

## 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-policy"
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](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

```
edureka@kmaster: ~  
File Edit View Search Terminal Help  
edureka@kmaster:~$ mvn --version  
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)  
Maven home: /opt/maven  
Java version: 1.8.0_201, vendor: Oracle Corporation, runtime: /usr/lib/jvm/java-8-oracle/jre  
Default locale: en, platform encoding: UTF-8  
OS name: "linux", version: "4.15.0-1021-aws", arch: "amd64", family: "unix"
```

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/com/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
2. Check java version if already install or install the required java version

: ~\$ sudo apt update

: ~\$ 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/apache-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... connected.
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 [11597709/11597709]
```

#### 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

: ~\$ sudo mkdir /opt/tomcat

#### d. Move all the content to that folder

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

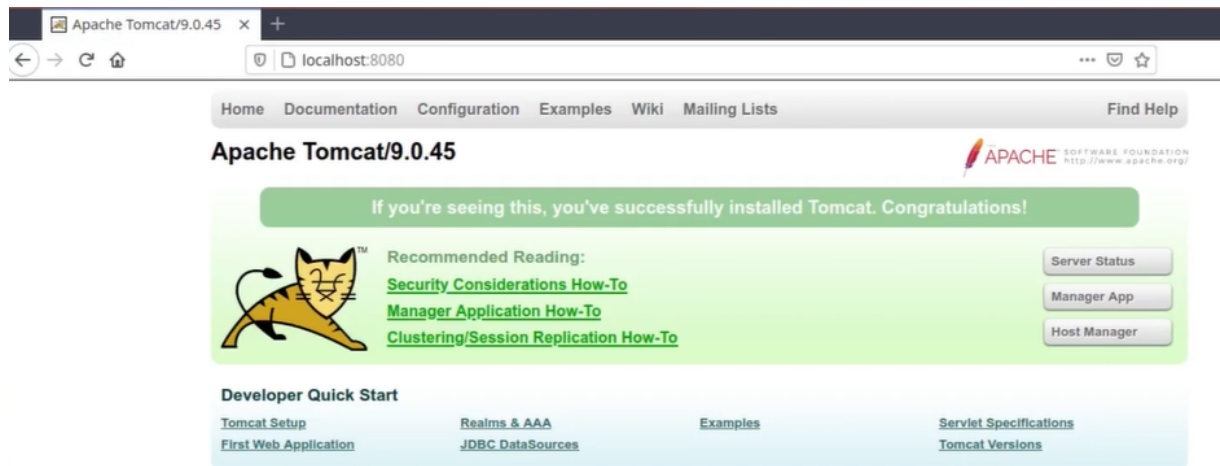
#### e. Change directory to /opt/tomcat/bin

#### 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.
```

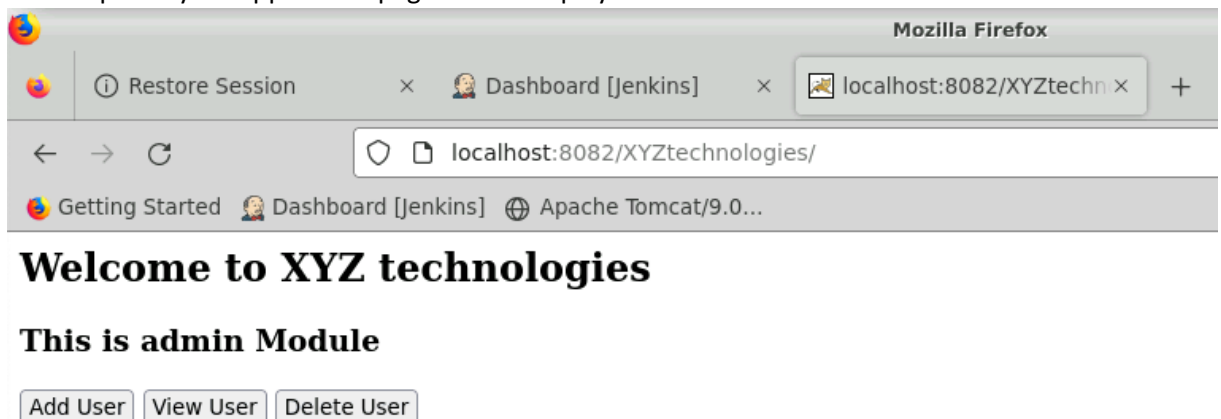
### 5. Access tomcat using http:localhost:8080



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 Manage Jenkins → Manage Plugins → Available, search for Deploy Container plugin and select and install without restart

The screenshot shows the Jenkins web interface. At the top is the Jenkins logo and a navigation breadcrumb: Dashboard > Manage Jenkins > Plugin Manager. On the left sidebar, there are links: 'Back to Dashboard' (with an up arrow icon), 'Manage Jenkins' (with a gear icon), and 'Update Center' (with a download icon). The main area is titled 'Plugin Manager' and has four tabs: 'Updates', 'Available' (which is selected and highlighted in blue), 'Installed', and 'Advanced'. Below the tabs is a search bar containing the text 'Deploy'. The search results are displayed in a table-like format with two entries:

Install	Name
<input checked="" type="checkbox"/>	<b>Deploy to container</b> 1.16 Artifact Uploaders This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment
<input type="checkbox"/>	<b>Docker Pipeline</b> 528.v7c193a_0b_e67c pipeline DevOps Deployment docker Build and use Docker containers from pipelines.

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

## Maven

Maven installations

List of Maven installations on this system

Add Maven

Maven Name

Maven3

MAVEN\_HOME

/opt/maven

☐ Install automatically ?

Add Maven

Save

Apply

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

## Plugin Manager

Updates

Available

Installed

Advanced

Q Deploy to container

Name ↓

Deploy to container Plugin 1.16

This plugin allows you to deploy a war to a container after a successful build.  
Glassfish 3.x remote deployment  
[Report an issue with this plugin](#)

## Plugin Manager

Updates

Available

Installed

Advanced

Q Jacoco

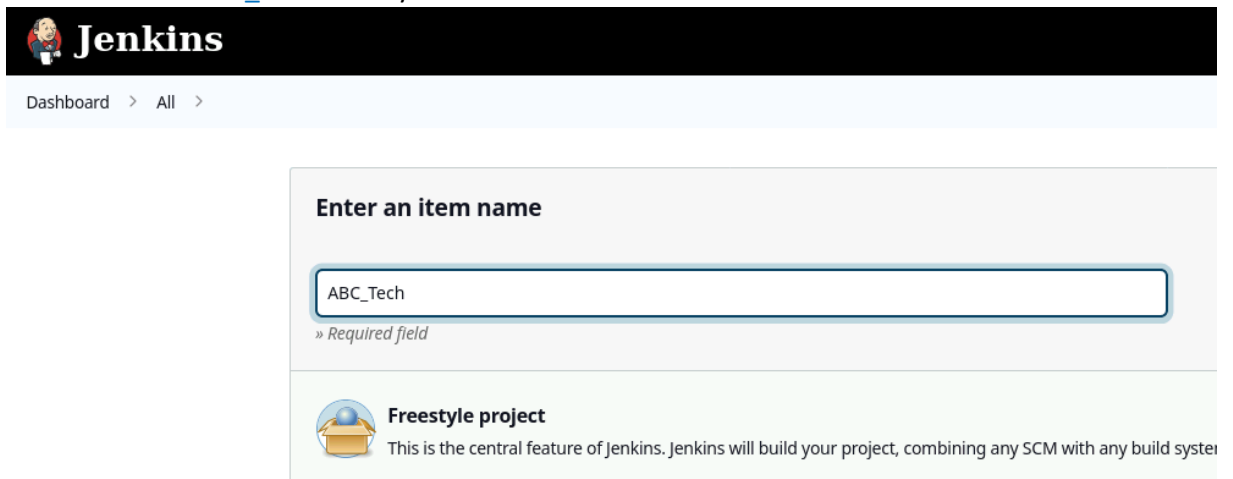
Install Name ↓

☐ JaCoCo 3.3.2

Build Reports

This plugin integrates [JaCoCo code coverage reports](#) to Jenkins.

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.



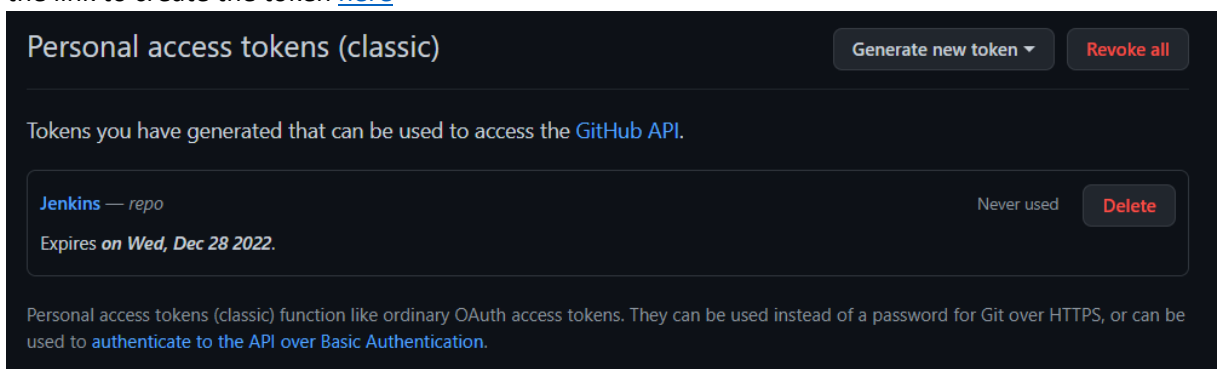
The image shows the Jenkins web interface. At the top, there's a header with the Jenkins logo and the word 'Jenkins'. Below it, a breadcrumb trail shows 'Dashboard > All >'. A modal dialog box titled 'Enter an item name' is open. It contains a text input field with 'ABC\_Tech' entered. Below the input field, it says '» Required field'. Below the dialog box, there's a section for 'Freestyle project' with a description: 'This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system'.

6. Under source code management add GitHub repository  
**Source Code Management**



The image shows the 'Source Code Management' configuration page in Jenkins. It has two radio buttons: 'None' and 'Git'. The 'Git' option is selected. Below it, there's a section for 'Repositories'. It contains a 'Repository URL' input field with the value 'https://github.com/yogeshk04/IndustryGradeProject-1.git'. Below that is a 'Credentials' dropdown menu with '- none -' selected. There are buttons for '+ Add' and 'Advanced...'. A red 'x' icon is visible in the top right corner of the repository configuration area.

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



The image shows the 'Personal access tokens (classic)' page in GitHub. It has a title 'Personal access tokens (classic)' and two buttons: 'Generate new token' and 'Revoke all'. Below the title, it says 'Tokens you have generated that can be used to access the GitHub API.' There's a table with one row showing a token named 'Jenkins — repo'. The token is marked as 'Never used' and has a 'Delete' button. The token expires on 'Wed, Dec 28 2022'. At the bottom, there's a note: 'Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.'

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



Credentials ?

- none -

+ Add

Jenkins

Advanced...

Jenkins Credentials Provider

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

#### Jenkins Credentials Provider: Jenkins

##### Add Credentials

Domain

Global credentials (unrestricted)

Kind

Username with password

Scope ?

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

Username ?

GitHub Username

yogeshk04

☐ Treat username as secret ?

Password ?

GitHub Personal access Token

10. Now select the credentials form dop-down

Credentials ?

yogeshk04/\*\*\*\*\* (GitHub personal access token)

11. Select the respective branch to deploy

Branches to build ?

Branch Specifier (blank for 'any') ?

\*/master

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

##### Build Steps

≡ Invoke top-level Maven targets ?

Maven Version

Maven3

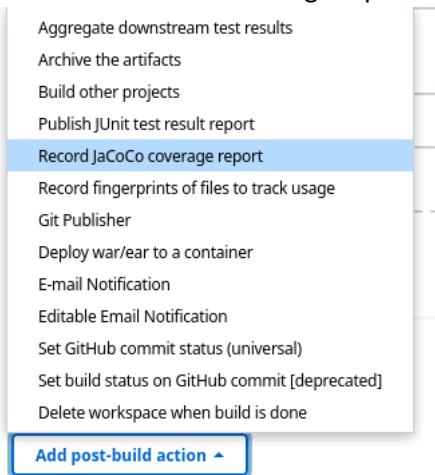
Goals

clean install

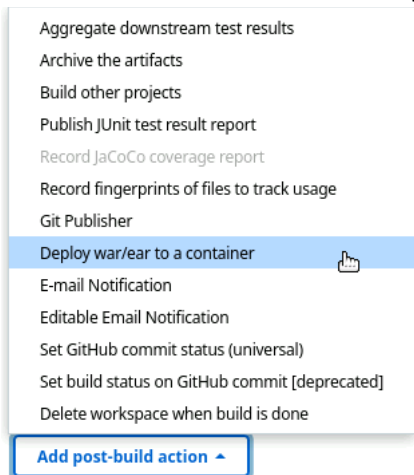
POM ?

ABC\_Technologies/pom.xml

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



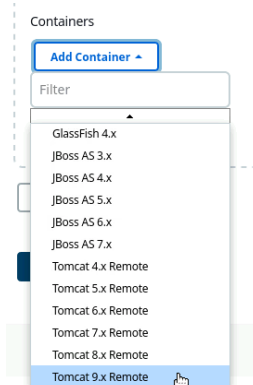
15. Add WAR files

**Deploy war/ear to a container**

WAR/EAR files ?

Context path ?

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



Containers

☰ **Tomcat 9.x Remote**

Credentials

- none -

+ Add

 Jenkins 

Tomcat URL

Jenkins Credentials Provider

### Add Credentials

Domain

Global credentials (unrestricted)

Kind

Username with password

Scope ?

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

Username ?

tomcat


☐ Treat username as secret ?

Password ?

••••••••

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

Containers

☰ **Tomcat 9.x Remote** 

Credentials

tomcat/\*\*\*\*\* (Tomcat credentials) ▼

+ Add

Tomcat URL ?

http://localhost:8082/

Advanced...

20. Apply and save the configuration job

21. Go the Job and click **Build Now**

Dashboard > ABC\_Tech >

↑ Back to Dashboard

 Status

</> Changes

 Workspace

 **Build Now**


 Configure

 Delete Project

 Git Polling Log

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
```



# Jenkins

Dashboard > ABC\_Tech > #1

[↑ Back to Project](#)

Status

[</> Changes](#)

[📄 Console Output](#)


[📄 View as plain text](#)

[📄 Edit Build Information](#)


[🗑️ Delete build '#1'](#)


[📄 Git Build Data](#)


[📄 Coverage Report](#)




## Build #1 (Oct 29, 2022, 1:37:22 PM)


No changes.


Started by user [yogesh nikam](#)


Revision: 709a2555d0ac34cba3a3b95c73900346212a60b7  
Repository: <https://github.com/yogeshk04/projects.git>


- refs/remotes/origin/master



### Jacoco - Overall Coverage Summary

INSTRUCTION	100%	<div></div>
BRANCH	100%	<div></div>
COMPLEXITY	100%	<div></div>
LINE	100%	<div></div>
CLASS	100%	<div></div>

23. Access the web application



Restore Session

Dashboard [Jenkins]

localhost:8082/XYZtechn

localhost:8082/XYZtechnologies/

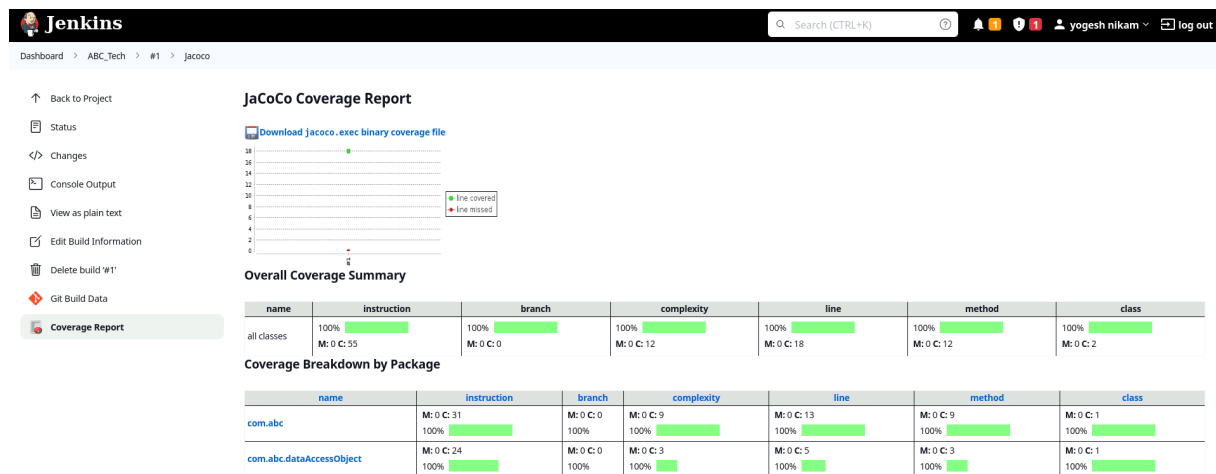
Getting Started Dashboard [Jenkins] Apache Tomcat/9.0...

# Welcome to XYZ technologies

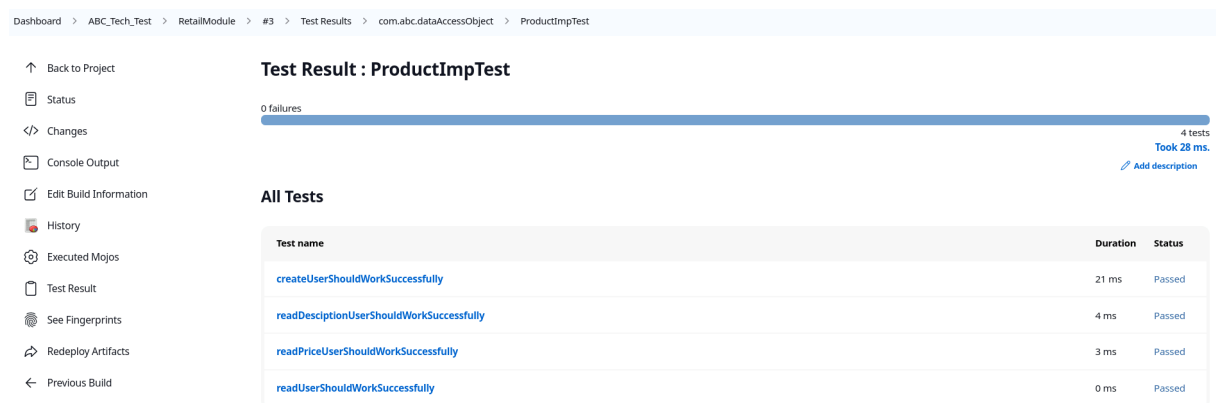
## This is admin Module

Add User View User Delete User

24. JaCoCo code coverage report



## Test result:



Build	Test	Deploy
235ms	159ms	157ms
184ms	159ms	157ms

## Create Pipeline for the project

- From Dashboard click on **New Item**
- Enter an item name for the pipeline and click **OK**

Status

</> Changes

▶ Build Now

⚙️ Configure

🗑️ Delete Pipeline

🔍 Full Stage View

✎ Rename

🔗 Pipeline Syntax

📈 Build History

trend

✔️ #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

Average stage times:  
(Average full run time: ~45s)

	Git checkout	Package	Docker Build and Tag	DockerHub Login	Publish image to Docker Hub	Run Docker container on Jenkins Agent
	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

Default user e-mail suffix ?

☒ Use SMTP Authentication ?

User Name

Password

☐ Use SSL ?

☐ Use TLS

SMTP Port ?

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.



yogeshk04

User Joined February 20, 2020

General

Security

Default Privacy

Notifications

Convert Account

Access Tokens

New Access Token

<input type="checkbox"/>	Description	Scope	Last Used	Created	Active
<input type="checkbox"/>	Jenkins	Read, Write, Delete	Nov 04, 2022 14:10:46	Nov 01, 2022 17:50:33	Yes

- b. Open Manage Jenkins → Manage Credentials

## Manage Jenkins

New version of Jenkins (2.361.3) is available for [download](#) ([changelog](#)).

### System Configuration



#### Configure System

Configure global settings and paths.



#### Global Tool Configuration

Configure tools, their locations and automatic installers.

### Security



#### Configure Global Security

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




#### Manage Credentials

Configure credentials


- c. Add the System Credentials into Jenkins







Jenkins


Search (CTRL+K)





1

yogesh nikam

log out

Dashboard









>

Manage Jenkins

>

Credentials

Credentials

T	P	Store	Domain	ID	Name
		System	(global)	9665c479-d29b-43b7-b413-58795572bf03	yogeshk04/***** (GitHub personal access token)
		System	(global)	54e9ca59-fd58-4a41-bd7c-c7021fe0b570	tomcat/***** (Tomcat credentials)
		System	(global)	dockerhub	yogeshk04/*****
		System	dockerhub	dockerhub-token	yogeshk04/***** (Docker hub credentials)

- 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*

#### Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/yogeshk04/projects.git

Credentials ?

yogeshk04/\*\*\*\*\* (GitHub personal access token)

+ Add

Advanced...

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

\*/master

Script Path ?

ABC\_Technologies/Jenkinsfile-docker-deploy

☒ Lightweight checkout ?

[Pipeline Syntax](#)

- Pipeline script

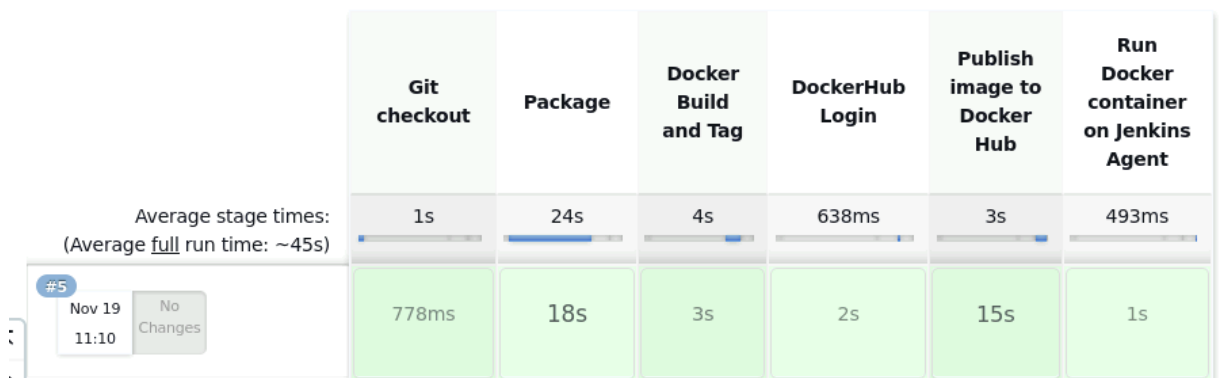
```

1 pipeline {
2     agent any
3
4     environment {
5         DOCKERHUB_CREDENTIALS=credentials('dockerhub-token')
6     }
7     stages {
8         stage('Git checkout') {
9             steps {
10                 git credentialsId: 'git_credentials', url: 'https://github.com/yogeshk04/projects.git'
11             }
12         }
13         stage('Package') {
14             steps {
15                 sh '''cd XYZ_Technologies/
16                     mvn clean package
17                     mv target/*.war target/xyztech.war
18                 '''
19             }
20         }
21         stage('Docker Build and Tag') {
22             steps {
23                 sh '''cd XYZ_Technologies/
24                     sudo docker build -t xyztech:latest .
25                     sudo docker tag xyztech yogeshk04/xyztech:latest
26                 '''
27             }
28         }
29         stage('DockerHub Login') {
30             steps {
31                 sh 'echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin'
32             }
33         }
34         stage('Publish image to Docker Hub') {
35             steps {
36                 sh 'sudo docker push yogeshk04/xyztech:latest'
37             }
38         }
39
40         stage('Run Docker container on Jenkins Agent') {
41             steps {
42                 sh "docker run -d -p 8003:8080 yogeshk04/xyztech"
43             }
44         }
45     }
46 }

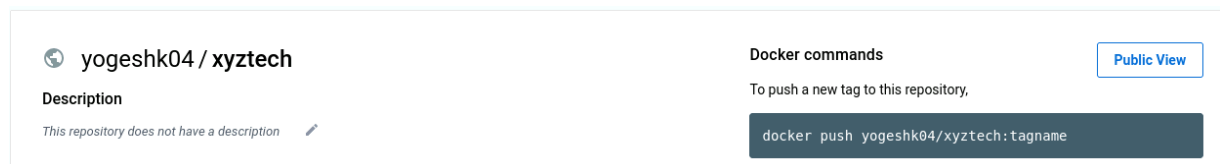
```

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

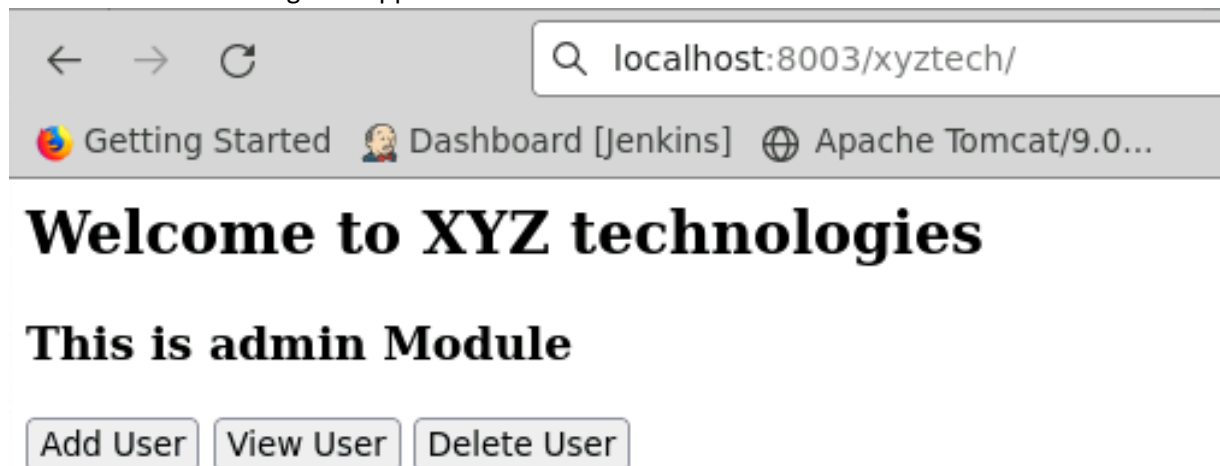
## Stage View



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](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$
```

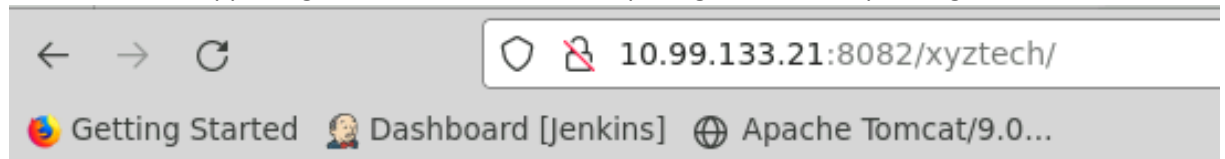
5. Create service

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

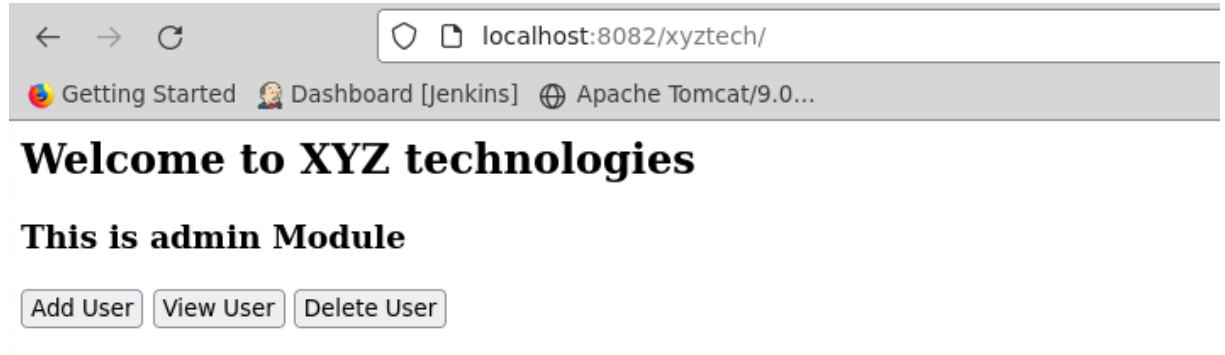
6. Get newly created service

```
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](https://github.com/edureka8/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$ █
```

---