IndustryGradeProject-2:

Project executed using following OS configuration:

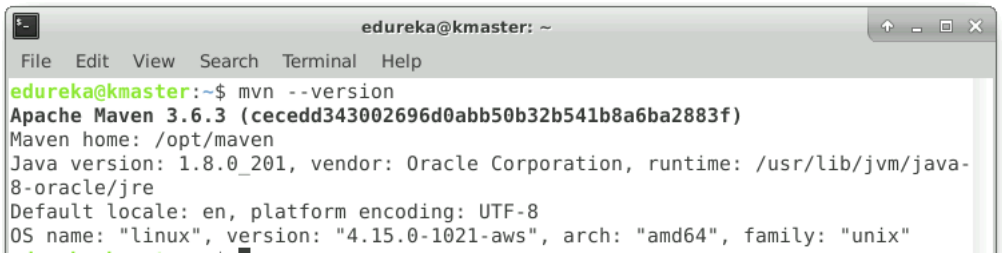
```
edureka@kmaster:~$ cat /etc/os-release
NAME="Ubuntu"
VERSION="18.04.3 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.3 LTS"
VERSION_ID="18.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-poli
cy"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic
```

Git: Git is free and open-source software for distributed version control, tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development

1. Create new repository in github and push the initial code to repository
2. https://github.com/yogeshk04/projects.git
    a. Project is in following directory https://github.com/yogeshk04/projects/tree/master/XYZ_Technologies
    b. Copy the industry grade repo from Edureka industry grade project section
    c. Go to directory and execute following commands
        i. *: ~$ git init*
        ii. *: ~$ git add .*
        iii. *: ~$ git commit -m "Initial project code commit"*
    d. Configure git for the first time to github (*Note – you need to have github or gitlab account)*
    e. Execute following commands
        i. *: ~$ git config –global user.name "yogeshk04"*
        ii. *: ~$ git config --global user.email "yogeshk04@gmail.com"*
        iii. Creating a personal access token to access the github – *Personal access token (PAT) are an alternative to using passwords for authentication to GitHub when using the GitHub API or the command line.*
    f. Push local repo to GitHub
        i. *: ~$ git remote add origin https://github.com/yogeshk04/projects.git*
        ii. *: ~$ git push –set-upstream origin master*
    g. Go back to GitHub and see that the repository has been updated.

Maven: Maven is a build automation tool used primarily for Java projects. Maven can also be used to build and manage projects written in C#, Ruby, Scala, and other languages. The Maven project is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project.

1. Check maven version and java version using *mvn –version command*



2. Run the following maven build command to clean the target folder

   : ~$ mvn clean install

```
edureka@kmaster:~/projects/XYZ_Technologies$ mvn clean install
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< com.xyz:xyztech >-------------------------
[INFO] Building adminModule 1.0
[INFO] --------------------------------[ war ]---------------------------------
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.pom (3.9 kB at 2.8 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/22/maven-plugins-22.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/22/maven-plugins-22.pom (13 kB at 46 kB/s)
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ xyztech ---
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0/plexus-utils-3.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0/plexus-utils-3.0.pom (4.1 kB at 16 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/sonatype/spice/spice-parent/16/spice-parent-16.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/spice/spice-parent/16/spice-parent-16.pom (8.4 kB at 32 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0/plexus-utils-3.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0/plexus-utils-3.0.jar (226 kB at 418 kB/s)
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.6:prepare-agent (jacoco-initialize) @ xyztech ---
[INFO] argLine set to -javaagent:/home/edureka/.m2/repository/org/jacoco/org.jacoco.agent/0.8.6/org.jacoco.agent-0.8.6-runtime.jar=destfile=/home/edureka/projects/XYZ_Technologies/target/jacoco.exec
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ xyztech ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/edureka/projects/XYZ_Technologies/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ xyztech ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 3 source files to /home/edureka/projects/XYZ_Technologies/target/classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ xyztech ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/edureka/projects/XYZ_Technologies/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ xyztech ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /home/edureka/projects/XYZ_Technologies/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ xyztech ---
[INFO] Surefire report directory: /home/edureka/projects/XYZ_Technologies/target/surefire-reports
```

3. To package the project, one should run following command

   : ~$ mvn package

4. Maven Run command
5. Mvn exec:java -Dexec.mainClass=
6. Run the test cases in the project

   : ~$ mvn test

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running com.xyz.dataAccessObject.AdminDataImpTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.141 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-war-plugin:3.2.2:war (default-war) @ xyztech ---
[INFO] Packaging webapp
[INFO] Assembling webapp [xyztech] in [/home/edureka/projects/XYZ_Technologies/target/xyztech-1.0]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/edureka/projects/XYZ_Technologies/src/main/webapp]
[INFO] Webapp assembled in [1094 msecs]
[INFO] Building war: /home/edureka/projects/XYZ_Technologies/target/xyztech-1.0.war
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.6:report (jacoco-site) @ xyztech ---
[INFO] Loading execution data file /home/edureka/projects/XYZ_Technologies/target/jacoco.exec
[INFO] Analyzed bundle 'adminModule' with 2 classes
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ xyztech ---
[INFO] Installing /home/edureka/projects/XYZ_Technologies/target/xyztech-1.0.war to /home/edureka/.
m2/repository/com/xyz/xyztech/1.0/xyztech-1.0.war
[INFO] Installing /home/edureka/projects/XYZ_Technologies/pom.xml to /home/edureka/.m2/repository/c
om/xyz/xyztech/1.0/xyztech-1.0.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  21.041 s
[INFO] Finished at: 2022-12-05T12:53:04Z
[INFO] ------------------------------------------------------------------------
```

Tomcat: Apache Tomcat is a free and open-source implementation of the Jakarta Servlet, Jakarta Expression Language, and WebSocket technologies. It provides a "pure Java" HTTP web server environment in which Java code can also run. Thus, it's a Java web application server, although not a full JEE application server

Installation of Tomcat.

1. Check for the updates

   : ~$ sudo apt update
2. Check java version if already install or install the required java version

   : ~$ java --version

```
edureka@kmaster:~$ java --version
openjdk 11.0.16 2022-07-19
OpenJDK Runtime Environment (build 11.0.16+8-post-Ubuntu-0ubuntu118.04)
OpenJDK 64-Bit Server VM (build 11.0.16+8-post-Ubuntu-0ubuntu118.04, mixed mode,
 sharing)
```

   To install java use

: ~$ sudo apt install default-jdk

3. Create Tomcat user

   sudo groupadd tomcat

```
edureka@kmaster:~$ sudo groupadd tomcat
groupadd: group 'tomcat' already exists
edureka@kmaster:~$ sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
useradd: user 'tomcat' already exists
edureka@kmaster:~$
```

4. Install Tomcat on Ubuntu

   a. Download the Tomcat .tar file using following command

   : ~$ wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.68/bin/apache-tomcat-9.0.68.tar.gz

```
edureka@kmaster:~/Downloads$ wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.
68/bin/apache-tomcat-9.0.68.tar.gz
--2022-10-29 06:01:33--  https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.68/bin/ap
ache-tomcat-9.0.68.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connecte
d.
HTTP request sent, awaiting response... 200 OK
Length: 11597709 (11M) [application/x-gzip]
Saving to: 'apache-tomcat-9.0.68.tar.gz'

apache-tomcat-9.0.6 100%[===================>]  11.06M  --.-KB/s    in 0.06s

2022-10-29 06:01:33 (197 MB/s) - 'apache-tomcat-9.0.68.tar.gz' saved [11597709/1
1597709]
```

   b. Extract the .tar file

   : ~$ tar -xzf apache-tomcat-9.0.68.tar.gz

   : ~$ ls

```
edureka@kmaster:~/Downloads$ tar -xzf apache-tomcat-9.0.68.tar.gz
edureka@kmaster:~/Downloads$ ls
apache-tomcat-9.0.68  apache-tomcat-9.0.68.tar.gz
```

   c. Create tomcat directory into opt folder

   : ~$ suod mkdir /opt/tomcat

   d. Move all the content to that folder

   : ~$ sudo mv apache-tomcat-9.0.68/ /opt/tomcat/*

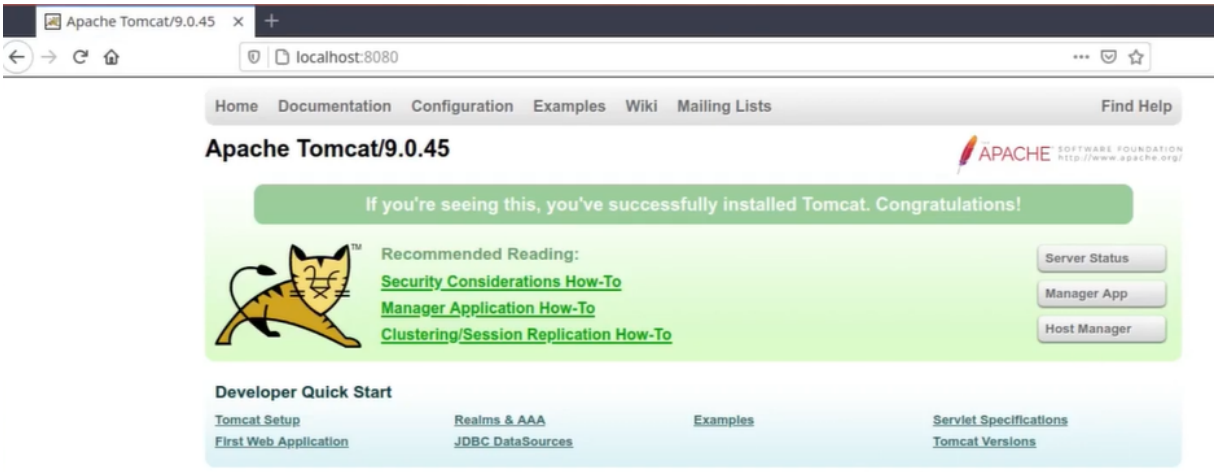
   e. Change directory to /opt/tomcat/bin

   f. Start the tomcat server using startup script file

   : ~$ sudo ./startup.sh

```
edureka@kmaster:/opt/tomcat/bin$ sudo ./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-8-oracle/jre
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli
.jar
Using CATALINA_OPTS:
Tomcat started.
```

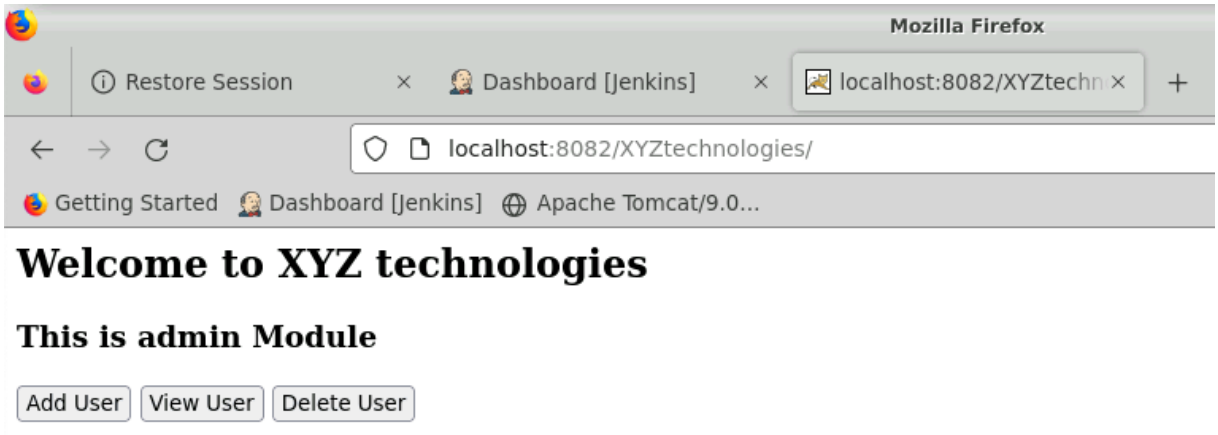
5. Access tomcat using http:localhost:8080



6. Deploy the .war file to Tomcat manually
7. Copy the WAR file you have created for XYZtechnologies to /opt/tomcat/webapps

   ~$ cp XZ`YZtechnologies-1.0.war /opt/tomcat/webapps/

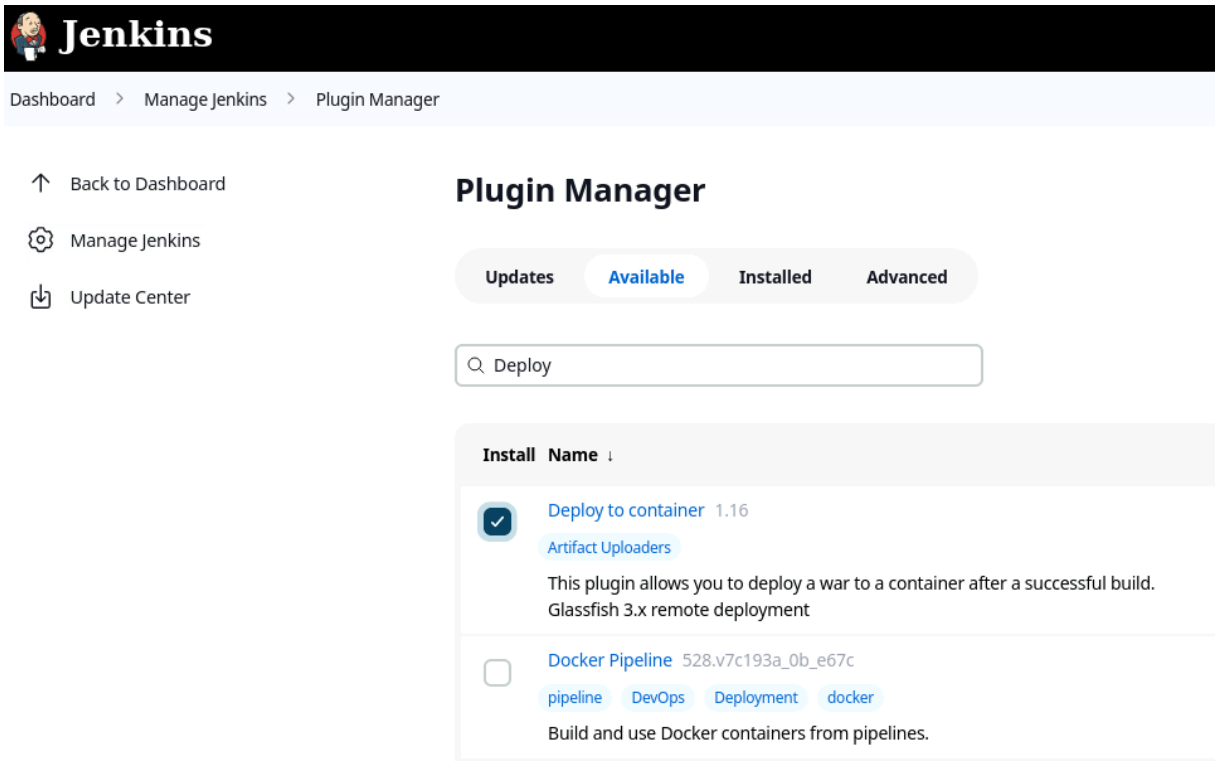
8. Start the Tomcat server.
9. In the address area of the browser, type *http://localhost:8080/XYZtechnologies-1.0* ***(Note: here I change the port to 8082 as 8080 was used by Jenkins server)***
10. The output of your application page will be displayed as



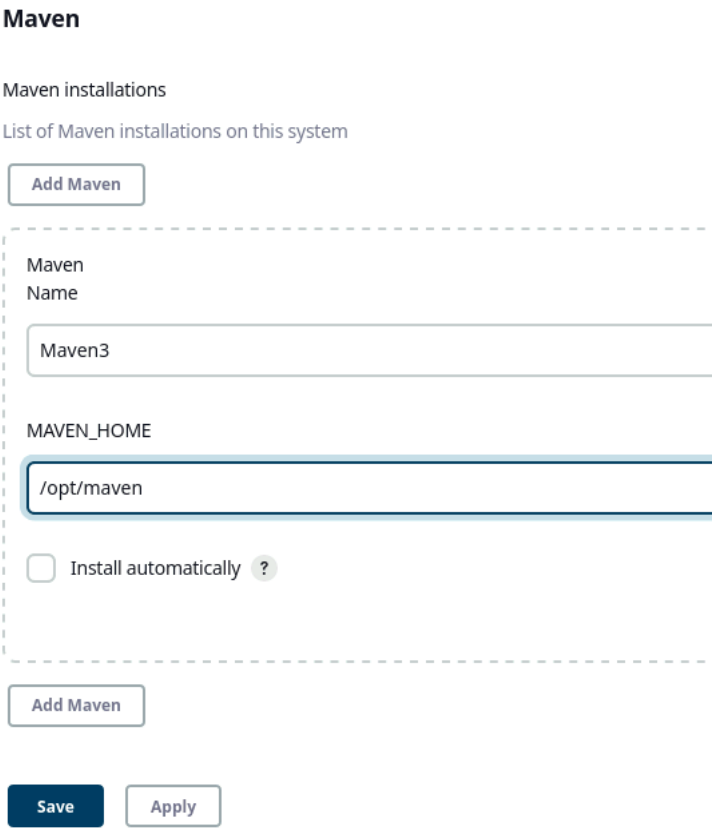
Jenkins: Jenkins is an open-source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat.

Deploy Application war file to Tomcat using Jenkins:

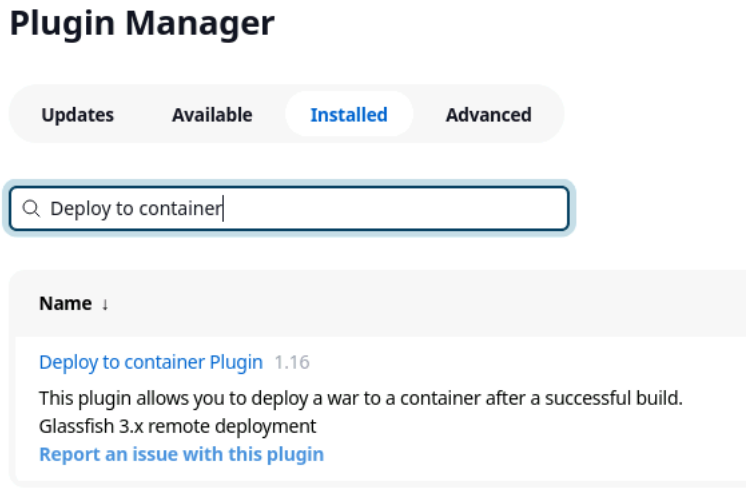
1. Configure Jenkins and install suggested plugins
2. Go to Mange Jenkins → Manage Plugins → Available, search for Deploy Container plugin and select and install without restart

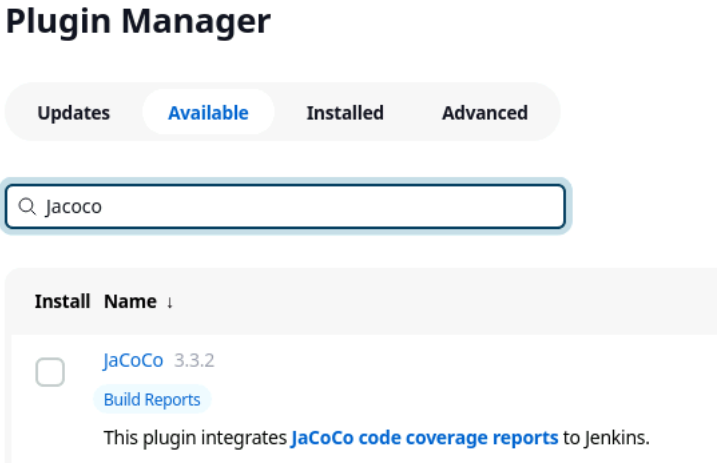


3. Configure maven installer.
    a. Go to Jenkins → Manage Jenkins → Global Tool Configuration.
    b. Under Maven installation provide the name and path of the maven installation directory as show in image.
    c. To check the maven installation director user *mvn –version* command

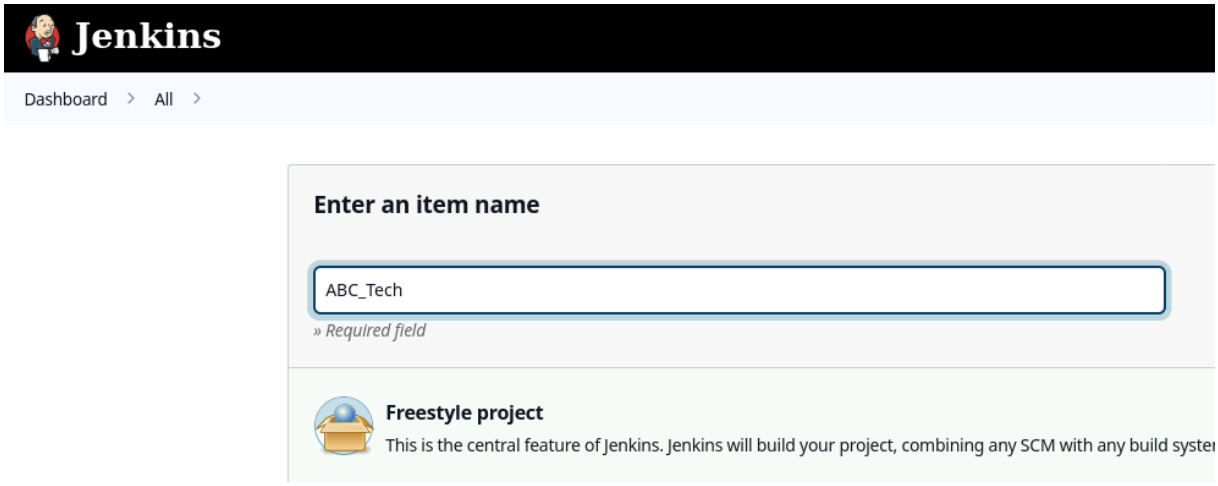


4. Also install deploy to container and Jacoco plugins under Jenkins → Manage Jenkins → Mange Plug-ins



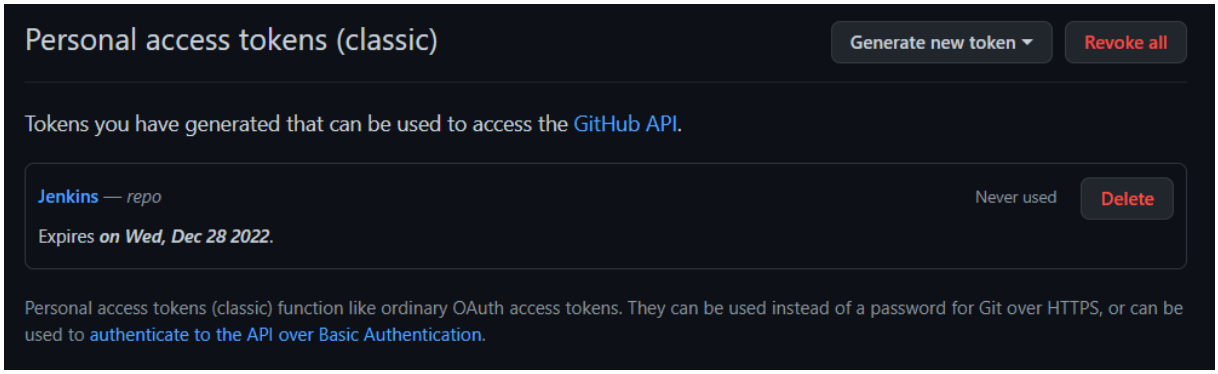


5. Create new item Job as free style project New Item → Freestyle project
   Enter name as "ABC_Tech" or any other suitable name and hit OK button.

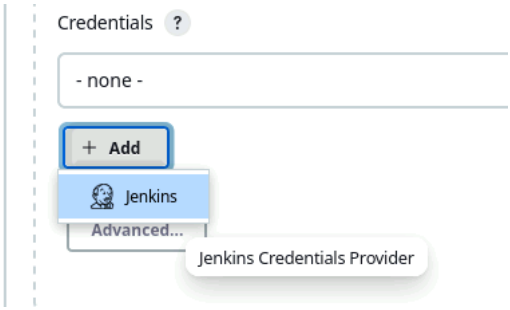


6. Under source code management add GitHub repository

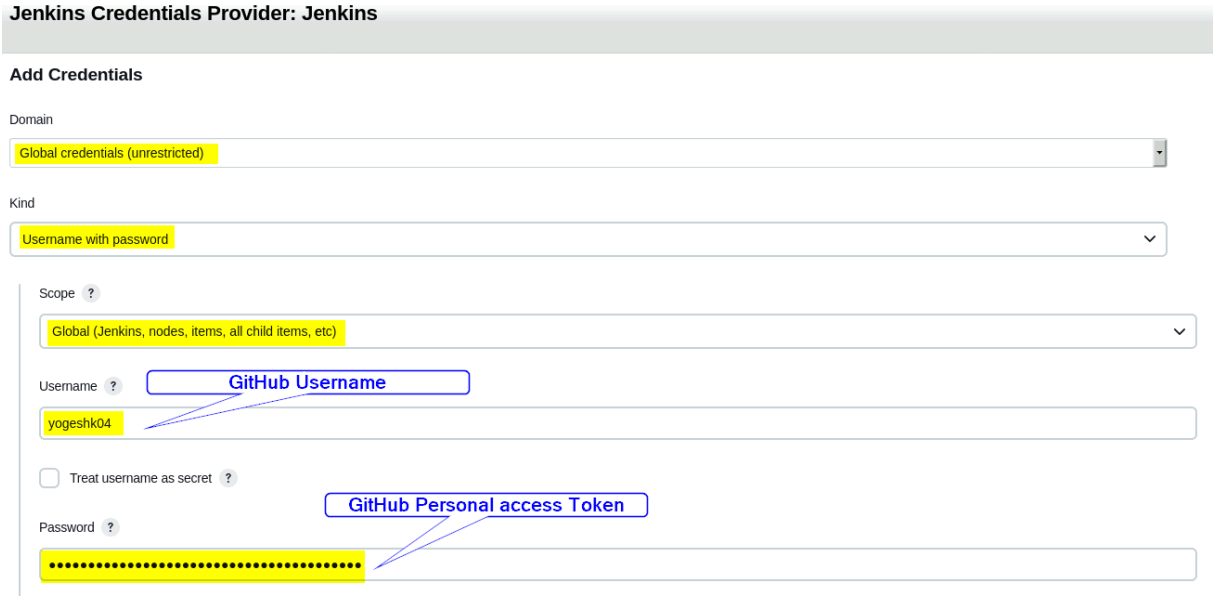
7. For accessing GitHub repo create new Personal Access Token and add copy the token. Follow the link to create the token here



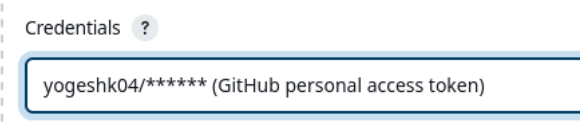
8. Click on + Add → Jenkins – Jenkins Credentials Provider



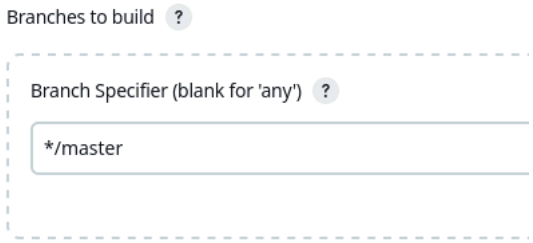
9. Add Credentials as shown in following image and click Add



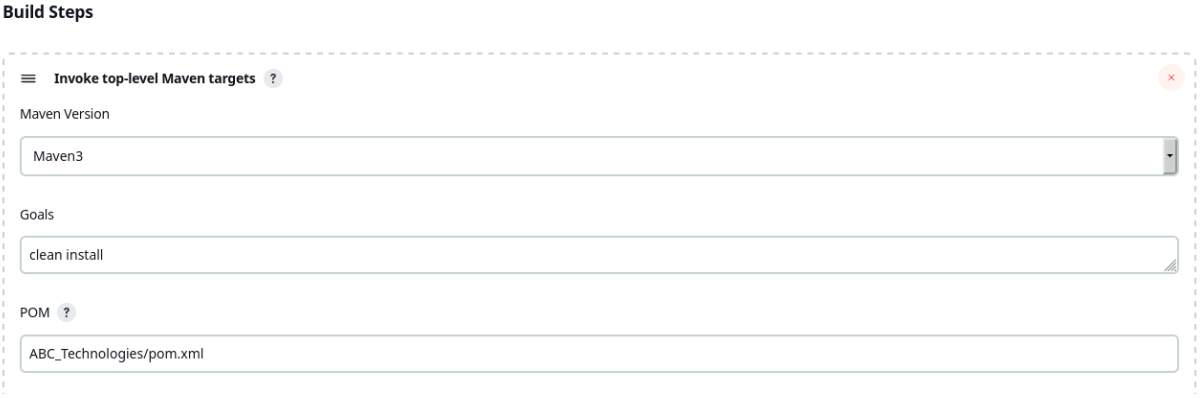
10. Now select the credentials form dop-down



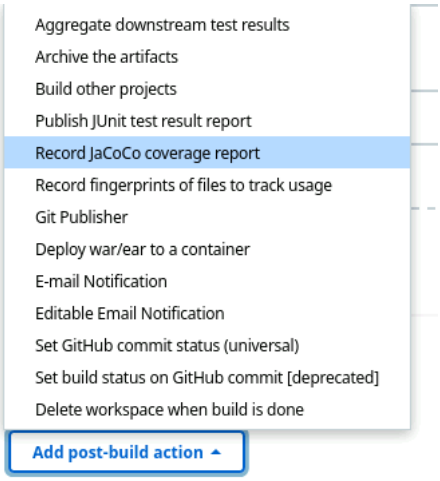
11. Select the respective branch to deploy



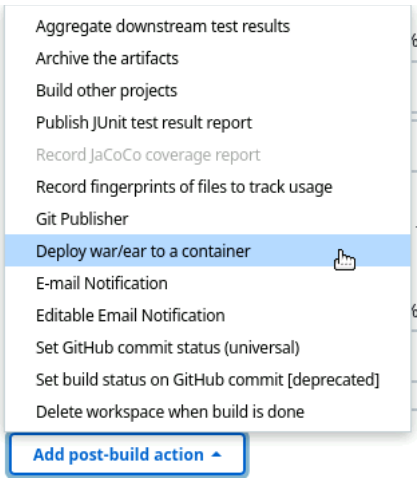
12. As we already configure maven installer in step 3, provide the build steps here and pom.xml file name



13. For the post build action setup, the JaCoCo code coverage into post build actions. Just enable the Record JaCoCo coverage report option

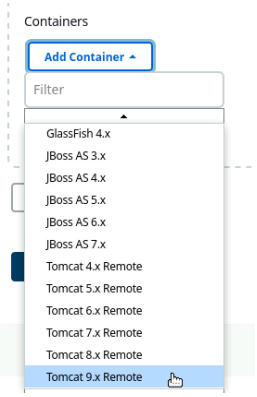


14. Select Add Post-build action → Deploy war/ear to a container



15. Add WAR files

16. Now, select Containers → Add Container → Tomcat 9.x Remote option as we have installed tomcat 9.

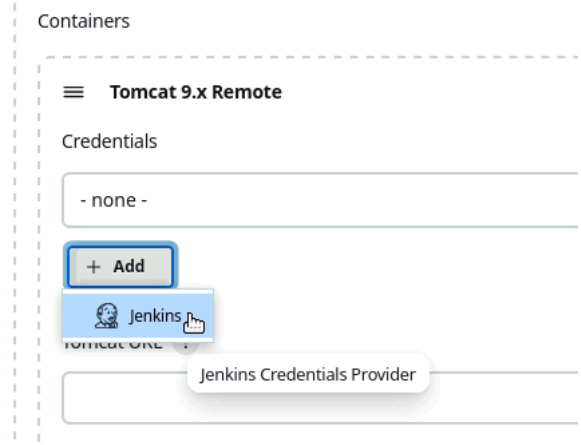


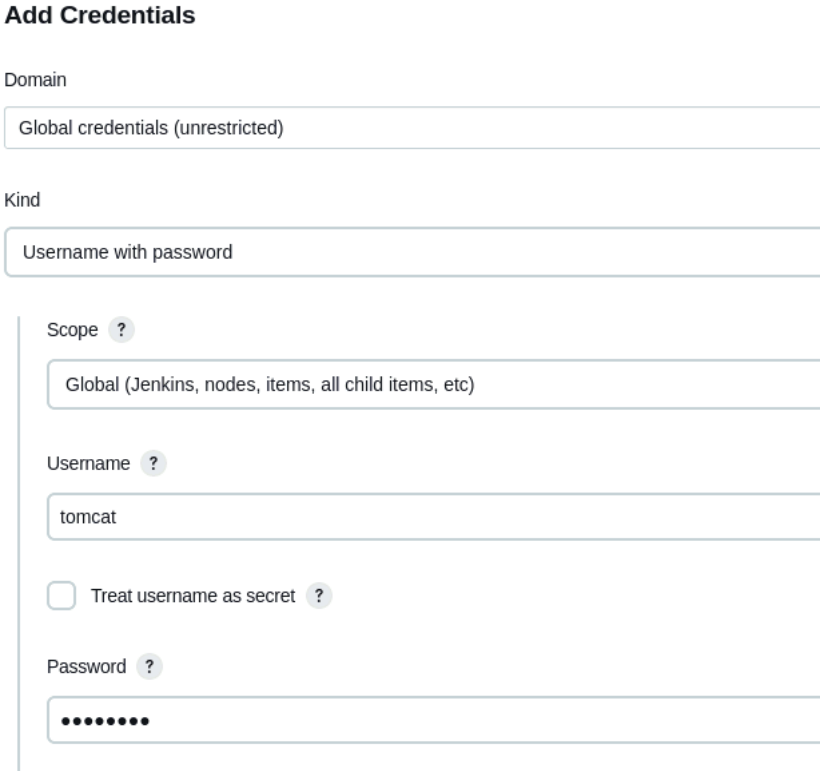
17. Add “*tomcat*” as a user into Tomcat conf file “*tomcat-users.xml*”

```
<user username="tomcat" password=" password " roles="manager-script"/>
```

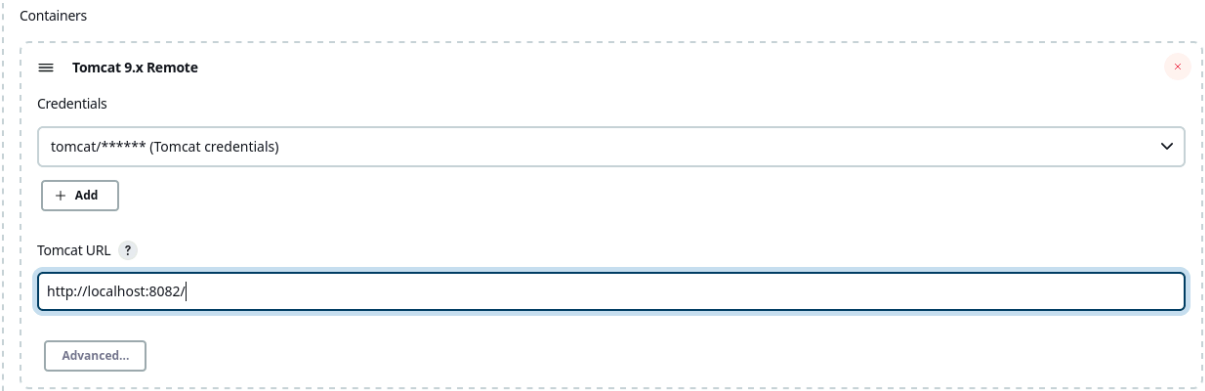
```
<role rolename="manager-script" />
<user username="tomcat" password="password" roles="manager-script" />
```

18. Add Tomcat credentials

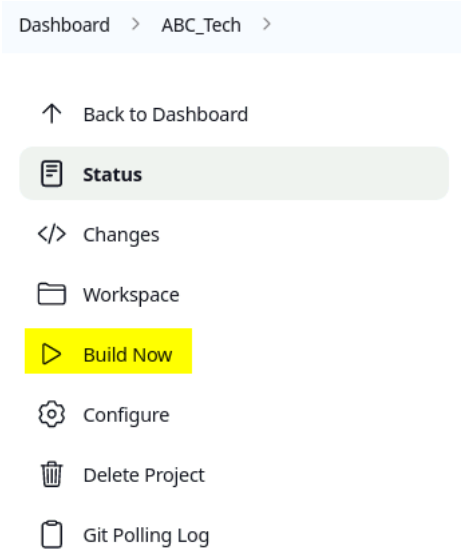




19. Select the Tomcat credentials from drop-down menu and Tomcat URL

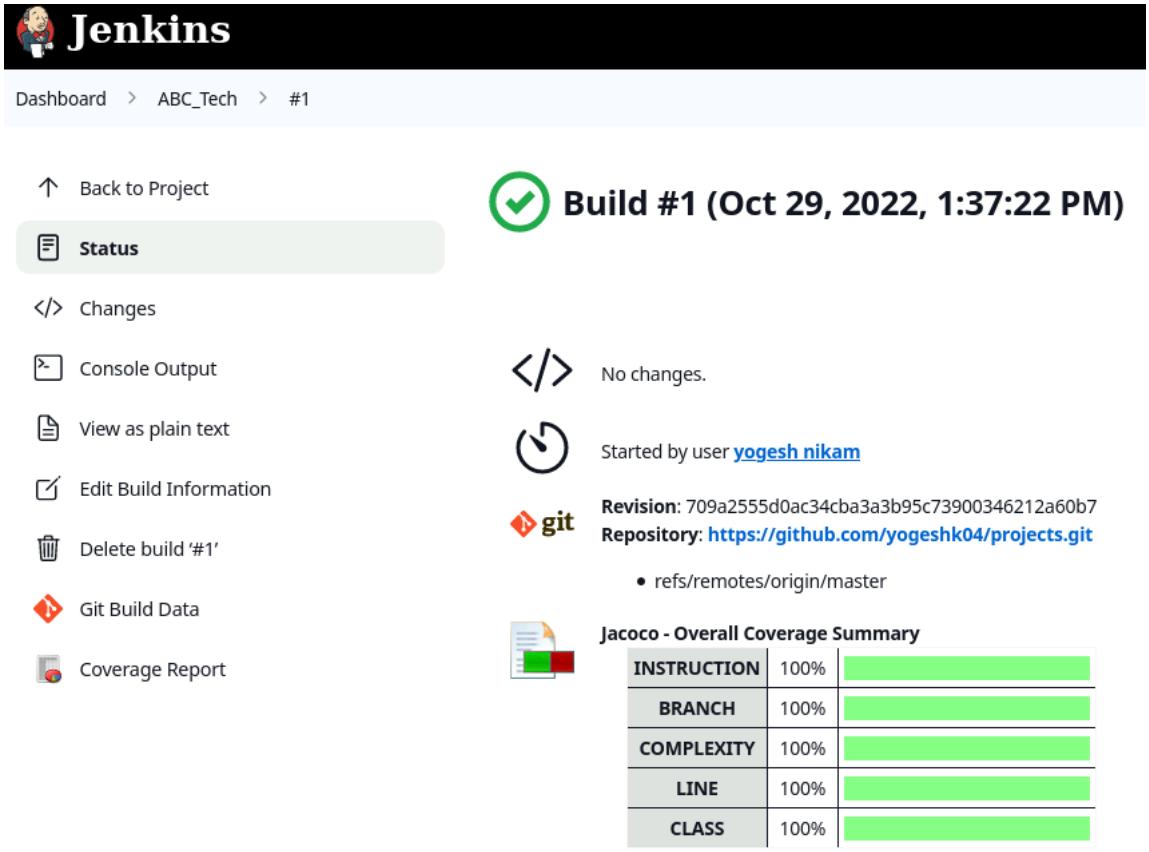


20. Apply and save the configuration job
21. Go the Job and click Build Now

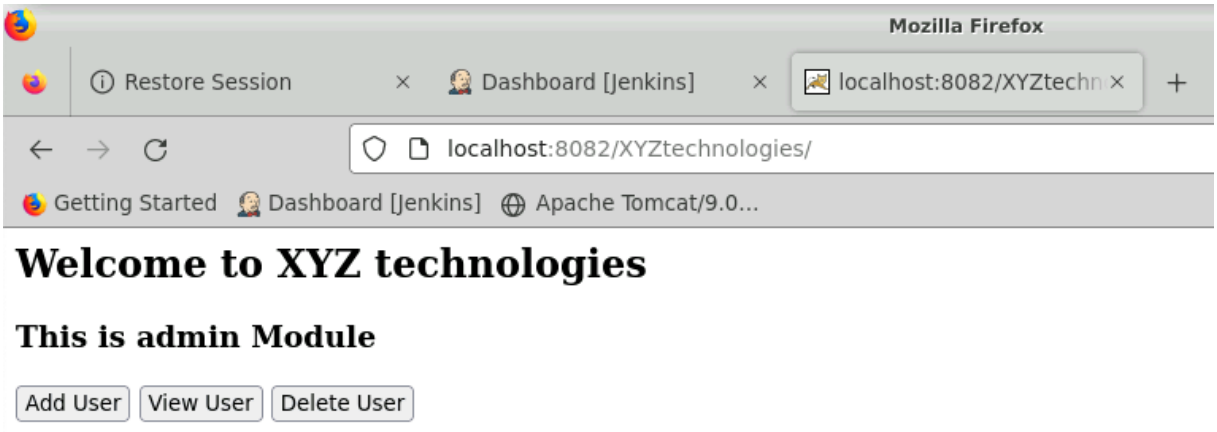


22. If the build is successful, you will see the following build success message

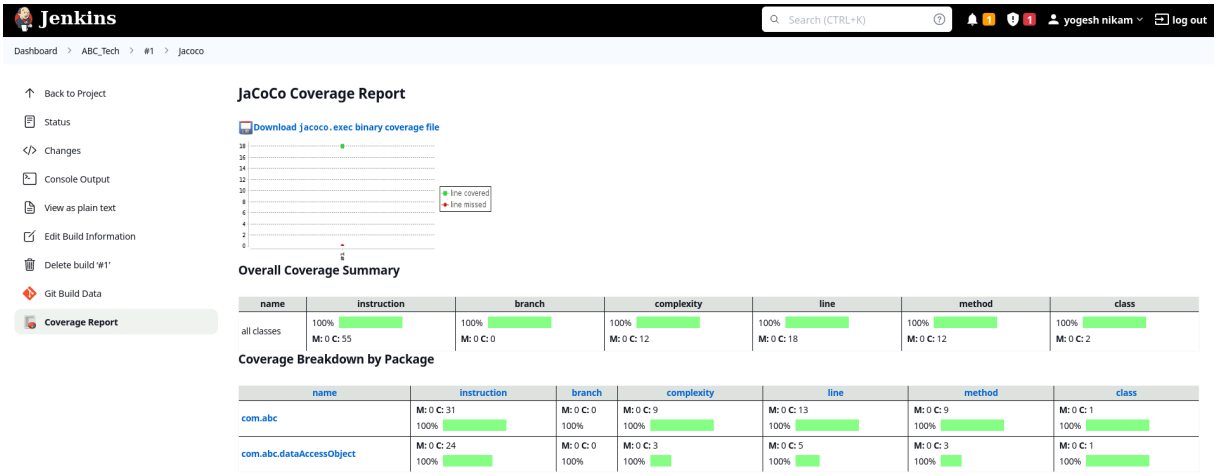
```
[INFO] Installing /var/lib/jenkins/workspace/ABC_Tech/ABC_Technologies/pom.xml to /var/lib/jenkins/.m2/repository/com/abc/ABCtechnologies/1.0/ABCtechnologies-1.0.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  02:12 min
[INFO] Finished at: 2022-10-29T13:39:43Z
[INFO] ------------------------------------------------------------------------
[JaCoCo plugin] Collecting JaCoCo coverage data...
[JaCoCo plugin] **/**.exec;**/classes;**/src/main/java; locations are configured
[JaCoCo plugin] Number of found exec files for pattern **/**.exec: 1
[JaCoCo plugin] Saving matched execfiles:  /var/lib/jenkins/workspace/ABC_Tech/ABC_Technologies/target/jacoco.exec
[JaCoCo plugin] Saving matched class directories for class-pattern: **/classes:
[JaCoCo plugin]  - /var/lib/jenkins/workspace/ABC_Tech/ABC_Technologies/target/ABCtechnologies-1.0/WEB-INF/classes 3 files
[JaCoCo plugin]  - /var/lib/jenkins/workspace/ABC_Tech/ABC_Technologies/target/classes 3 files
[JaCoCo plugin] Saving matched source directories for source-pattern: **/src/main/java:
[JaCoCo plugin] Source Inclusions: **/*.java,**/*.groovy,**/*.kt,**/*.kts
[JaCoCo plugin] Source Exclusions:
[JaCoCo plugin] - /var/lib/jenkins/workspace/ABC_Tech/ABC_Technologies/src/main/java 3 files
[JaCoCo plugin] Loading inclusions files..
[JaCoCo plugin] inclusions: []
[JaCoCo plugin] exclusions: []
[JaCoCo plugin] Thresholds: JacocoHealthReportThresholds [minClass=0, maxClass=0, minMethod=0, maxMethod=0, minLine=0, maxLine=0, minBranch=0, maxBranch=0, minInstruction=0,
maxInstruction=0, minComplexity=0, maxComplexity=0]
[JaCoCo plugin] Publishing the results..
[JaCoCo plugin] Loading packages..
[JaCoCo plugin] Done.
[JaCoCo plugin] Overall coverage: class: 100.0, method: 100.0, line: 100.0, branch: 100.0, instruction: 100.0, complexity: 100.0
[DeployPublisher][INFO] Attempting to deploy 1 war file(s)
[DeployPublisher][INFO] Deploying /var/lib/jenkins/workspace/ABC_Tech/ABC_Technologies/target/ABCtechnologies-1.0.war to container Tomcat 9.x Remote with context null
  Redeploying [/var/lib/jenkins/workspace/ABC_Tech/ABC_Technologies/target/ABCtechnologies-1.0.war]
  Undeploying [/var/lib/jenkins/workspace/ABC_Tech/ABC_Technologies/target/ABCtechnologies-1.0.war]
  Deploying [/var/lib/jenkins/workspace/ABC_Tech/ABC_Technologies/target/ABCtechnologies-1.0.war]
Finished: SUCCESS
```



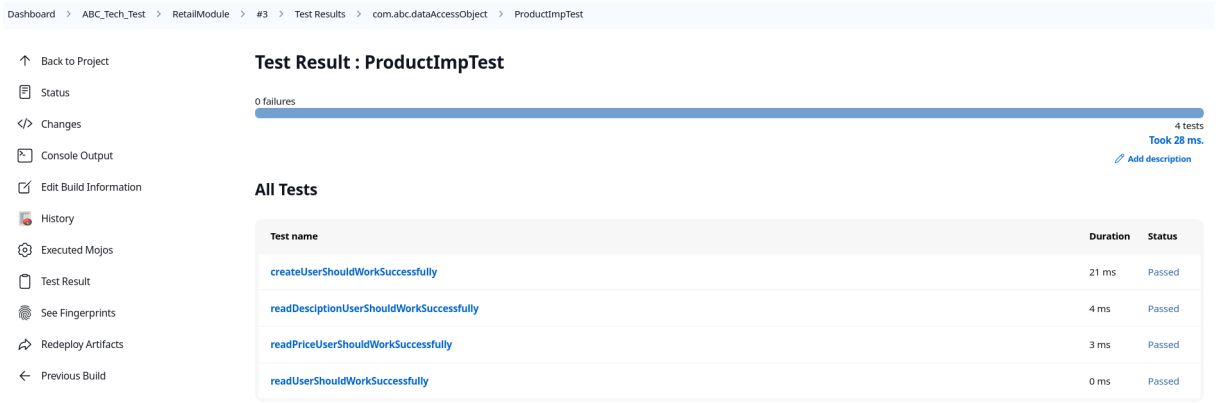
23. Access the web application

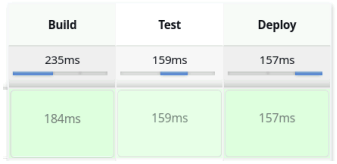


24. JaCoCo code coverage report



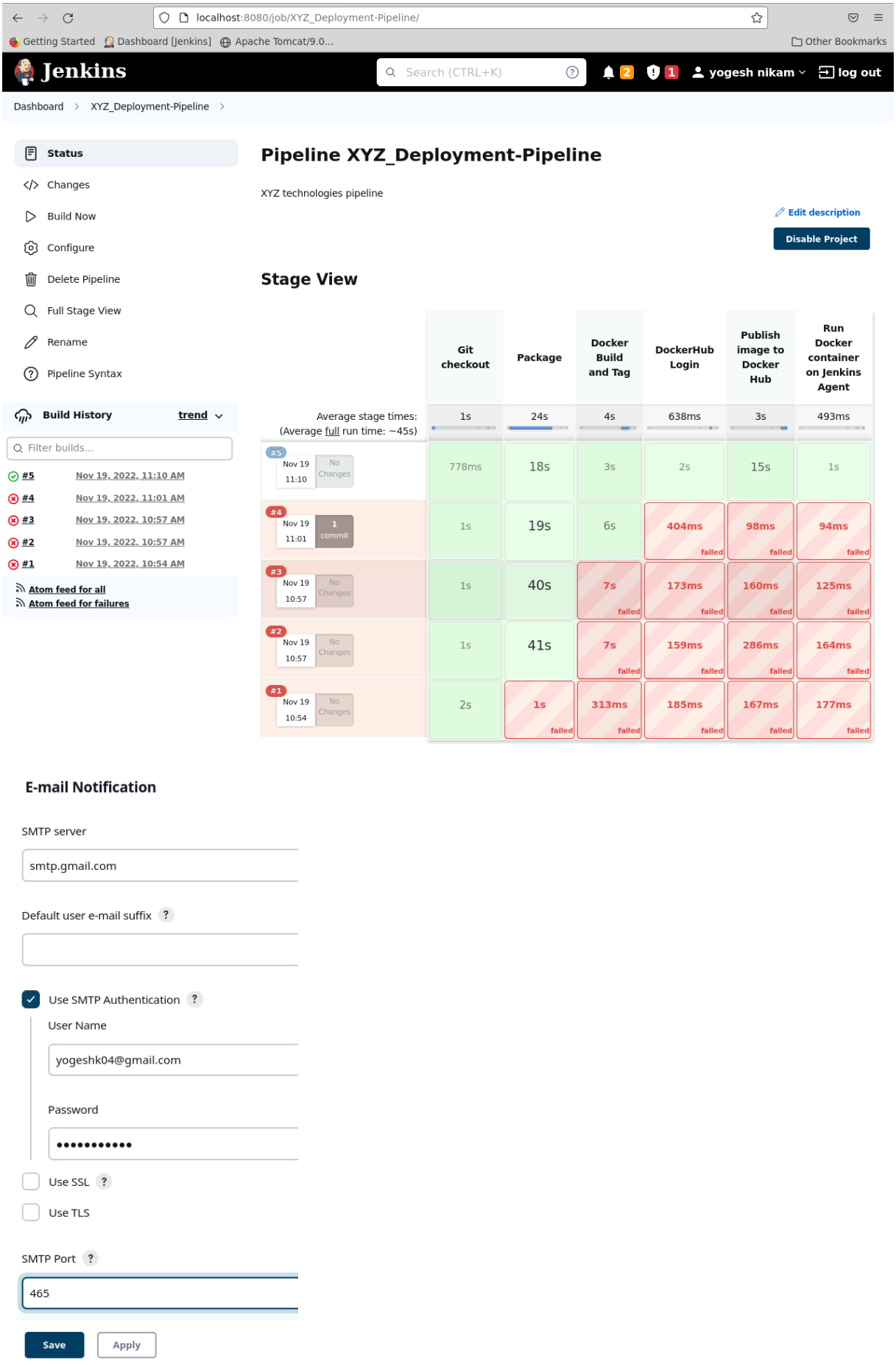
Test result:





Create Pipeline for the project

1. From Dashboard click on New Item
2. Enter an item name for the pipeline and click OK

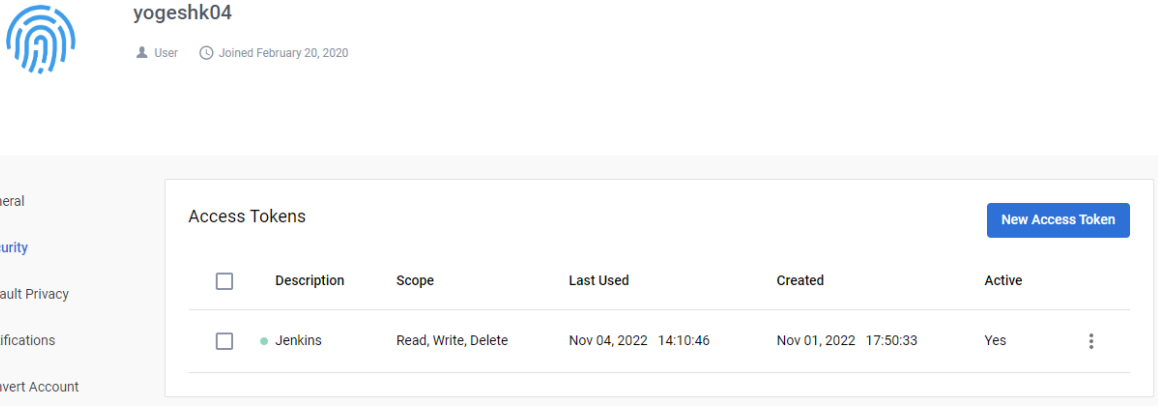

localhost:8080/job/XYZ_Deployment-Pipeline/
Getting Started
Dashboard [Jenkins]
Apache Tomcat/9.0...
Other Bookmarks
Jenkins
Search (CTRL+K)
yogesh nikam
log out
Dashboard
XYZ_Deployment-Pipeline
Status
Changes
Build Now
Configure
Delete Pipeline
Full Stage View
Rename
Pipeline Syntax
Build History
trend
Filter builds...
#5 Nov 19, 2022, 11:10 AM
#4 Nov 19, 2022, 11:01 AM
#3 Nov 19, 2022, 10:57 AM
#2 Nov 19, 2022, 10:57 AM
#1 Nov 19, 2022, 10:54 AM
Atom feed for all
Atom feed for failures
Pipeline XYZ_Deployment-Pipeline
XYZ technologies pipeline
Edit description
Disable Project
Stage View
Git checkout
Package
Docker Build and Tag
DockerHub Login
Publish image to Docker Hub
Run Docker container on Jenkins Agent
Average stage times:
(Average full run time: ~45s)
1s
24s
4s
638ms
3s
493ms
#5 Nov 19 11:10 No Changes
778ms
18s
3s
2s
15s
1s
#4 Nov 19 11:01 1 commit
1s
19s
6s
404ms failed
98ms failed
94ms failed
#3 Nov 19 10:57 No Changes
1s
40s
7s failed
173ms failed
160ms failed
125ms failed
#2 Nov 19 10:57 No Changes
1s
41s
7s failed
159ms failed
286ms failed
164ms failed
#1 Nov 19 10:54 No Changes
2s
1s failed
313ms failed
185ms failed
167ms failed
177ms failed

E-mail Notification
SMTP server
smtp.gmail.com
Default user e-mail suffix
Use SMTP Authentication
User Name
yogeshk04@gmail.com
Password
••••••••••••
Use SSL
Use TLS
SMTP Port
465
Save
Apply

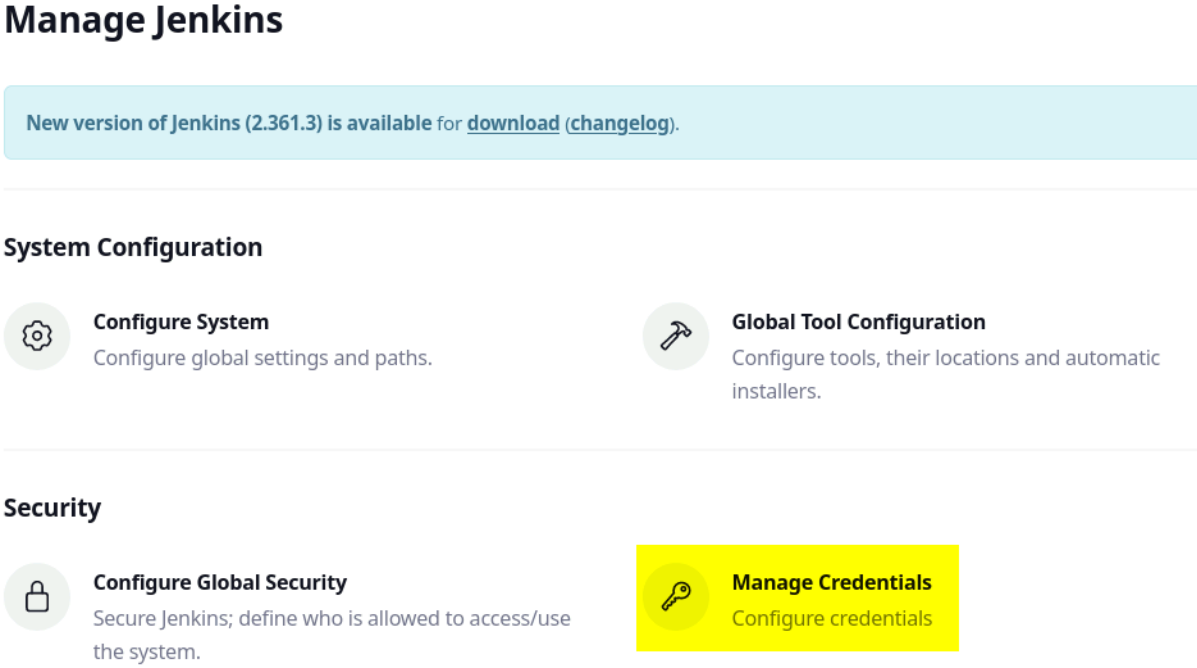
Docker: Docker is a set of platforms as a service product that use OS-level virtualization to deliver software in packages called containers. The service has both free and premium tiers. The software that hosts the containers is called Docker Engine.

Task 3: Write a Docker file. Create an Image and container on the Docker host. Integrate docker host with Jenkins. Create CI/CD job on Jenkins to build and deploy on a container

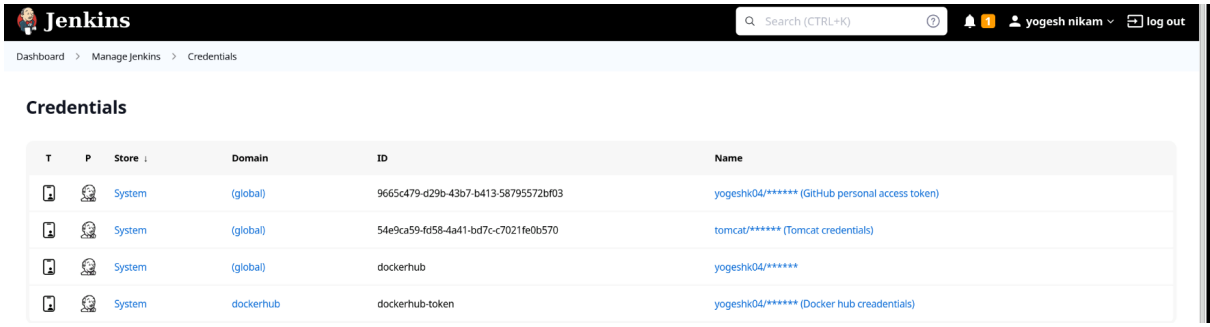
1. Create new item → Pipeline → provide the suitable name
2. Add GitHub credential
3. Add Docker Hub access token into Jenkins
    a. Create new Access token on your DockerHub account under Security section.



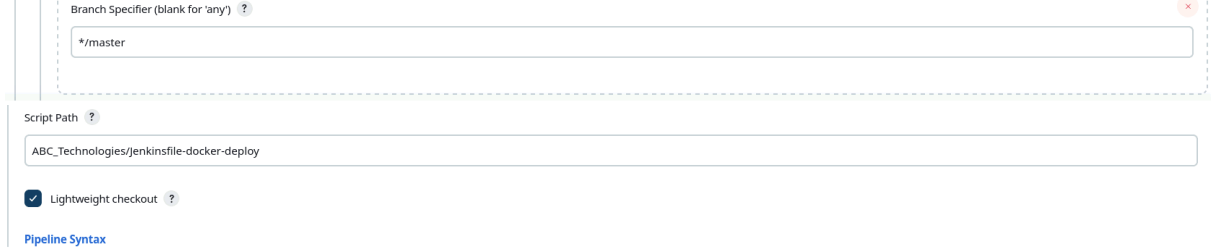
    b. Open Manage Jenkins –> Mange Credentials



    c. Add the System Credentials into Jenkins



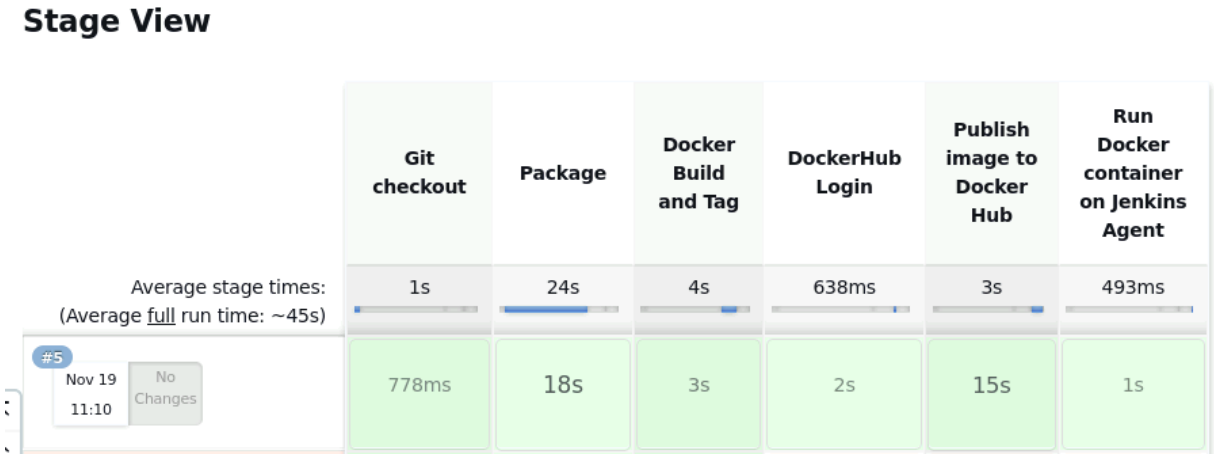
4. Go to Pipeline section add definition pipeline script form SCM
   Select the details as show in the following image.
   Jenkins file can be found at repo. file name is *Jenkinsfile*



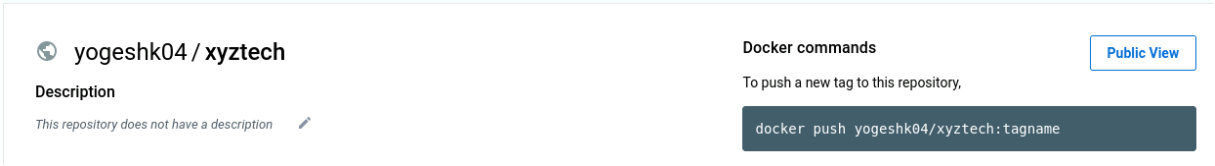
5. Pipeline script

```
pipeline {
    agent any

  environment {
        DOCKERHUB_CREDENTIALS=credentials('dockerhub-token')
    }
  stages {
    stage('Git checkout') {
      steps {
        git credentialsId: 'git_credentials', url: 'https://github.com/yogeshk04/projects.git'
      }
    }
    stage('Package') {
      steps {
        sh '''cd XYZ_Technologies/
            mvn clean package
            mv target/*.war target/xyztech.war
        '''
      }
    }
    stage('Docker Build and Tag') {
      steps {
        sh '''cd XYZ_Technologies/
            sudo docker build -t xyztech:latest .
            sudo docker tag xyztech yogeshk04/xyztech:latest
        '''
      }
    }
    stage('DockerHub Login') {
      steps {
        sh 'echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin'
      }
    }
    stage('Publish image to Docker Hub') {
      steps {
          sh 'sudo docker push yogeshk04/xyztech:latest'
      }
    }

    stage('Run Docker container on Jenkins Agent') {
      steps {
        sh "docker run -d -p 8003:8080 yogeshk04/xyztech"
      }
    }
  }
}
```

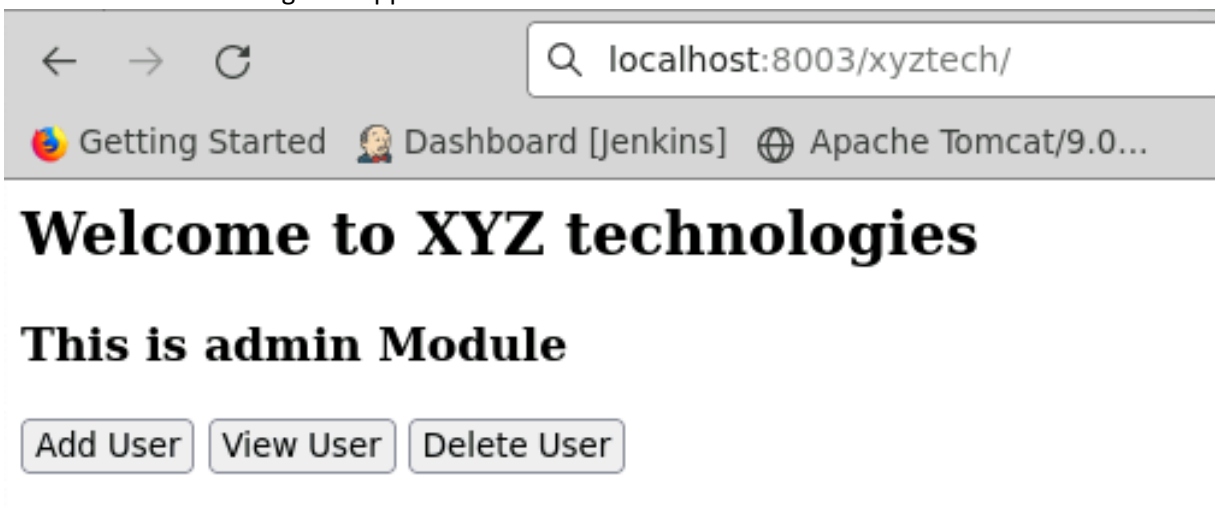
6. Apply and Save
7. Build Now → After completing the pipeline you should see the following pipeline results.



8. Login to Docker Hub account and verify the image is successfully pushed to docker hub account.



9. Access the web application running as a **container** using http://localhost:8003/xyztech/ you should see the following web application



Kubernetes: Kubernetes is an open-source container orchestration system for automating software deployment, scaling, and management. Google originally designed Kubernetes, but the Cloud Native Computing Foundation now maintains the project.

1. User **contexts**: A Kubernetes context is used to group access parameters under an easily recognizable name in a kubeconfig file – a file used to configure access to clusters. It is the connection to a particular cluster used by kubectl. This concept only applies in the place where the kubectl command is run

```
edureka@kmaster:~$ kubectl config get-contexts
CURRENT   NAME                          CLUSTER      AUTHINFO           NAMESPACE
*         kubernetes-admin@kubernetes   kubernetes   kubernetes-admin
edureka@kmaster:~$ kubectl config use-context kubernetes-admin@kubernetes
Switched to context "kubernetes-admin@kubernetes".
edureka@kmaster:~$
```

2. Manifest files are updated into github repo
   https://github.com/yogeshk04/projects/tree/master/XYZ_Technologies/kubernetes
3. Create deployment

```
edureka@kmaster:~/projects/XYZ_Technologies$ cd kubernetes/
edureka@kmaster:~/projects/XYZ_Technologies/kubernetes$ kubectl create -f app-deployment.yaml
deployment.apps/xyztech-deployment created
edureka@kmaster:~/projects/XYZ_Technologies/kubernetes$
```

4. Get newly created pods

```
xyztech-deployment-5cff4ff84c-gfv86                    1/1     Running   0          23s
xyztech-deployment-5cff4ff84c-s9rwk                    1/1     Running   0          23s
xyztech-deployment-5cff4ff84c-ww44c                    1/1     Running   0          23s
edureka@kmaster:~/projects/XYZ_Technologies/kubernetes$
```

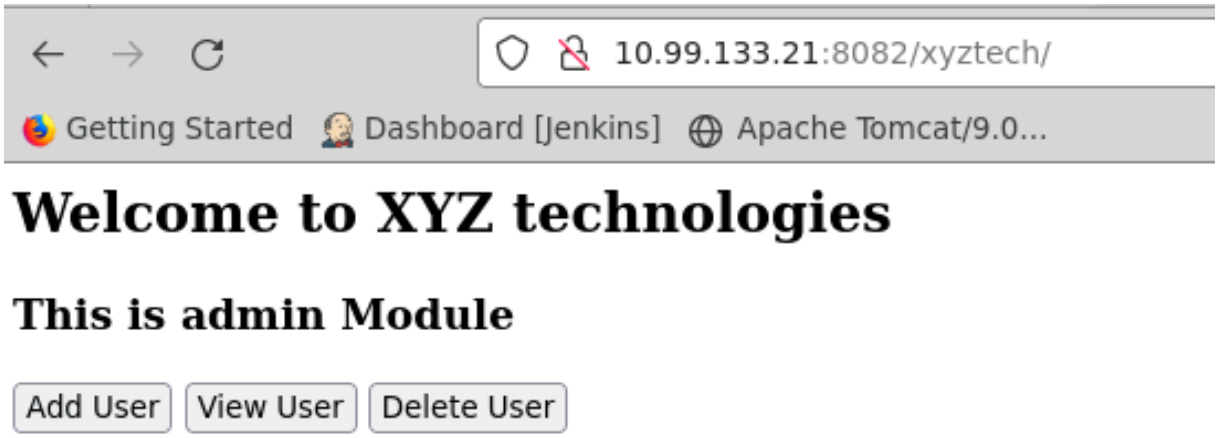
5. Create service

```
edureka@kmaster:~/projects/XYZ_Technologies/kubernetes$ kubectl create -f app-service.yaml
service/xyztech-service created
edureka@kmaster:~/projects/XYZ_Technologies/kubernetes$
```

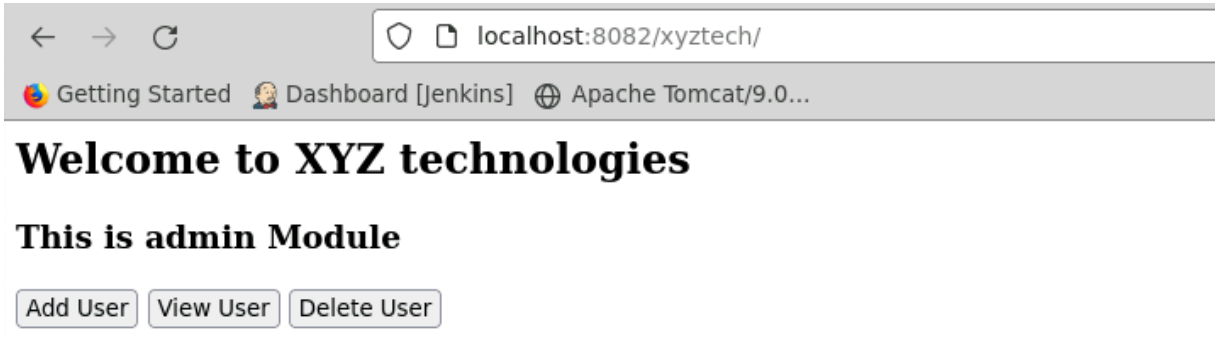
6. Get newly created service

```
xyztech-service                NodePort       10.99.133.21     <none>        8082:31197/TCP
```

7. Access the web app using cluster IP as the external ip assignment is still pending



8.



Helm: Helm helps you manage Kubernetes applications. Helm Charts help you define, install, and upgrade even the most complex Kubernetes application.

Deploy application on Kubernetes using Helm

1. Install helm

```
edureka@kmaster:/tmp$ curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3
edureka@kmaster:/tmp$ chmod 700 get_helm.sh
edureka@kmaster:/tmp$ ./get_helm.sh
Downloading https://get.helm.sh/helm-v3.10.2-linux-amd64.tar.gz
Verifying checksum... Done.
Preparing to install helm into /usr/local/bin
helm installed into /usr/local/bin/helm
edureka@kmaster:/tmp$ helm --version
Error: unknown flag: --version
edureka@kmaster:/tmp$ helm version
version.BuildInfo{Version:"v3.10.2", GitCommit:"50f003e5ee8704ec937a756c646870227d7c8b58", GitTreeState:"clean", GoVersion:"go1.18.8"}
edureka@kmaster:/tmp$
```

2. Update helm

```
edureka@kmaster:/tmp$ helm repo add stable https://charts.helm.sh/stable
"stable" has been added to your repositories
edureka@kmaster:/tmp$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "stable" chart repository
Update Complete. *Happy Helming!*
edureka@kmaster:/tmp$
```

3. Clone the project repository on Kubernetes master and check the nodes

```
Dockerfile  Jenkins  README.md  ansible  helm-xyztech  kubernetes  pom.xml  pom.xml.bak  src  target
edureka@kmaster:~/projects/XYZ_Technologies$
```

4. Deploy application using following helm reference repository directory helm-charts

   Helm deployment command

   : ~$ helm install xyztechapp helm-chart/

```
edureka@kmaster:~/projects/XYZ_Technologies$ helm install xzytechapp helm-xyztech/
NAME: xzytechapp
LAST DEPLOYED: Mon Dec  5 14:01:49 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
edureka@kmaster:~/projects/XYZ_Technologies$
```