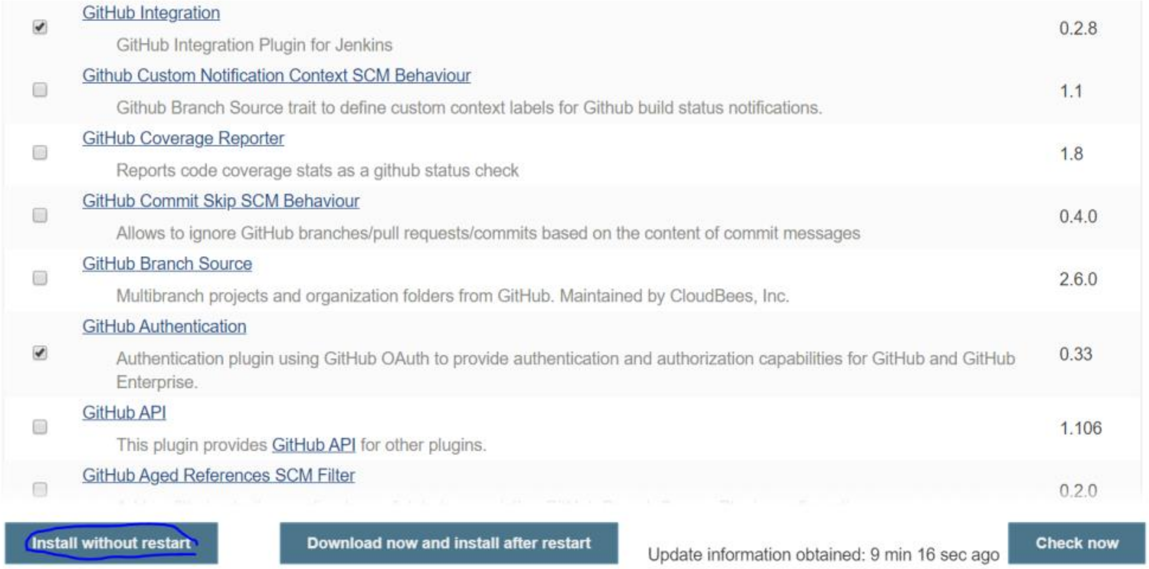
Sign in

□ Keep me signed in

Step3.1 Install GIT rpm on server terminal and Then go Jenkins console

#yum install git

@Then go Jenkins console, click on manage Jenkins → manage plugins → then install GIT plugins → select GIT,GIT HUB from available tab → click install without restart



☑	GitHub Integration GitHub Integration Plugin for Jenkins	0.2.8
□	Github Custom Notification Context SCM Behaviour Github Branch Source trait to define custom context labels for Github build status notifications.	1.1
□	GitHub Coverage Reporter Reports code coverage stats as a github status check	1.8
□	GitHub Commit Skip SCM Behaviour Allows to ignore GitHub branches/pull requests/commits based on the content of commit messages	0.4.0
□	GitHub Branch Source Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc.	2.6.0
☑	GitHub Authentication Authentication plugin using GitHub OAuth to provide authentication and authorization capabilities for GitHub and GitHub Enterprise.	0.33
□	GitHub API This plugin provides GitHub API for other plugins.	1.106
□	GitHub Aged References SCM Filter	0.2.0

Install without restart
Download now and install after restart
Update information obtained: 9 min 16 sec ago
Check now

@@Go back to manage plugins → Global Tool Configuration → in GIT (specify the GIT binary path)@@

Step 3.2:- Install Maven on jenkins server and point location to Jenkins plugins.

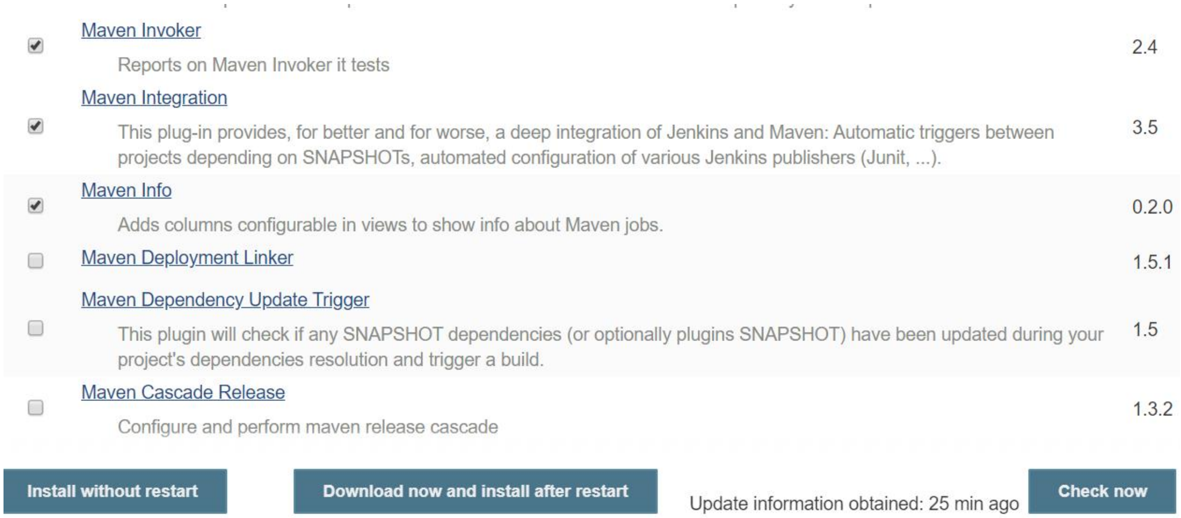
URL :- https://maven.apache.org/download.cgi

```
#wget http://apachemirror.wuchna.com/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz .
#mv apache-maven-3.6.3 maven
```

```
#vim .bash_profile
JAVA_HOME=/usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.242.b08-0.el7_7.x86_64
M2_HOME=/opt/maven
M2=/opt/maven/bin
PATH=$PATH:$HOME/bin:$JAVA_HOME:$M2_HOME:$M2
```

```
#source .bash_profile
```

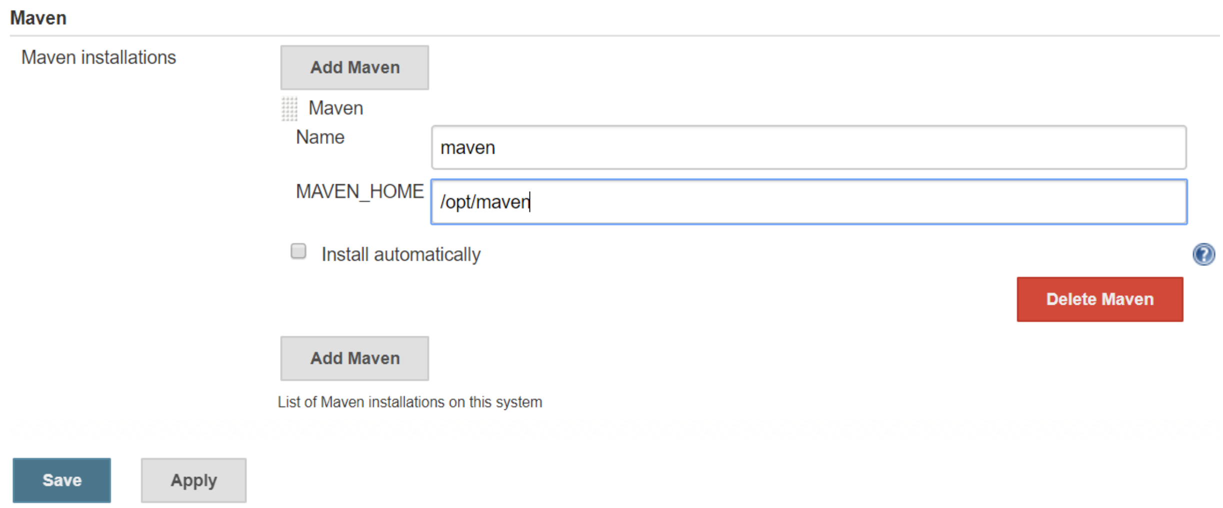
@@Install maven plugins on Jenkins console. (Maven invoker, Maven Integration, Maven Info)@@



☑	Maven Invoker Reports on Maven Invoker it tests	2.4
☑	Maven Integration This plug-in provides, for better and for worse, a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTs, automated configuration of various Jenkins publishers (Junit, ...).	3.5
☑	Maven Info Adds columns configurable in views to show info about Maven jobs.	0.2.0
□	Maven Deployment Linker	1.5.1
□	Maven Dependency Update Trigger This plugin will check if any SNAPSHOT dependencies (or optionally plugins SNAPSHOT) have been updated during your project's dependencies resolution and trigger a build.	1.5
□	Maven Cascade Release Configure and perform maven release cascade	1.3.2

Update information obtained: 25 min ago

@Now Point then Maven path from Jenkins console
Manage Jenkins → Global Tool Configuration → Maven (Add maven → [name & maven_home]



Maven

Maven installations

Maven

Name: maven

MAVEN_HOME: /opt/maven

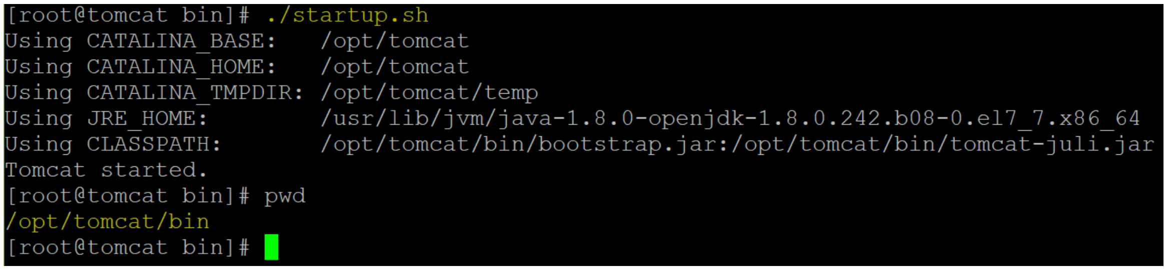
□ Install automatically

List of Maven installations on this system

Step 4:- Install tomcat service by following below steps.

URL:- Go to this link http://tomcat.apache.org/download-80.cgi and right-click copy the hyper link, then go to server and download the package via wget.

```
#cd /opt
#wget <paste the copied hyperlink>
#tar -zxvf apache-tomcat-8.5.53.tar.gz
#mv apache-tomcat-8.5.53 tomcat
#cd /opt/tomcat/bin
#./startup.sh  → start the service
```

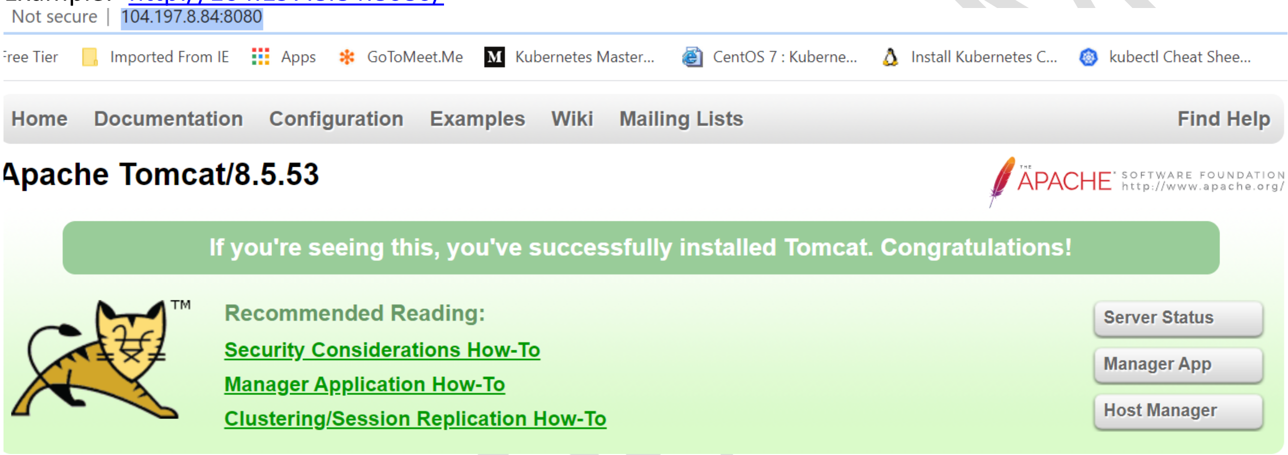


```
[root@tomcat bin]# ./startup.sh
Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b08-0.el7_7.x86_64
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Tomcat started.
[root@tomcat bin]# pwd
/opt/tomcat/bin
[root@tomcat bin]# 
```

@Now try to access tomcat server via port 8080

Note: - make sure to allow firewall rule 8080 port on GCP.

Example:- http://104.197.8.84:8080/



Apache Tomcat/8.5.53

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

Recommended Reading:

Security Considerations How-To

Manager Application How-To

Clustering/Session Replication How-To

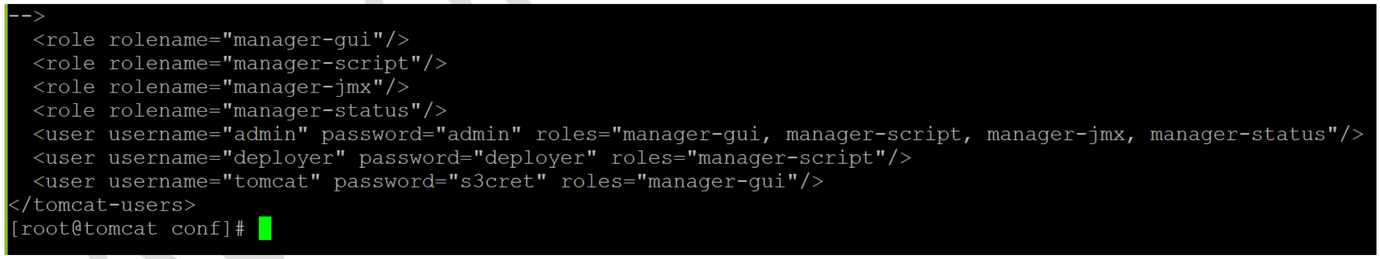
Step 4.1:- enable tomcat user credentials and allow external ip's to access the tomcat service.

#/opt/tomcat/conf

#cp -p tomcat-users.xml tomcat-users.xml.bkp

#vim tomcat-users.xml

#cat tomcat-users.xml

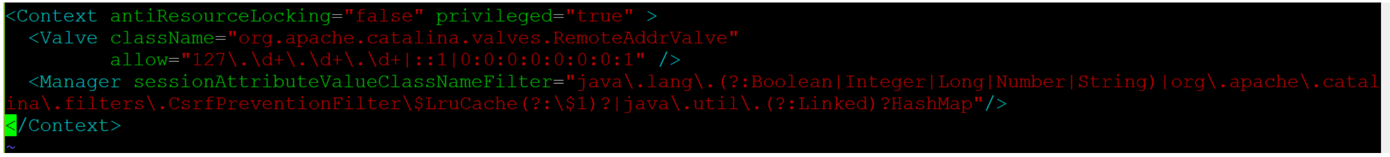


```
-->
  <role rolename="manager-gui"/>
  <role rolename="manager-script"/>
  <role rolename="manager-jmx"/>
  <role rolename="manager-status"/>
  <user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>
  <user username="deployer" password="deployer" roles="manager-script"/>
  <user username="tomcat" password="s3cret" roles="manager-gui"/>
</tomcat-users>
[root@tomcat conf]# 
```

#cd /opt/tomcat/webapps/manager/META-INF

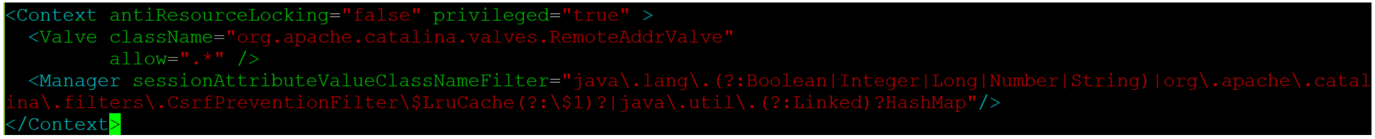
# vim context.xml

Before:-



```
<Context antiResourceLocking="false" privileged="true" >
  <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>
~
```

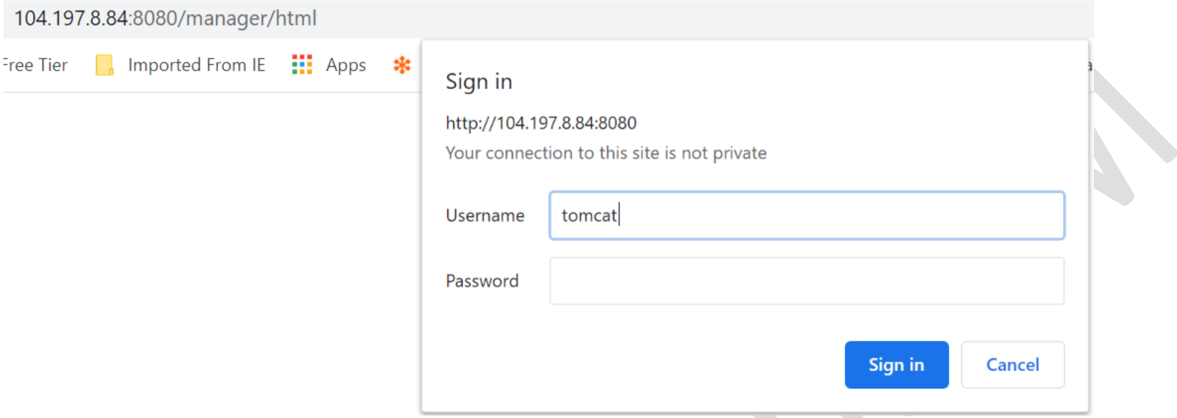
After:- replaced the content on allow”.*”



```
<Context antiResourceLocking="false" privileged="true" >
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
         allow=".*" />
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|String)|org\.apache\.catalina\.filters\.CsrfPreventionFilter\$LruCache(?:\$1)?|java\.util\.(?:Linked)?HashMap"/>
</Context>
```

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

@@Try to access with credentials@@



104.197.8.84:8080/manager/html

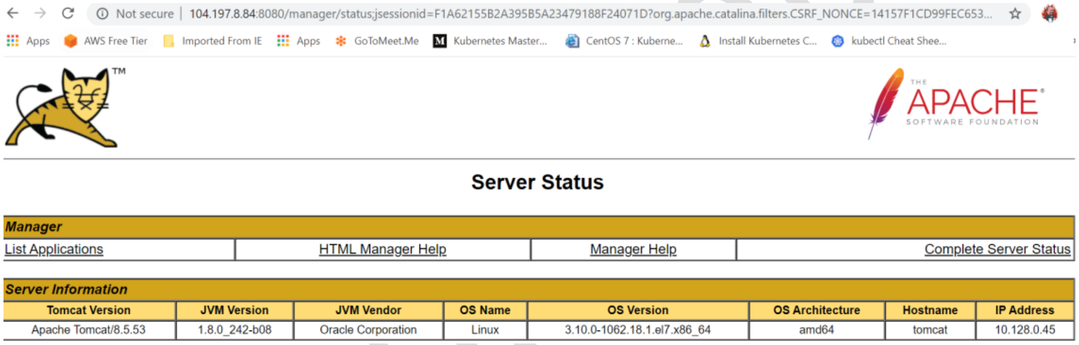
Sign in

http://104.197.8.84:8080

Your connection to this site is not private

Username: tomcat

Password:



Not secure | 104.197.8.84:8080/manager/status;jsessionid=F1A62155B2A395B5A23479188F24071D?org.apache.catalina.filters.CSRF_NONCE=14157F1CD99FEC653...

Server Status

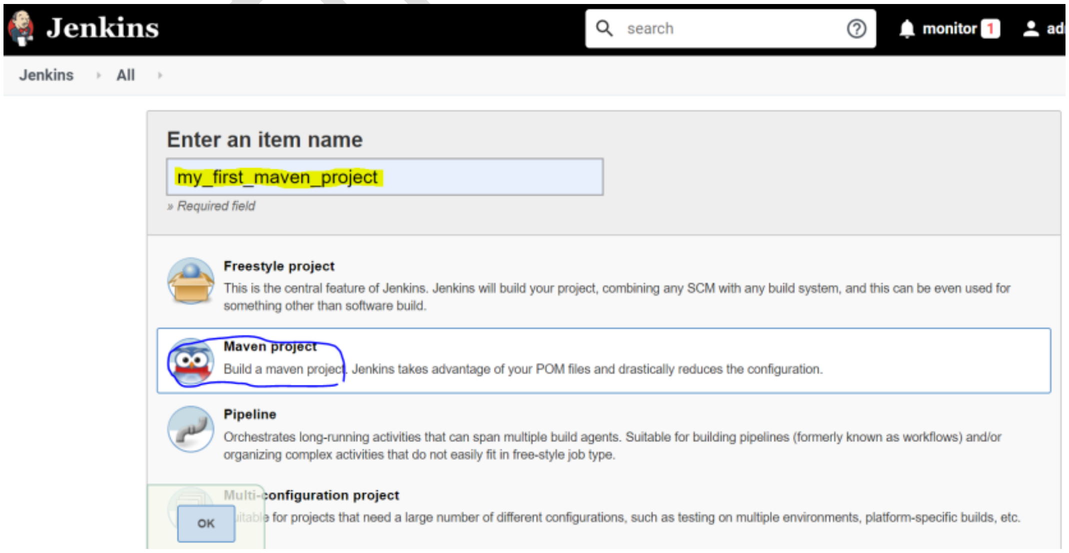
Manager			
List Applications	HTML Manager Help	Manager Help	Complete Server Status

Server Information

Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address
Apache Tomcat/8.5.53	1.8.0_242-b08	Oracle Corporation	Linux	3.10.0-1062.18.1.el7.x86_64	amd64	tomcat	10.128.0.45

Note:- Now Tomcat server is ready, next approach with Jenkins CI CD.

Step 5:- Go to Jenkins console and create new item CI/CD project.



Enter an item name

my_first_maven_project

» 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.

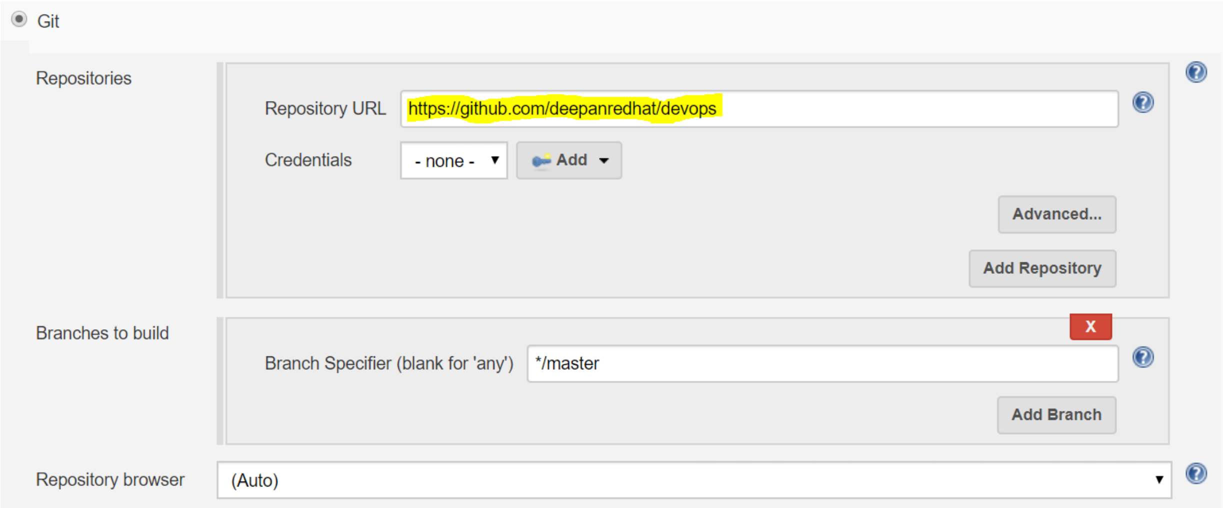
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

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

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

@Enter the GIT repo details@

# you can fork from this repo https://github.com/vsknalli/devops to your GIT hub ID.



● Git

Repositories

Repository URL: https://github.com/deepanredhat/devops

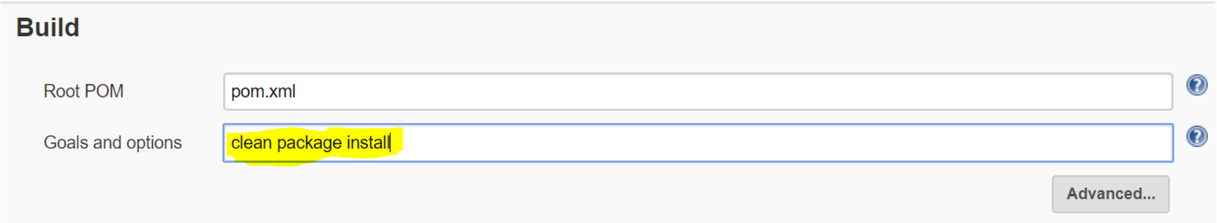
Credentials: - none -

Branches to build

Branch Specifier (blank for 'any'): */master

Repository browser: (Auto)

@Enter the build details@



Build

Root POM: pom.xml

Goals and options: clean package install

@@Click on Save and Apply → then from → work space select →build now

Console Output:-

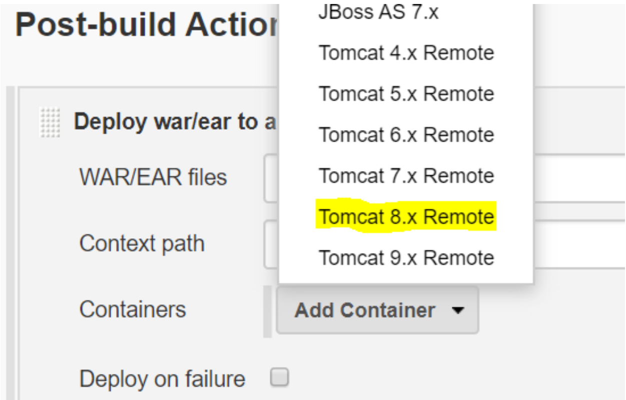
```
[JENKINS] Archiving /var/lib/jenkins/workspace/my_first_maven_project/webapp/pom.xml to com.example.maven-
project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/my_first_maven_project/webapp/target/webapp.war to com.example.maven-
project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.war
[JENKINS] Archiving /var/lib/jenkins/workspace/my_first_maven_project/server/pom.xml to com.example.maven-
project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/my_first_maven_project/server/target/server.jar to com.example.maven-
project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.jar
[JENKINS] Archiving /var/lib/jenkins/workspace/my_first_maven_project/pom.xml to com.example.maven-project/maven-
project/1.0-SNAPSHOT/maven-project-1.0-SNAPSHOT.pom
channel stopped
Finished: SUCCESS
```

Note: - CI is success.

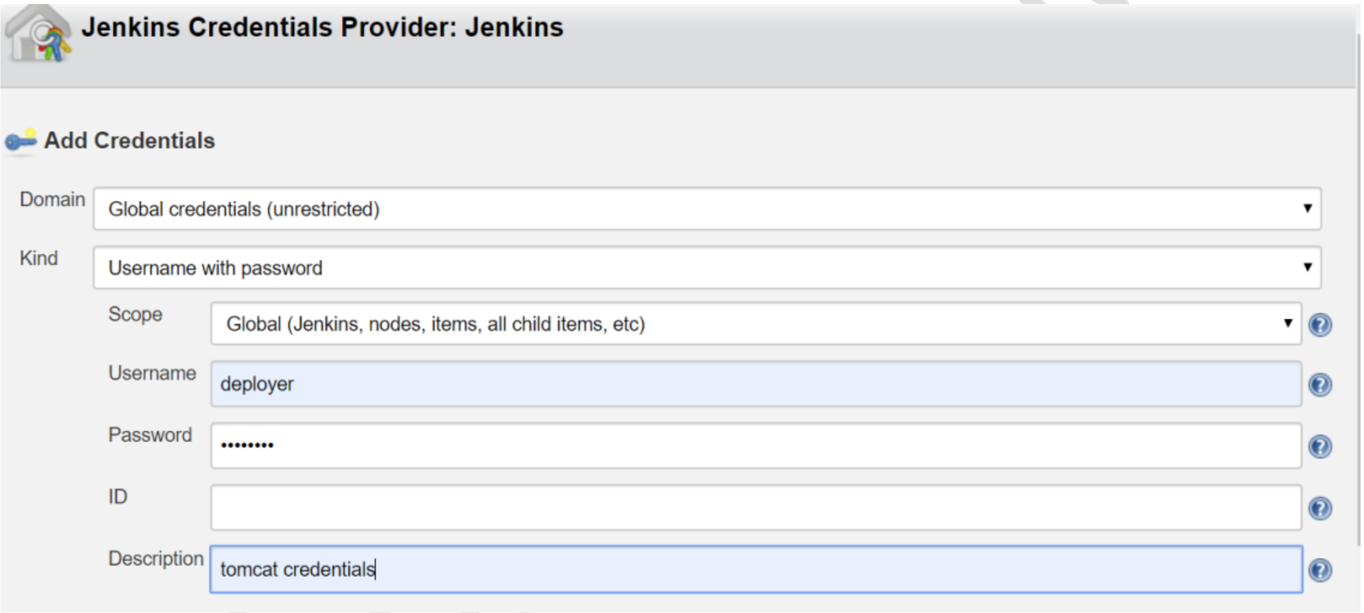
Step 5.1:- Now we are going to deploy the artifact on tomcat server.

@@Then go manage Jenkins → manage plugins → select available tab → install (deploy to container)

@@Go to dashboard → select project → click on configure → drag down → select Post build Actions → then select (deploy war/ear to a container) →click on add container → select tomcat version



@Next click on Credentials → Add tomcat script execution credentials on the tab.



Jenkins Credentials Provider: Jenkins

Add Credentials

Domain: Global credentials (unrestricted)

Kind: Username with password

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

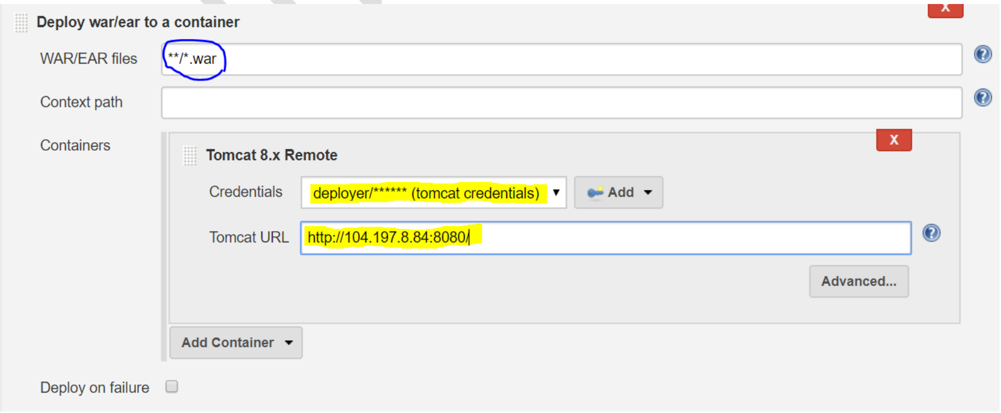
Username: deployer

Password: ••••••••

ID:

Description: tomcat credentials

Step 5.2:- Click on post build action and enter the credentials (user name :- deployer and password deployer)



Deploy war/ear to a container

WAR/EAR files: **/*.war

Context path:

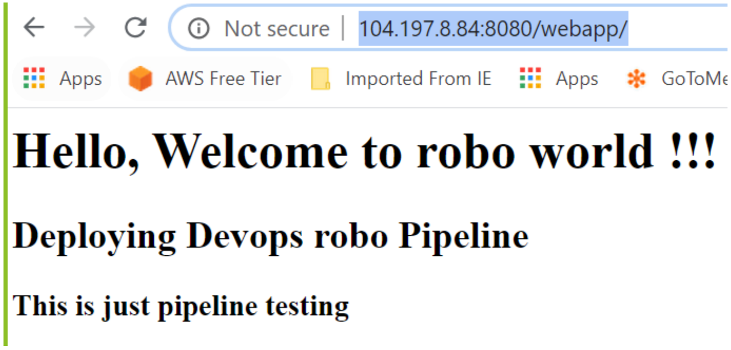
Containers: Tomcat 8.x Remote

Credentials: deployer/****** (tomcat credentials)

Tomcat URL: http://104.197.8.84:8080/

@Click save and apply → click on build now (to view output, select console output).

@Try to open the URL http://104.197.8.84:8080/webapp/



Not secure | 104.197.8.84:8080/webapp/

Hello, Welcome to robo world !!!

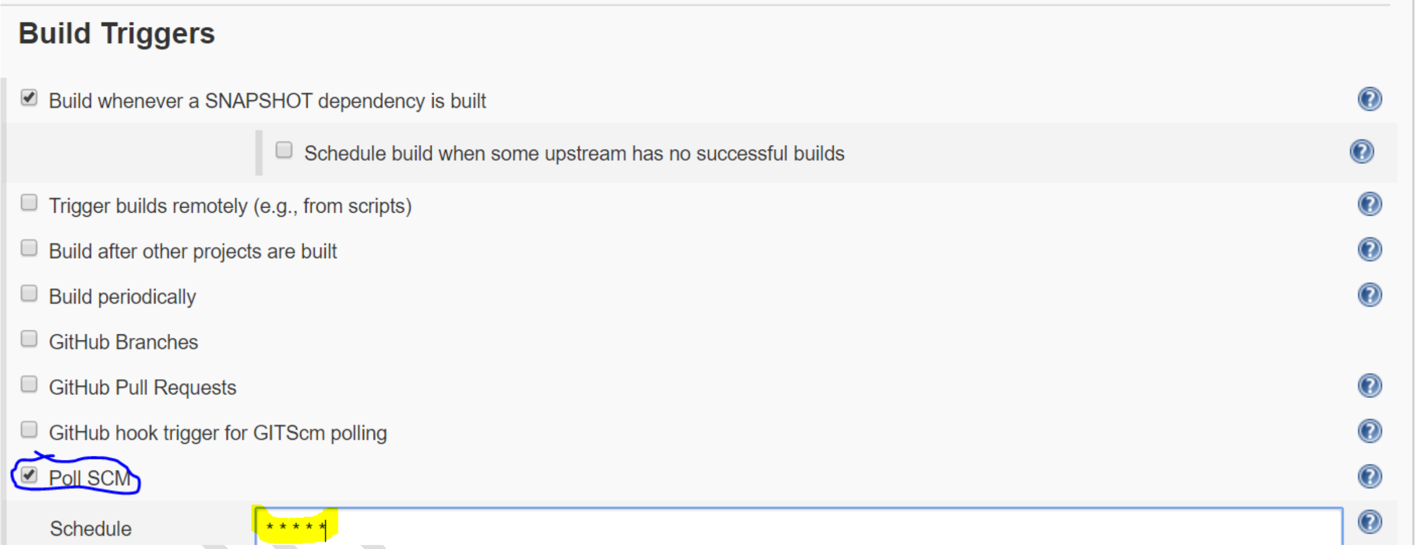
Deploying Devops robo Pipeline

This is just pipeline testing

Note: - Successfully CI and CD practice has been completed.

Step 6:-Now going to test with automate process CI & CD after GIT commit with new code.

@Go to Jenkins console → select our project from dashboard and click on configure → select Build Triggers → select Poll SCM → save



Build Triggers

☑ Build whenever a SNAPSHOT dependency is built

□ Schedule build when some upstream has no successful builds

□ Trigger builds remotely (e.g., from scripts)

□ Build after other projects are built

□ Build periodically

□ GitHub Branches

□ GitHub Pull Requests

□ GitHub hook trigger for GITScm polling

☑ Poll SCM

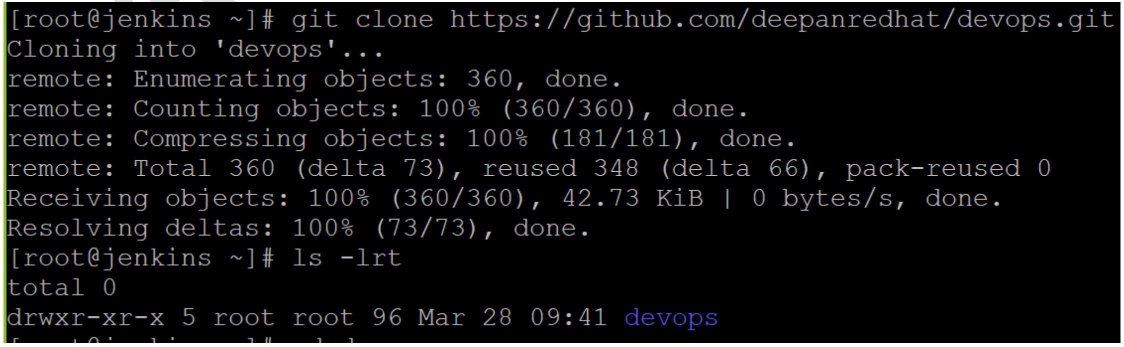
Schedule * * * * *

@Go to Jenkins server terminal and clone the git repo first.

# you can fork from this repo https://github.com/vsknalli/devops to your GIT hub ID.

@Below Is my GIT repo, you better to fork devops repo from above link.

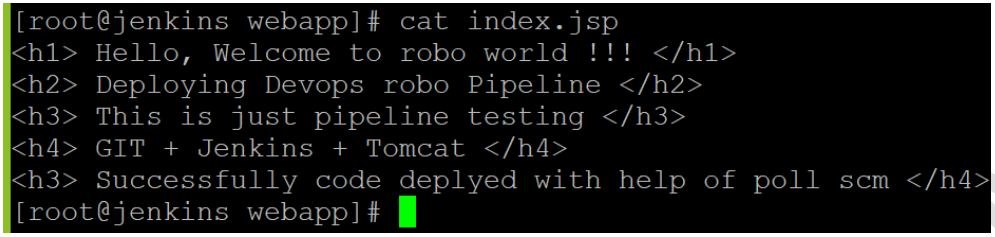
git clone https://github.com/deepanredhat/devops.git



```
[root@jenkins ~]# git clone https://github.com/deepanredhat/devops.git
Cloning into 'devops'...
remote: Enumerating objects: 360, done.
remote: Counting objects: 100% (360/360), done.
remote: Compressing objects: 100% (181/181), done.
remote: Total 360 (delta 73), reused 348 (delta 66), pack-reused 0
Receiving objects: 100% (360/360), 42.73 KiB | 0 bytes/s, done.
Resolving deltas: 100% (73/73), done.
[root@jenkins ~]# ls -lrt
total 0
drwxr-xr-x 5 root root 96 Mar 28 09:41 devops
```

#cd /root/devops/webapp/src/main/webapp

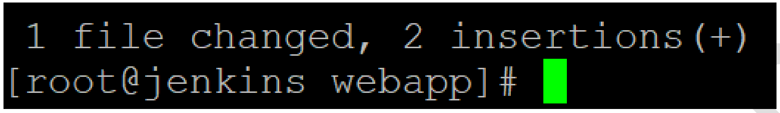
@add few lines on tomcat index file.
#vim index.jsp



```
[root@jenkins webapp]# cat index.jsp
<h1> Hello, Welcome to robo world !!! </h1>
<h2> Deploying Devops robo Pipeline </h2>
<h3> This is just pipeline testing </h3>
<h4> GIT + Jenkins + Tomcat </h4>
<h3> Successfully code deplyed with help of poll scm </h4>
[root@jenkins webapp]#
```

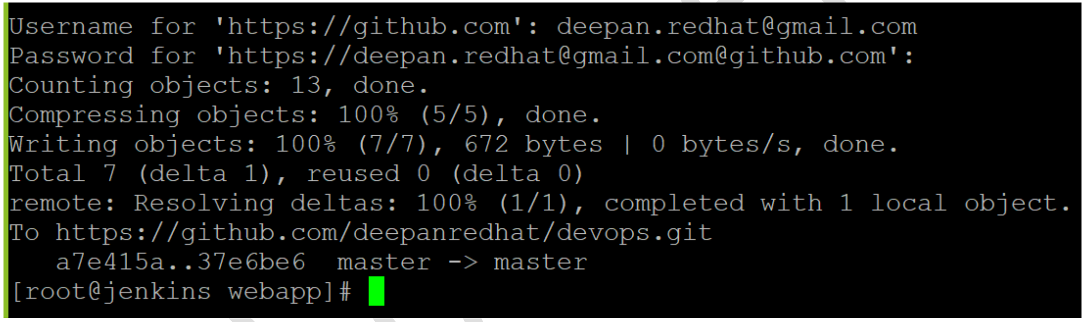
# git add .

# git commit -m "intial commit"



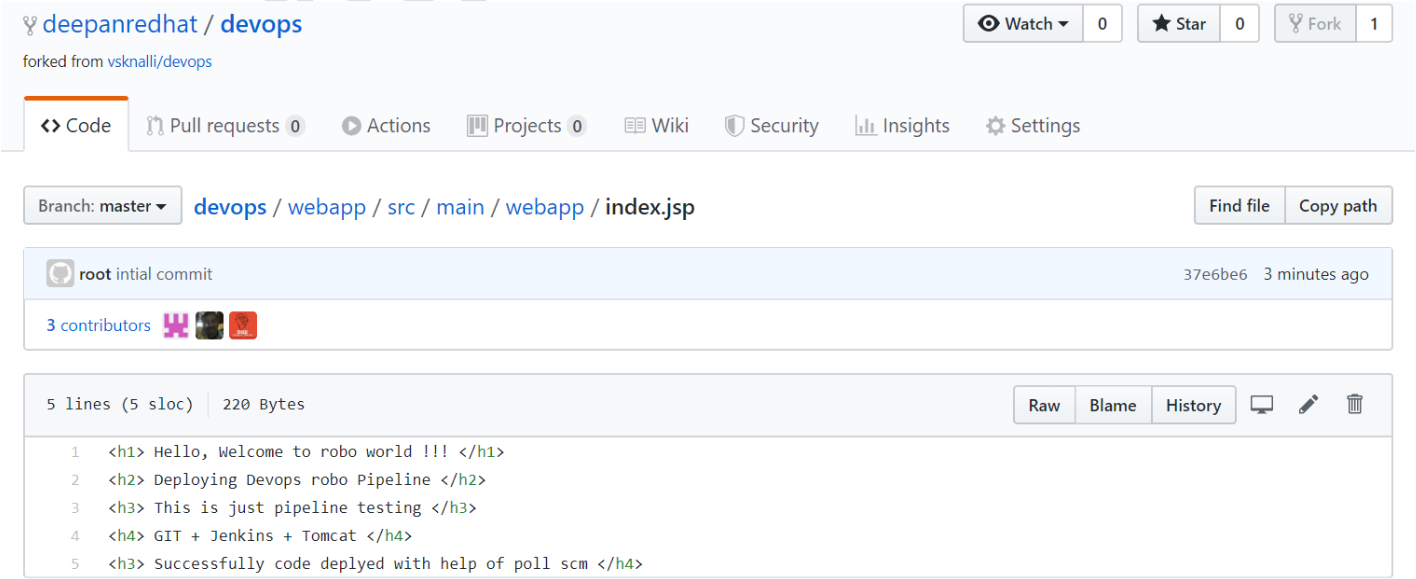
```
 1 file changed, 2 insertions(+)
[root@jenkins webapp]#
```

# git push



```
Username for 'https://github.com': deepan.redhat@gmail.com
Password for 'https://deepan.redhat@gmail.com@github.com':
Counting objects: 13, done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 672 bytes | 0 bytes/s, done.
Total 7 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/deepanredhat/devops.git
   a7e415a..37e6be6  master -> master
[root@jenkins webapp]#
```

@@You can verify on GIT hub repository, whether changes have been done or not@@



deepanredhat / devops

forked from vsknalli/devops

Branch: master devops / webapp / src / main / webapp / index.jsp

root intial commit 37e6be6 3 minutes ago

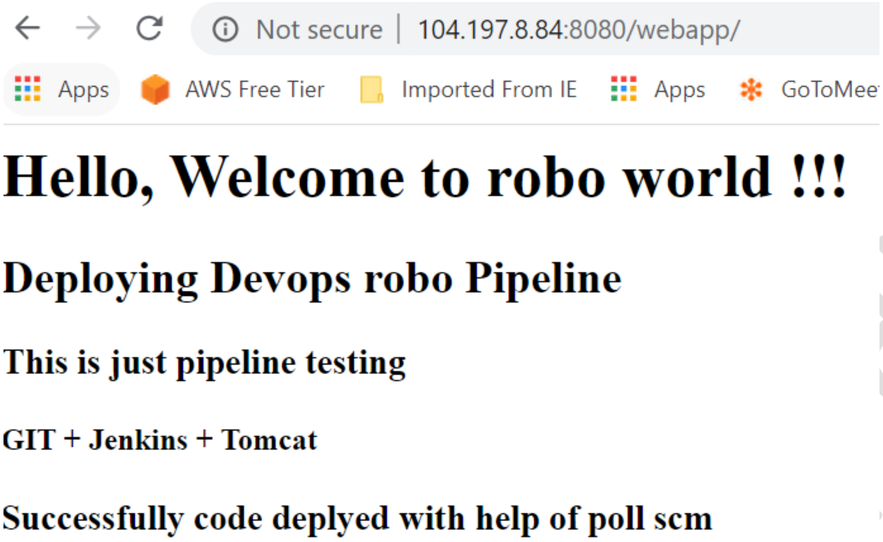
3 contributors

5 lines (5 sloc) 220 Bytes

```
<h1> Hello, Welcome to robo world !!! </h1>
<h2> Deploying Devops robo Pipeline </h2>
<h3> This is just pipeline testing </h3>
<h4> GIT + Jenkins + Tomcat </h4>
<h3> Successfully code deplyed with help of poll scm </h4>
```

@Now go to Jenkins console output and see the build status of your project.

@Then refresh tomcat web page and you can able to see the content which committed newly on GIT.



Hello, Welcome to robo world !!!

Deploying Devops robo Pipeline

This is just pipeline testing

GIT + Jenkins + Tomcat

Successfully code deplyed with help of poll scm

Note:- Automate process after code commit is working perfectly with help of Jenkins poll scm.