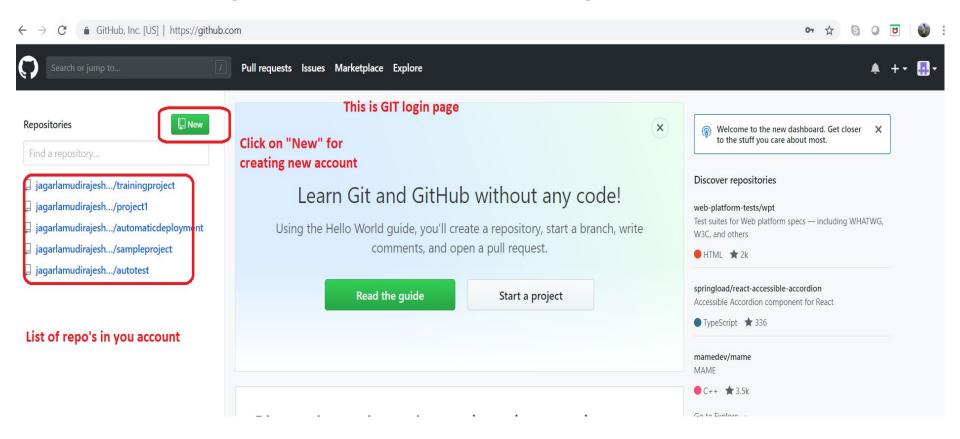
# Jenkins

Continuous Integration Continuous Deployment

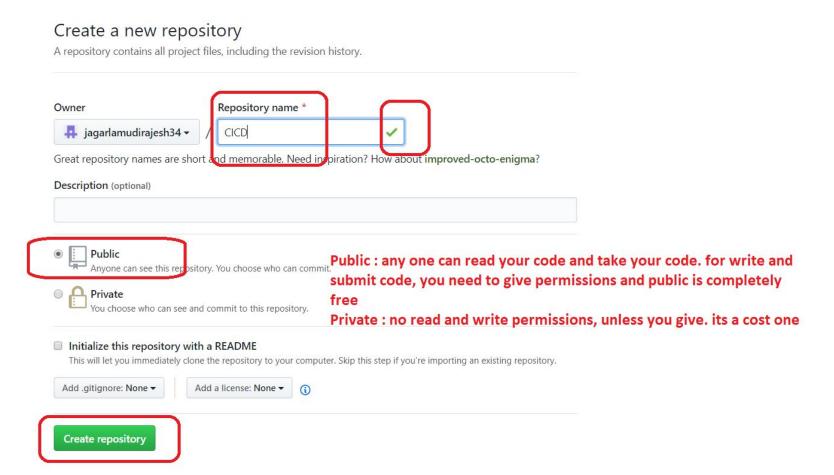
### Steps Wise Process

- 1. Create an GITHUB account  $\rightarrow$  Login GITHUB  $\rightarrow$  create a repository  $\rightarrow$  import repository from another repository  $\rightarrow$  copy the URL of your repository.
- Login to instance via putty ---> download JDK & tomcat and set class path and update path.
- 3. Login to instance via putty configure GIT
- 4. Login to Jenkins as admin --> manage Jenkins --> Global Tool Configuration --> add JDK installations, GIT Installations & Maven Installations
- 5. Go to login dashboard → New Item → Enter an item name --> free style project ---> ok

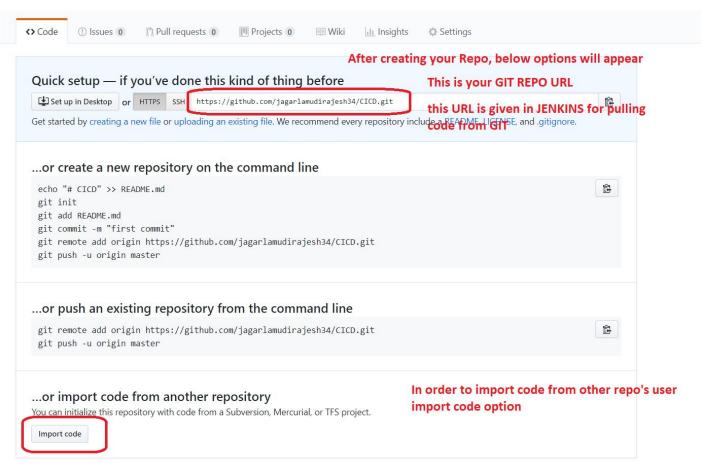
## After Creating GITHUB account → Login to GITHUB Account



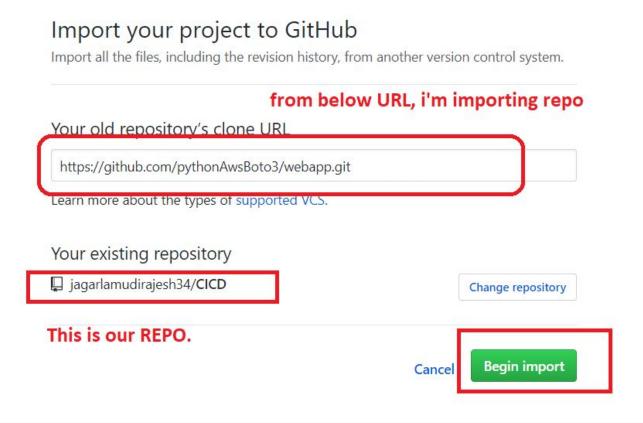
### Enter your Repository name



## Importing code from Other Repo's



## Provide URL of repo from which you are importing



## You will receive mail and importing status is shown below

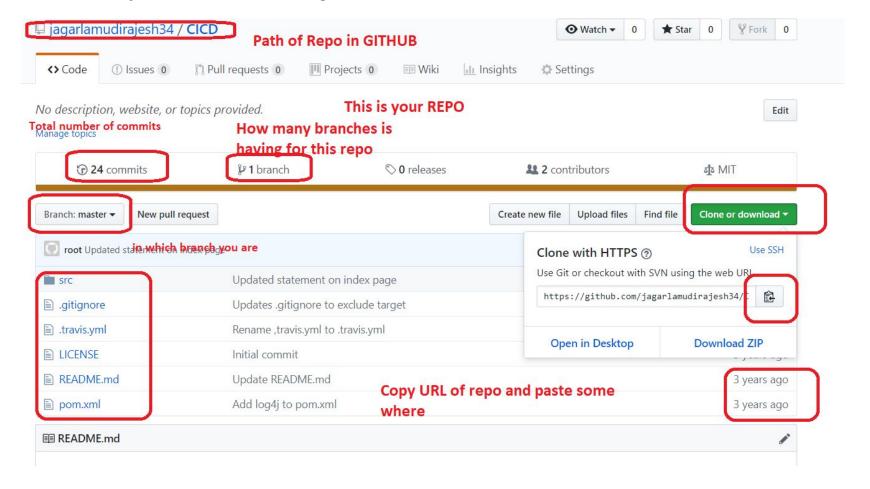
### Preparing your new repository

There is no need to keep this window open, we'll email you when the import is done.

📘 jagarlamudirajesh34/CICD

✓ Importing complete! Your new repository jagarlamudirajesh34/CICD is ready.

### Go back to your GITHUB login dashboard and check the REPO and code



Login to Instance via putty and download JDK & tomcat, set path for both JDK and tomcat.

Go to webapps dir of tomcat and download Jenkins.war and start tomcat

This process is covered in previous class and for any info, please revise previous class materials

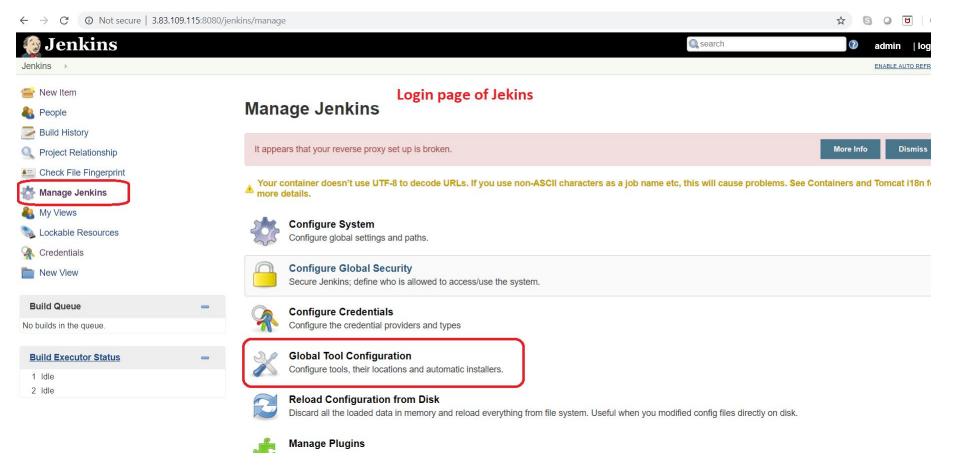
### Add user in tomcat-users.xml which is in conf dir of tomcat

```
root@ip-1/2-31-93-24: /opt/tomcat//conf
 lser username= manager passworq= manager roles= manager
 tomcat-users>
```

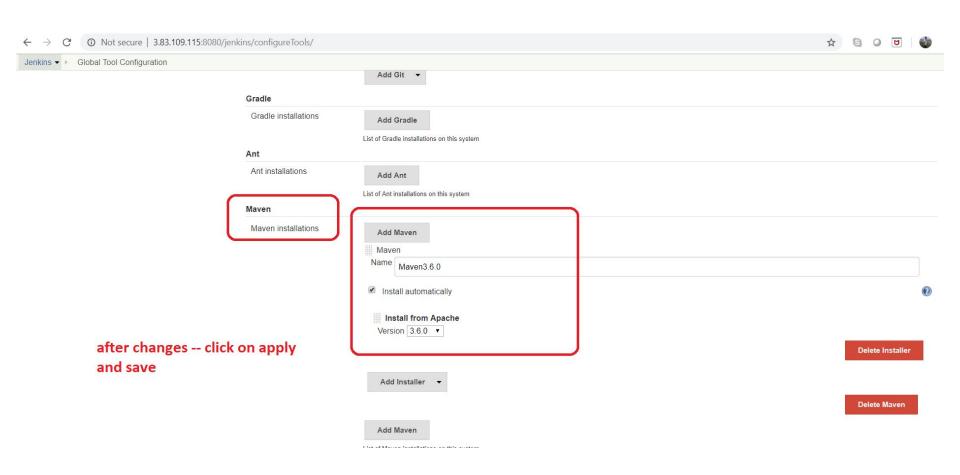
### Installing GIT Instance - use below commands to install

```
Login to instance and switch to root user and update instance
#sudo su root
#cd /
#apt-get update
#apt-get install git-core -y
#git --version
#git config --global user.name <your git account name>
#git config --global user.email <your git login mail>
#git config --list
#mkdir repos
#cd repos
#git init
#ls -a
#git clone <your git repo url>
Eg
#git clone https://github.com/jagarlamudirajesh34/CICD.git
#ls
#cd <repo name> eg cd CICD
#cd CICD/src/main/webapp
```

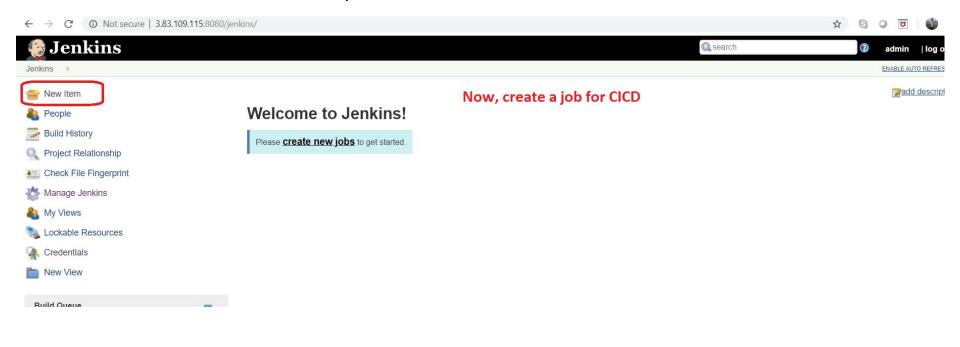
## Login to Jenkins and configure JDK, GIT MAVEN



Gradle



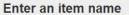
### Now, create a JOB for CICD











#### AutomaticDeployment



#### Freestyle project

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



#### 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.



#### Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



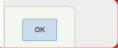
#### GitHub Organization

Scans a GitHub organization (or user account) for all repositories matching some defined markers.

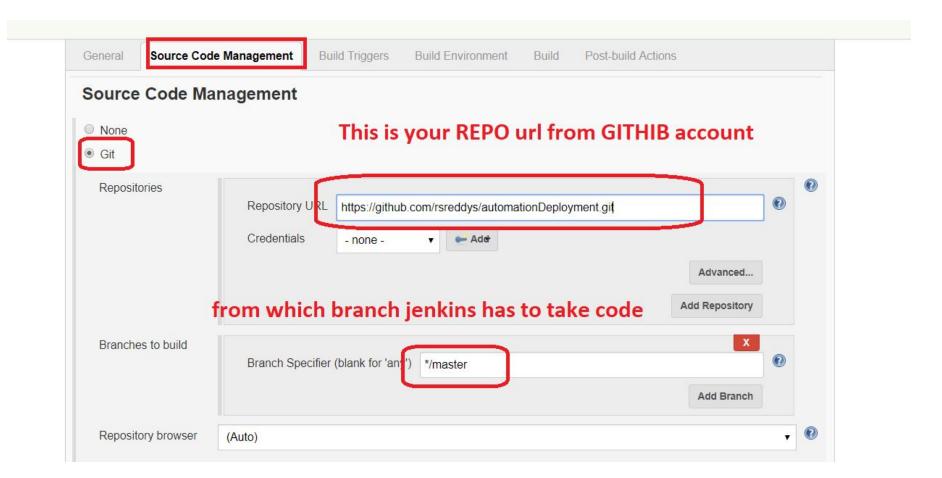


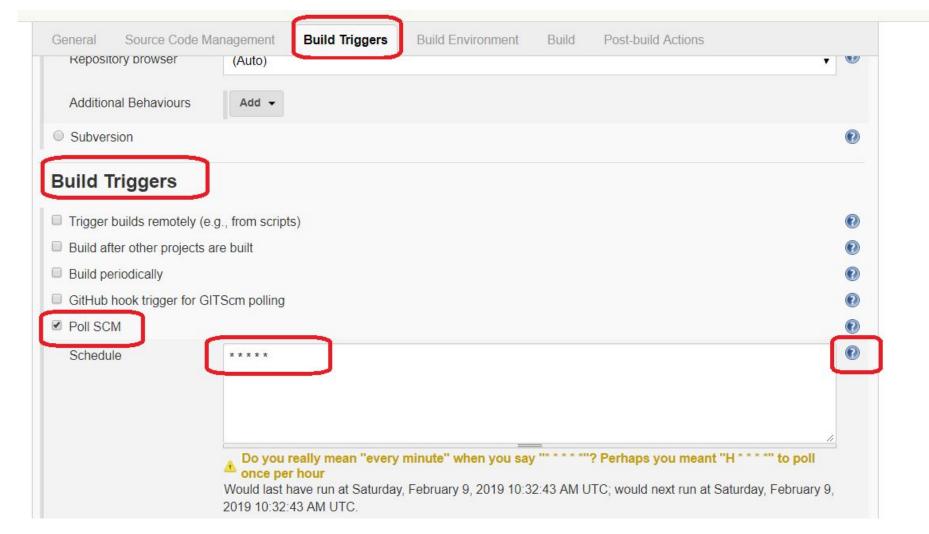
#### **Multibranch Pipeline**

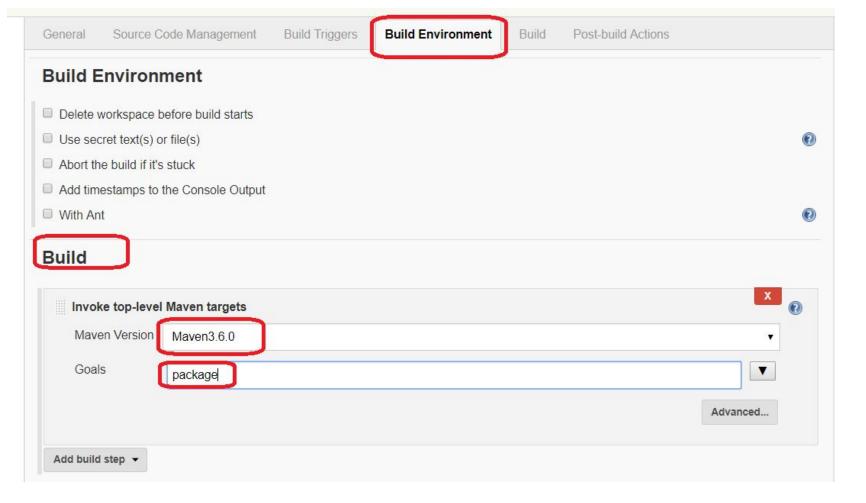
Creates a set of Pipeline projects according to detected branches in one SCM repository.



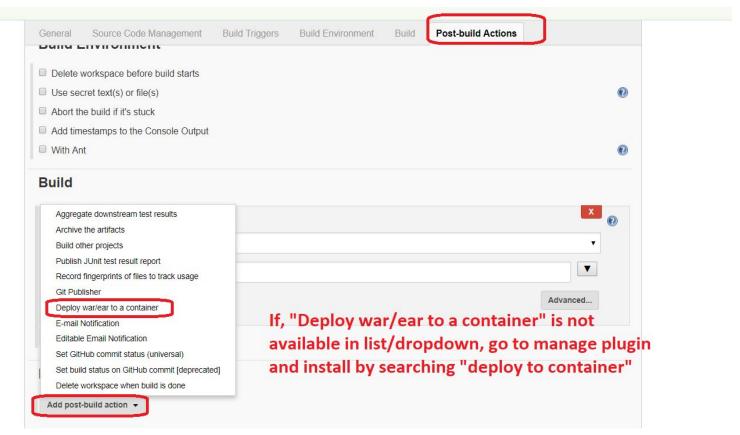
### Now, configure GIT, job triggering details, Tomcat details



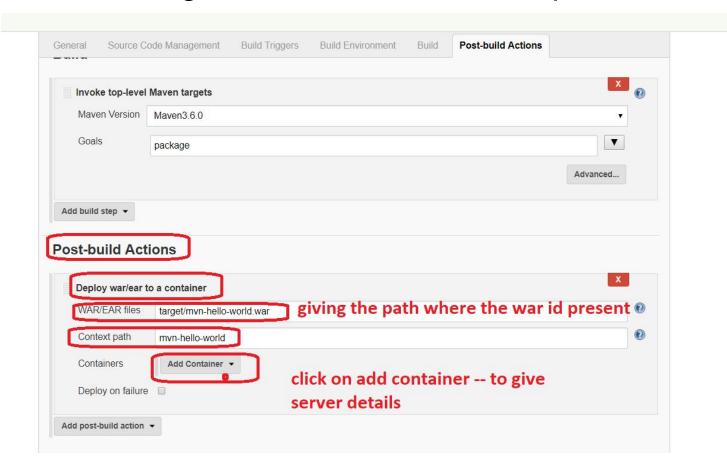




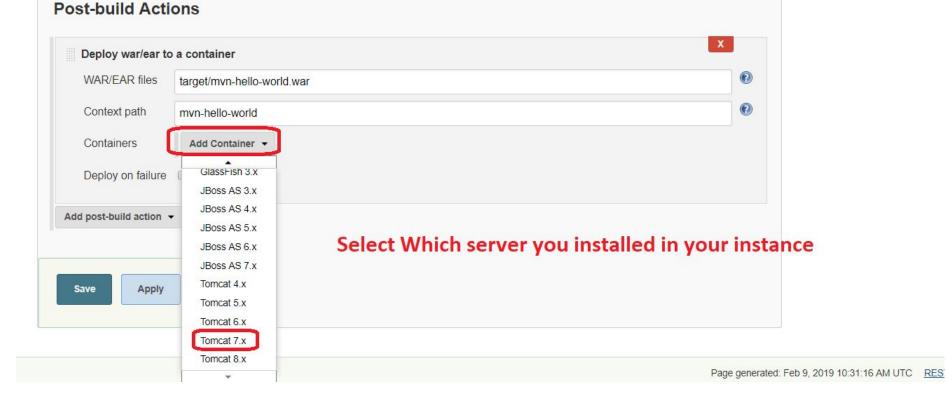
# Configuring tomcat



### Pointing where the .war / build is present



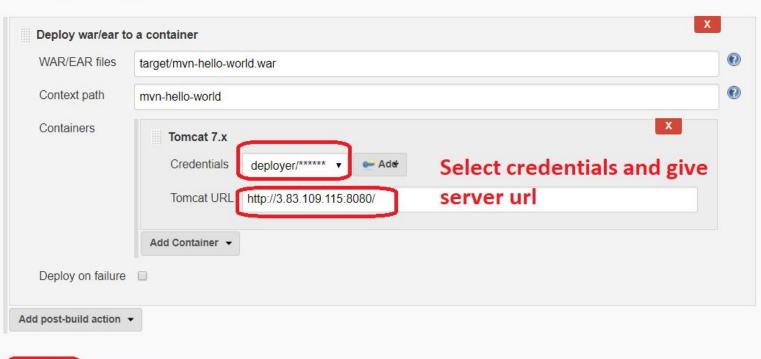
### **Giving Server Details**



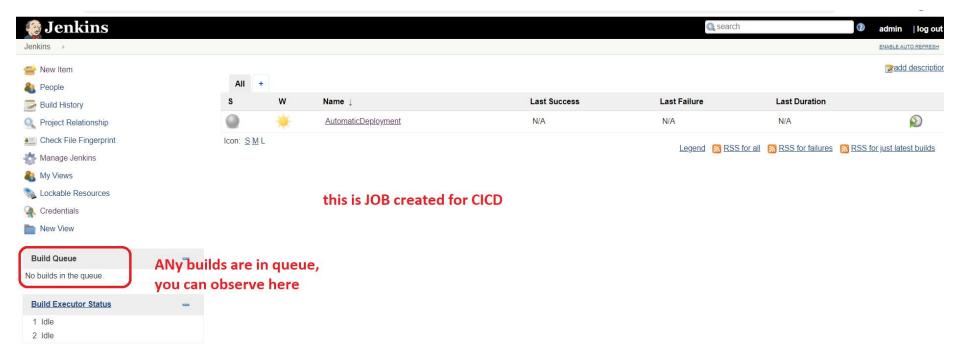
#### Post-build Actions × Deploy war/ear to a container WAR/EAR files target/mvn-hello-world.war Context path mvn-hello-world × Containers Tomcat 7.x Credentials - none -Tomcat URL Add Container -



### **Post-build Actions**







Goto putty → goto git repo where you configured and open index.jsp page and do some changes and add, commit and push to repo

```
root@ip-172-31-93-24:/repos/automaticdeployment/src/main/webapp# ls
index.isp WEB-INF
root@ip-172-31-93-24:/repos/automaticdeployment/src/main/webapp# vi index.j
root@ip-172-31-93-24:/repos/automaticgepioyment/src/main/wepapp# vi index.j:
root@ip-172-31-93-24:/repos/automaticdeployment/src/main/webapp# git add ind
sp
root@ip-172-31-93-24:/repos/automaticdeployment/src/main/webapp# git commit
comments"
[master cb72631] comments
1 file changed, 1 insertion(+), 1 deletion(-)
root@ip-172-31-93-24:/repos/automaticdeployment/src/main/webapp# git push o
 sername for 'https://qithub.com': jaqarlamudirajesh34
 assword for 'https://jagarlamudirajesh34@github.com':
Counting objects: 6, done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (646) 478 bytes this, your done.
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/jagarlamudirajesh34/automaticceployment.git
741022f..cb72631 master master to server root@ip-172-31-93-24:/repos/automaticdeployment/src/main/webapp#
```

Commands is given in next slide if you not get from above snap

Go to dir in which index.jsp is available #vi index.jsp
Change some code #git add index.jsp
#git commit -m "Comments"
#git push origin master

Provide your git credentials

For cross check, go to webapps dir of tomcat and see the war is deployed or not.

Open tomcat in browser: <ip>:8080/mvn-hello-world

```
riting objects: 100% (6/6), 478 bytes | 0 bytes/s, done.

otal 6 (delta 2), reused 0 (delta 0)

emote: Resolving deltas: 100% (2/2), completed with 2 local objects.

o https://github.com/jagarlamudirajesh34/automaticdeployment.git
    741022f..cb72631 master -> master

oot@ip-172-31-93-24:/repos/automaticdeployment/src/main/webapp# ls

ndex.jsp WEB-INF

oot@ip-172-31-93-24:/repos/automaticdeployment/src/main/webapp# cd /

oot@ip-172-31-93-24:/# cd opt/tomcat7/webapps/
oot@ip-172-31-93-24:/opt/tomcat7/webapps# ls

ocs examples host-manager jenkins jenkins.war manager

oot@ip-172-31-93-24:/opt/tomcat7/webapps#
```