

IndustryGradeProject-1:

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
```

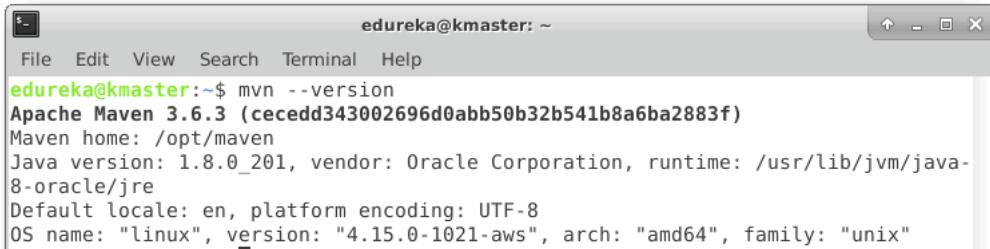


Git: Git is free and open-source software for distributed version control, tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development

1. Create new repository in github and push the initial code to repository
2. <https://github.com/yogeshk04/projects.git>
 - a. Copy the industry grade repo from Edureka industry grade project section
 - b. Go to directory and execute following commands
 - i. : ~\$ git init
 - ii. : ~\$ git add .
 - iii. : ~\$ git commit -m "Initial project code commit"
 - c. Configure git for the first time to github (*Note – you need to have github or gitlab account*)
 - d. 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.*
 - e. Push local repo to GitHub
 - i. : ~\$ git remote add origin https://github.com/yogeshk04/projects.git
 - ii. : ~\$ git push -set-upstream origin master
 - f. Go back to GitHub and see that the repository has been updated.

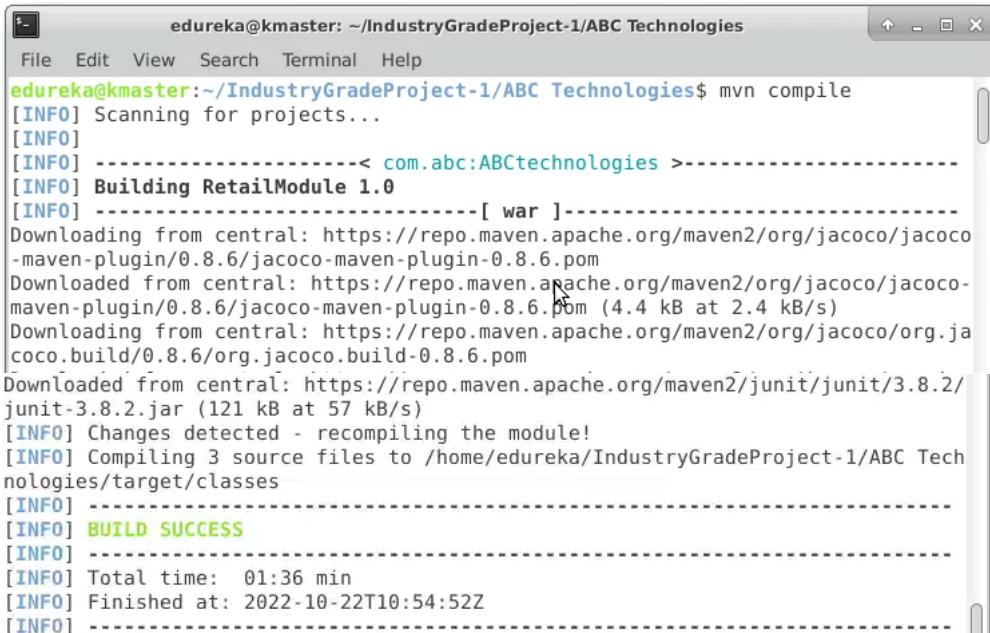
Maven: Maven is a build automation tool used primarily for Java projects. Maven can also be used to build and manage projects written in C#, Ruby, Scala, and other languages. The Maven project is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project.

1. Check maven version and java version using `mvn --version` command



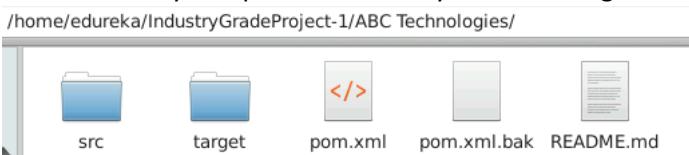
```
edureka@kmaster: ~$ mvn --version
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: /opt/maven
Java version: 1.8.0_201, vendor: Oracle Corporation, runtime: /usr/lib/jvm/java-8-oracle/jre
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "4.15.0-1021-aws",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

2. Compile the project and generate target folder using `mvn compile`

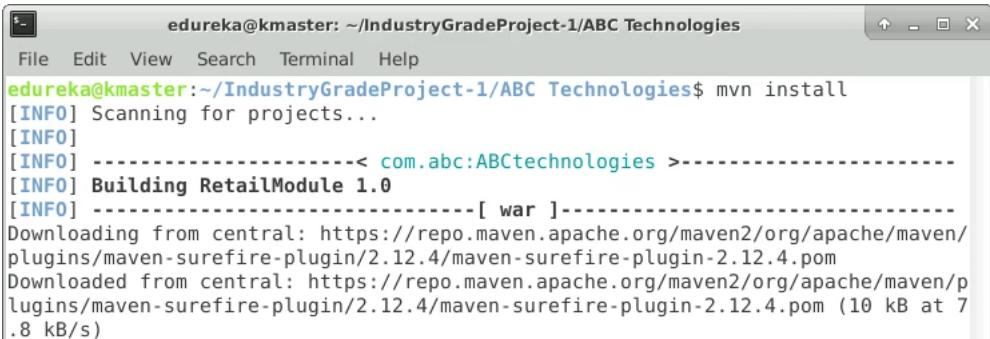


```
edureka@kmaster: ~/IndustryGradeProject-1/ABC Technologies$ mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.abc:ABCtechnologies >-----
[INFO] Building RetailModule 1.0
[INFO] -----[ war ]-----
Downloading from central: https://repo.maven.apache.org/maven2/org/jacoco/jacoco-maven-plugin/0.8.6/jacoco-maven-plugin-0.8.6.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/jacoco/jacoco-maven-plugin/0.8.6/jacoco-maven-plugin-0.8.6.pom (4.4 kB at 2.4 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/jacoco/org.jacoco.build/0.8.6/org.jacoco.build-0.8.6.pom
Downloaded from central: https://repo.maven.apache.org/maven2/junit/junit/3.8.2/junit-3.8.2.jar (121 kB at 57 kB/s)
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 3 source files to /home/edureka/IndustryGradeProject-1/ABC Technologies/target/classes
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 01:36 min
[INFO] Finished at: 2022-10-22T10:54:52Z
[INFO]
```

3. On successfully compilation directory will have target folder.



4. Now build the maven project and install it into local maven repository `mvn install`



```
edureka@kmaster: ~/IndustryGradeProject-1/ABC Technologies$ mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.abc:ABCtechnologies >-----
[INFO] Building RetailModule 1.0
[INFO] -----[ war ]-----
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-surefire-plugin/2.12.4/maven-surefire-plugin-2.12.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-surefire-plugin/2.12.4/maven-surefire-plugin-2.12.4.pom (10 kB at 7.8 kB/s)
```

```

[INFO] Installing /home/edureka/IndustryGradeProject-1/ABC Technologies/target/ABC Technologies-1.0.war to /home/edureka/.m2/repository/com/abc/ABCtechnologies/1.0/ABCtechnologies-1.0.war
[INFO] Installing /home/edureka/IndustryGradeProject-1/ABC Technologies/pom.xml to /home/edureka/.m2/repository/com/abc/ABCtechnologies/1.0/ABCtechnologies-1.0.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 39.267 s
[INFO] Finished at: 2022-10-22T11:01:24Z
[INFO] -----

```

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

:~\$ mvn clean install

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

:~\$ mvn package

7. Maven Run command

8. Mvn exec:java -Dexec.mainClass=

9. Run the test cases in the project

:~\$ mvn test

```

edureka@kmaster: ~/IndustryGradeProject-1/ABC Technologies
File Edit View Search Terminal Help
edureka@kmaster:~/IndustryGradeProject-1/ABC Technologies$ mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] ----- < com.abc:ABCtechnologies > -----
[INFO] Building RetailModule 1.0
[INFO] ----- [ war ] -----
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.6:prepare-agent (jacoco-initialize) @ ABCtechnologies ---
[INFO] argLine set to "-javaagent:/home/edureka/.m2/repository/org/jacoco/org.ja
coco.agent/0.8.6/org.jacoco.agent-0.8.6-runtime.jar=destfile=/home/edureka/Indus
tryGradeProject-1/ABC Technologies/target/jacoco.exec"
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ ABCtechnolo
gies ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/edureka/IndustryGradeProject-1/
ABC Technologies/src/main/resources
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ ABCtechnologies --
-
[INFO] Surefire report directory: /home/edureka/IndustryGradeProject-1/ABC Techn
ologies/target/surefire-reports

-----
T E S T S
-----
Running com.abc.dataAccessObject.ProductImpTest
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.112 sec

Results :

Tests run: 4, Failures: 0, Errors: 0, Skipped: 0

[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 8.694 s
[INFO] Finished at: 2022-10-23T14:23:01Z
[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/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

```
:~$ suod mkdir /opt/tomcat
```

- d. Move all the content to that folder

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

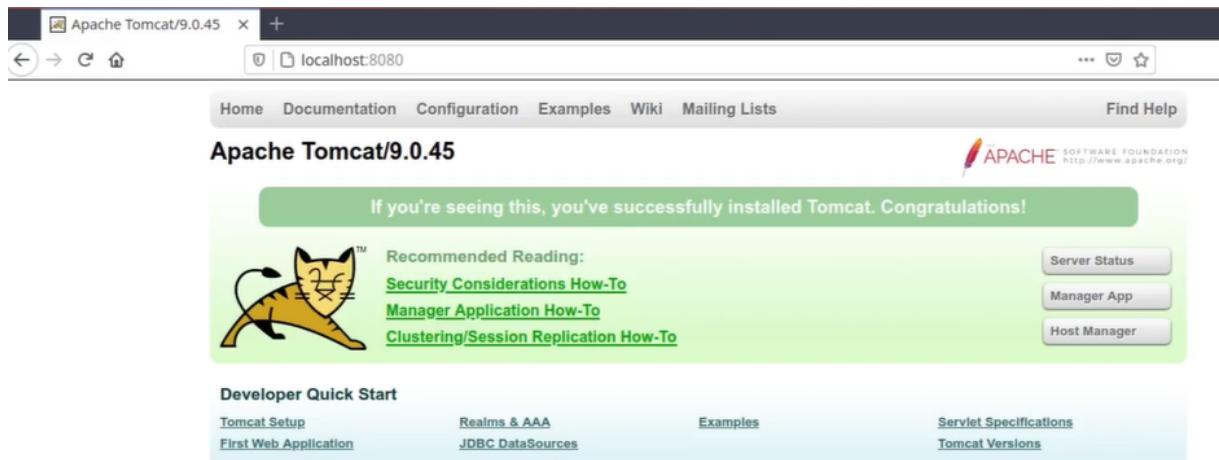
- e. Change directory to /opt/tomcat/bin

- 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

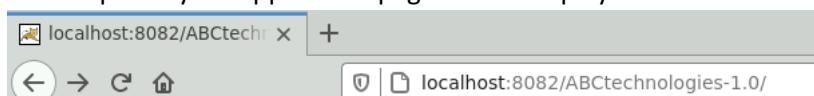
7. Copy the WAR file you have created for ABCtechnologies to /opt/tomcat/webapps

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

8. Start the Tomcat server.

9. In the address area of the browser, type <http://localhost:8080/ABCtechnologies-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 Plugin Manager page. At the top, there are navigation links: Dashboard > Manage Jenkins > Plugin Manager. Below this, there are links to Back to Dashboard, Manage Jenkins, and Update Center. A search bar contains the text "Deploy". There are four tabs at the top right: Updates, Available (which is selected), Installed, and Advanced. The main area displays a list of available plugins. One plugin, "Deploy to container 1.16", is highlighted with a checked checkbox. Its description states: "This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment". Other visible plugins include "Docker Pipeline 528.v7c193a_0b_e67c" which is described as "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 → Mange Plug-ins

Plugin Manager

[Updates](#)

[Available](#)

[Installed](#)

[Advanced](#)

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](#)

Jacoco

Install Name ↓

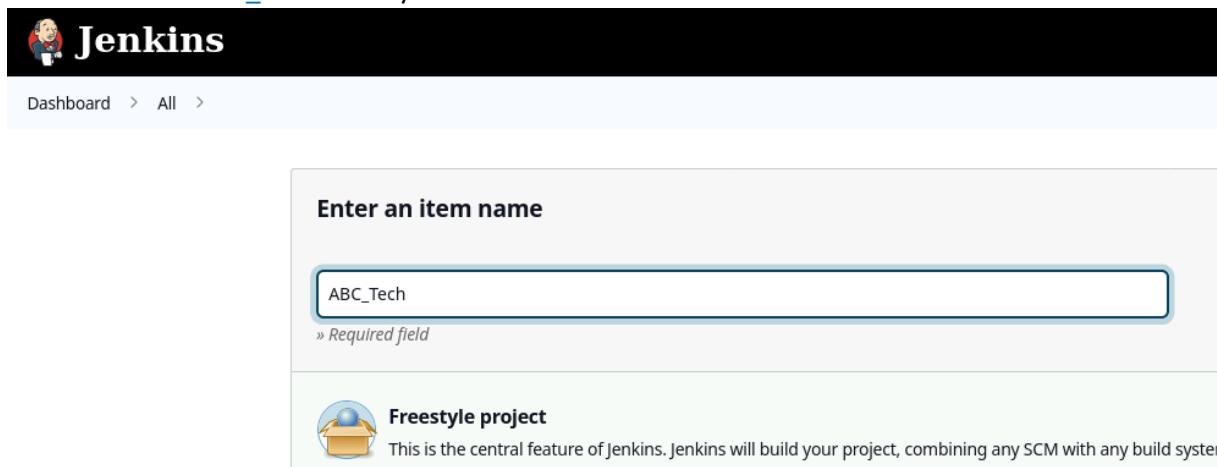


[JaCoCo 3.3.2](#)

[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 screenshot shows the Jenkins interface for creating a new item. At the top, there's a navigation bar with 'Dashboard' and 'All'. Below it is a form titled 'Enter an item name' with a text input field containing 'ABC_Tech'. A note below the field says '» Required field'. To the right of the input field is a 'Freestyle project' icon and its description: 'This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system'.

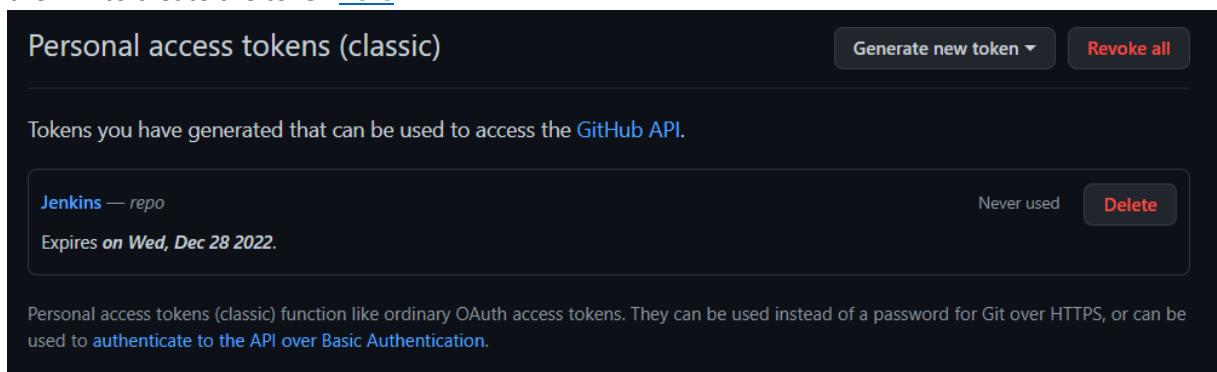
6. Under source code management add GitHub repository

Source Code Management



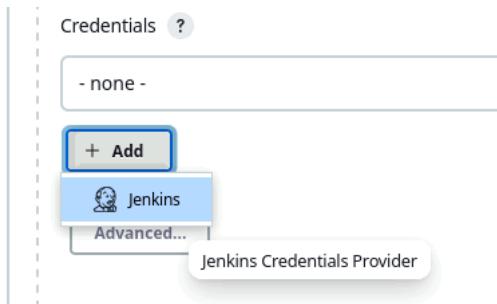
The screenshot shows the 'Source Code Management' configuration for a Jenkins job. It is set to 'Git'. The 'Repositories' section contains one entry: 'Repository URL' is set to 'https://github.com/yogeshk04/IndustryGradeProject-1.git'. The 'Credentials' dropdown is set to '- none -'. There are buttons for '+ Add' and 'Advanced...'.

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



The screenshot shows the GitHub 'Personal access tokens (classic)' page. It displays a single token named 'Jenkins — repo' with the note 'Never used' and a 'Delete' button. The token has an expiration date of 'Wed, Dec 28 2022'. A note at the bottom explains that personal access tokens function like OAuth access tokens and can be used for Git over HTTPS or Basic Authentication.

8. Click on + Add → Jenkins – 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

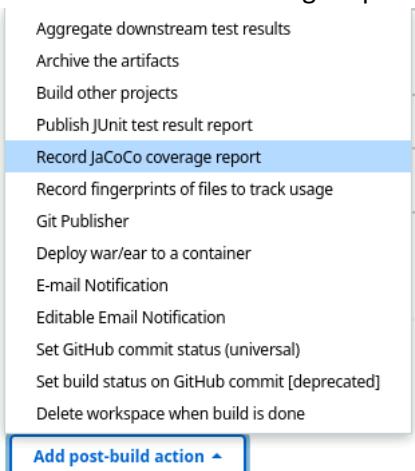
Invoke top-level Maven targets

Maven Version: Maven3

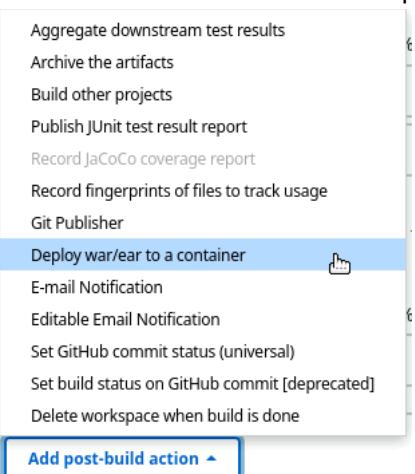
Goals: clean install

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



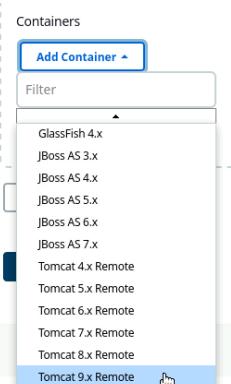
14. Select Add Post-build action → Deploy war/ear to a container



15. Add WAR files

The screenshot shows the configuration for the 'Deploy war/ear to a container' action. It includes fields for 'WAR/EAR files' (containing '**/*.war'), 'Context path' (empty), and a 'Containers' section.

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

The screenshot shows the Jenkins 'Credentials' screen under the 'Tomcat 9.x Remote' section. A blue button labeled '+ Add' is highlighted. Below it, a credential named 'Jenkins' is listed, with a small Jenkins icon and a 'Tomcat URL' dropdown menu. A tooltip for 'Jenkins Credentials Provider' is visible.

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

The screenshot shows the Jenkins 'Credentials' screen under the 'Tomcat 9.x Remote' section. A dropdown menu shows 'tomcat/***** (Tomcat credentials)'. The 'Tomcat URL' field contains 'http://localhost:8082/'. A 'Advanced...' button is at the bottom.

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 directores 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 directores 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
```

The screenshot shows the Jenkins build summary for ABC_Tech #1. It includes a 'Status' section with a green checkmark indicating success, and a 'Jacoco - Overall Coverage Summary' section showing 100% coverage across all metrics.

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

Coverage Report:

Metric	Value
INSTRUCTION	100%
BRANCH	100%
COMPLEXITY	100%
LINE	100%
CLASS	100%

23. Access the web application

The screenshot shows the home page of the ABC Technologies web application. It features a header with the company name and a main content area with a welcome message and navigation buttons for adding and viewing products.

Welcome to ABC technologies

This is retail portal

Add Product **View Product**

24. JaCoCo code coverage report

The screenshot shows the JaCoCo Coverage Report in Jenkins. It displays coverage details for packages like com.abc and com.abc.dataAccessObject, along with overall coverage summaries.

Overall Coverage Summary

name	instruction	branch	complexity	line	method	class
all classes	100% M: 0 C: 55	100% M: 0 C: 0	100% M: 0 C: 12	100% M: 0 C: 18	100% M: 0 C: 12	100% M: 0 C: 2

Coverage Breakdown by Package

name	instruction	branch	complexity	line	method	class
com.abc	M: 0 C: 31 100%	M: 0 C: 0 100%	M: 0 C: 9 100%	M: 0 C: 13 100%	M: 0 C: 9 100%	M: 0 C: 1 100%
com.abc.dataAccessObject	M: 0 C: 24 100%	M: 0 C: 0 100%	M: 0 C: 3 100%	M: 0 C: 5 100%	M: 0 C: 3 100%	M: 0 C: 1 100%

Test result:

Dashboard > ABC_Tech_Test > RetailModule > #3 > Test Results > com.abc.dataAccessObject > ProductImpTest

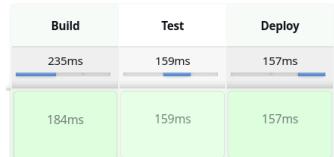
[Back to Project](#) **Test Result : ProductImpTest**

Status: 0 failures 4 tests Took 28 ms. Add description

Changes Console Output Edit Build Information History Executed Mojos Test Result See Fingerprints Redeploy Artifacts Previous Build

All Tests

Test name	Duration	Status
createUserShouldWorkSuccessfully	21 ms	Passed
readDescriptionUserShouldWorkSuccessfully	4 ms	Passed
readPriceUserShouldWorkSuccessfully	3 ms	Passed
readUserShouldWorkSuccessfully	0 ms	Passed



Create Pipeline for the project

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

Enter an item name

» 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.

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.

3. Configure email notification in Jenkins

The screenshot shows the Jenkins 'Configure System' page. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. The main area is titled 'Configure System' and contains sections for 'Home directory' (set to '/var/lib/jenkins'), 'System Message' (a text input field), and 'Build Queue' (which is empty). Below these is the 'E-mail Notification' section, which includes fields for 'SMTP server' (set to 'smtp.gmail.com'), 'Default user e-mail suffix' (empty), 'User Name' (set to 'yogeshk04@gmail.com'), 'Password' (redacted), 'Use SSL' (unchecked), 'Use TLS' (unchecked), 'SMTP Port' (set to '465'), and two buttons at the bottom: 'Save' and 'Apply'.

4. Google has removed this service for low secure application.

Less secure apps & your Google Account

To help keep your account secure, from **May 30, 2022**, Google no longer supports the use of third-party apps or devices which ask you to sign in to your Google Account using only your username and password.

Important: This deadline does not apply to Google Workspace or Google Cloud Identity customers. The enforcement date for these customers will be announced on the Workspace blog at a later date.

For more information, continue to read.

If an app or site doesn't meet our [security standards](#), Google might block anyone who's trying to sign in to your account from it. Less secure apps can make it easier for hackers to get in to your account, so blocking sign-ins from these apps helps keep your account safe.

5. Instead, user <https://app.sendinblue.com/> reference link [here](#)

6. Use App password form google security

← App passwords

App passwords let you sign in to your Google Account from apps on devices that don't support 2-Step Verification. You'll only need to enter it once so you don't need to remember it. [Learn more](#)

Your app passwords

Name	Created	Last used
Jenkins (2)	Nov 1	-

Select the app and device you want to generate the app password for.

Select app ▾ Select device ▾

GENERATE



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

Enter an item name

» 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.

2. Add GitHub credential
3. Add Docker Hub access token into Jenkins
 - a. Create new Access token on your DockerHub account under Security section.

Description	Scope	Last Used	Created	Active
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

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		(global)	dockerhub-token		yogeshk04/******** (Docker hub credentials)

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-docker-deploy*

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](#)

The screenshot shows the Jenkins Pipeline configuration page. It starts with a 'Definition' section set to 'Pipeline script from SCM'. Below this, under 'SCM', 'Git' is selected. A 'Repositories' section contains a single repository with the URL 'https://github.com/yogeshk04/projects.git' and a credential named 'yogeshk04/********'. There are buttons for '+ Add' and 'Advanced...'. An 'Add Repository' button is also present. The 'Branches to build' section shows a 'Branch Specifier' of '*/master'. Under 'Script Path', the value 'ABC_Technologies/Jenkinsfile-docker-deploy' is listed. A checkbox for 'Lightweight checkout' is checked. At the bottom, there's a link to 'Pipeline Syntax'.

5. Pipeline script

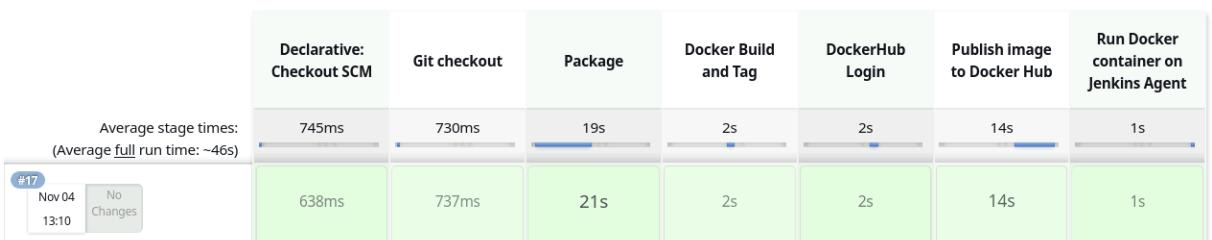
```

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 ABC_Technologies/
                    mvn clean package
                    mv target/*.war target/abctech.war
                    ...
                '''
            }
        }
        stage('Docker Build and Tag') {
            steps {
                sh '''cd ABC_Technologies/
                    sudo docker build -t abctechapp:latest .
                    sudo docker tag abctechapp yogeshk04/abctechapp: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/abctechapp:latest'
            }
        }
        stage('Run Docker container on Jenkins Agent') {
            steps {
                sh "docker run -d -p 8003:8080 yogeshk04/abctechapp"
            }
        }
    }
]

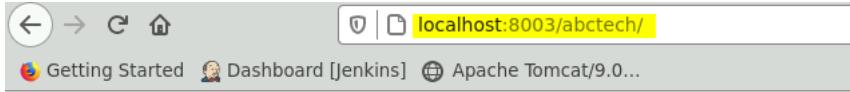
```

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/abctech/> you should see the following web application



Welcome to ABC technologies

This is retail portal

[Add Product](#) [View Product](#)



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
3. https://github.com/yogeshk04/projects/tree/master/ABC_Technologies/kubernetes
4. Create deployment

```
edureka@kmaster:~/projects/ABC_Technologies/kubernetes$ kubectl apply -f app-deployment.yaml
deployment.apps/abctech-deployment created
```

5. Get newly created pods

```
edureka@kmaster:~/projects/ABC_Technologies/kubernetes$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
abctech-deployment-98859f777-4ltfl   1/1     Running   0          107s
abctech-deployment-98859f777-658vw   1/1     Running   0          107s
abctech-deployment-98859f777-ghpg9   1/1     Running   0          107s
edureka@kmaster:~/projects/ABC_Technologies/kubernetes$
```

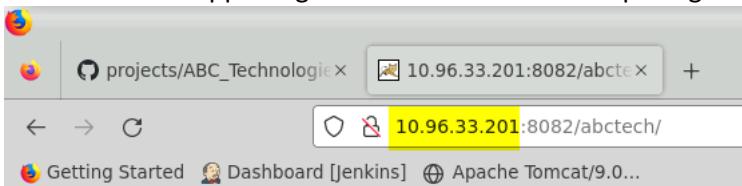
6. Create service

```
edureka@kmaster:~/projects/ABC_Technologies/kubernetes$ kubectl apply -f app-service.yaml
service/abctech-service created
```

7. Get newly created service

```
edureka@kmaster:~/projects/ABC_Technologies/kubernetes$ kubectl get svc
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
abctech-service LoadBalancer  10.96.33.201  <pending>    8082:31900/TCP  40m
kubernetes     ClusterIP  10.96.0.1    <none>       443/TCP       31d
```

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



Welcome to ABC technologies

This is retail portal

[Add Product](#) [View Product](#)



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

```
edureka@kmaster:~/projects/ABC_Technologies$ ls
Dockerfile  ansible      helm-created-manifest.yaml  pom.xml.bak
Jenkins     docker       kubernetes                 src
README.md   helm-charts  pom.xml
edureka@kmaster:~/projects/ABC_Technologies$ kubectl get nodes
NAME      STATUS  ROLES   AGE   VERSION
kmaster   Ready   master   32d   v1.18.3
kslave1   Ready   <none>  32d   v1.18.3
edureka@kmaster:~/projects/ABC_Technologies$
```

4. Deploy application using following helm reference repository directory [helm-charts](#)

Helm deployment command

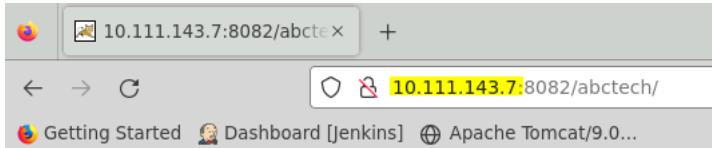
```
:~$ helm install abctechapp helm-chart/
```

```
edureka@kmaster:~/projects/ABC_Technologies$ helm install abctechapp helm-charts/
NAME: abctechapp
LAST DEPLOYED: Sun Nov 13 18:46:02 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
edureka@kmaster:~/projects/ABC_Technologies$ kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
abctech-deployment-9944f7cdf-2ctkj 1/1     Running   2          30h
abctech-deployment-9944f7cdf-q8s9m  1/1     Running   2          30h
abctech-deployment-9944f7cdf-vh6zk  1/1     Running   2          30h
abctechapp-678bf8f98-b2cgr        1/1     Running   0          17s
abctechapp-678bf8f98-ph4q6        1/1     Running   0          17s
edureka@kmaster:~/projects/ABC_Technologies$
```

5. Check the service abctech-svc created by helm chart

```
edureka@kmaster:~/projects/ABC_Technologies$ kubectl get svc
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
abctech-service LoadBalancer  10.96.33.201 <pending>   8082:31900/TCP 30h
abctech-svc    NodePort   10.111.143.7  <none>       8082:31611/TCP 3m36s
kubernetes     ClusterIP  10.96.0.1    <none>       443/TCP   32d
edureka@kmaster:~/projects/ABC_Technologies$
```

6. Access the application using cluster ip



Prometheus: An open-source monitoring system with a dimensional data model, flexible query language, efficient time series database and modern alerting approach.

Grafana: Grafana is a multi-platform open-source analytics and interactive visualization web application. It provides charts, graphs, and alerts for the web when connected to supported data sources

Monitoring using Prometheus and Grafana

1. Add Prometheus community

User command: helm repo add prometheus-community https://prometheus-community.github.io/helm-charts

```
edureka@kmaster:~$ helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
"prometheus-community" has been added to your repositories
edureka@kmaster:~$
```

2. Install Prometheus community release as NAME: prometheus

```
edureka@kmaster:~$ helm install prometheus prometheus-community/prometheus
NAME: prometheus
LAST DEPLOYED: Mon Nov 14 09:26:44 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
The Prometheus server can be accessed via port 80 on the following DNS name from within your cluster:
prometheus-server.default.svc.cluster.local
```

Get the Prometheus server URL by running these commands in the same shell:
export POD_NAME=\$(kubectl get pods --namespace default -l "app=prometheus,component=server" -o jsonpath=".items[0].metadata.name")
kubectl --namespace default port-forward \$POD_NAME 9090

The Prometheus alertmanager can be accessed via port 80 on the following DNS name from within your cluster:
prometheus-alertmanager.default.svc.cluster.local

Get the Alertmanager URL by running these commands in the same shell:
export POD_NAME=\$(kubectl get pods --namespace default -l "app=prometheus,component=alertmanager" -o jsonpath=".items[0].metadata.name")
kubectl --namespace default port-forward \$POD_NAME 9093
WARNING: Pod Security Policy has been moved to a global property.
use .Values.podSecurityPolicy.enabled with pod-based
annotations
(e.g. .Values.nodeExporter.podSecurityPolicy.annotations)
#####

The Prometheus PushGateway can be accessed via port 9091 on the following DNS name from within your cluster:
prometheus-pushgateway.default.svc.cluster.local

Get the PushGateway URL by running these commands in the same shell:
export POD_NAME=\$(kubectl get pods --namespace default -l "app=prometheus,component=pushgateway" -o jsonpath=".items[0].metadata.name")
kubectl --namespace default port-forward \$POD_NAME 9091
For more information on running Prometheus, visit:
<https://prometheus.io/>

3. Check the pods

```
edureka@kmaster:/$ kubectl get pods
NAME                                         READY   STATUS    RESTARTS   AGE
abctech-deployment-9944f7cdf-2ctkj          1/1     Running   4          2d1h
abctech-deployment-9944f7cdf-q8s9m          1/1     Running   4          2d1h
abctech-deployment-9944f7cdf-vh6zk          1/1     Running   4          2d1h
abctechapp-678bf8f98-b2cgr                  1/1     Running   2          18h
abctechapp-678bf8f98-ph4q6                  1/1     Running   2          18h
alertmanager-prometheus-prometheus-oper-alertmanager-0 2/2     Running   0          44s
grafana-77484ccf69-kbzcj                    1/1     Running   1          3h54m
prometheus-grafana-67596ff846-7qp2j        2/2     Running   0          62s
prometheus-kube-state-metrics-c65b87574-4594r 1/1     Running   0          62s
prometheus-prometheus-node-exporter-fcppt    1/1     Running   0          62s
prometheus-prometheus-node-exporter-v5qqh    1/1     Running   0          62s
prometheus-prometheus-oper-operator-5ff8fb5fb-5vdnw 2/2     Running   0          62s
prometheus-prometheus-prometheus-oper-prometheus-0 3/3     Running   1          33s
```

4. Check kubectl get all

```

edureka@kmaster:~$ kubectl get all
NAME                                         READY   STATUS    RESTARTS   AGE
pod/abctech-deployment-9944f7cdf-2ctkj      1/1    Running   3          45h
pod/abctech-deployment-9944f7cdf-q8s9m      1/1    Running   3          45h
pod/abctech-deployment-9944f7cdf-vh6zk      1/1    Running   3          45h
pod/abctechapp-678bf8f98-b2cgr               1/1    Running   1          14h
pod/abctechapp-678bf8f98-ph4q6               1/1    Running   1          14h
pod/prometheus-alertmanager-79ddf8f85c-pr6hd  0/2    Pending    0          6m3s
pod/prometheus-kube-state-metrics-84b6468898-xkl92 1/1    Running   0          6m3s
pod/prometheus-node-exporter-qc7vl            1/1    Running   0          6m3s
pod/prometheus-pushgateway-69d8dbb754-pf9sr   1/1    Running   0          6m3s
pod/prometheus-server-77588f7cd5-tssqj        0/2    Pending    0          6m3s

NAME                           TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/abctech-service       LoadBalancer  10.96.33.201 <pending>     8082:31900/TCP  45h
service/abctech-svc           NodePort     10.111.143.7  <none>        8082:31611/TCP  14h
service/kubernetes             ClusterIP    10.96.0.1    <none>        443/TCP     33d
service/prometheus-alertmanager ClusterIP    10.105.39.111 <none>        80/TCP      6m4s
service/prometheus-kube-state-metrics ClusterIP    10.103.143.139 <none>        8080/TCP    6m4s
service/prometheus-node-exporter ClusterIP    10.97.164.123  <none>        9100/TCP    6m4s
service/prometheus-pushgateway ClusterIP    10.101.176.47  <none>        9091/TCP    6m4s
service/prometheus-server      ClusterIP    10.101.122.218 <none>        80/TCP      6m4s

NAME          DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/prometheus-node-exporter  1          1          1          1           1           <none>      6m3s

NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/abctech-deployment         3/3    3           3           45h
deployment.apps/abctechapp                 2/2    2           2           14h
deployment.apps/prometheus-alertmanager   0/1    1           0           6m3s
deployment.apps/prometheus-kube-state-metrics 1/1    1           1           6m3s
deployment.apps/prometheus-pushgateway    1/1    1           1           6m3s
deployment.apps/prometheus-server          0/1    1           0           6m3s

NAME          DESIRED   CURRENT   READY   AGE
replicaset.apps/abctech-deployment-9944f7cdf 3          3           3           45h
replicaset.apps/abctechapp-678bf8f98          2          2           2           14h
replicaset.apps/prometheus-alertmanager-79ddf8f85c 1          1           0           6m3s
replicaset.apps/prometheus-kube-state-metrics-84b6468898 1          1           1           6m3s
replicaset.apps/prometheus-pushgateway-69d8dbb754 1          1           1           6m3s
replicaset.apps/prometheus-server-77588f7cd5    1          1           0           6m3s
edureka@kmaster:~$
```

5. Create new service to access Prometheus from outside

```

edureka@kmaster:~$ kubectl expose service prometheus-server --type=NodePort --target-port=9090 --name=prometheus-server-ext
service/prometheus-server-ext exposed
edureka@kmaster:~$
```

6. Access the Prometheus using IP address:

The screenshot shows a browser window with the following details:

- Address Bar:** 10.109.96.20:9090/graph
- Toolbar:** Includes icons for Home, Back, Forward, Stop, and Refresh.
- Header:** Prometheus Time Series
- Query Editor:**
 - Enable query history (checkbox)
 - Expression (press Shift+Enter for newlines):
 - Execute button
 - insert metric at cursor -
- Graph Panel:**
 - Graph tab (selected)
 - Console tab
 - Moment dropdown: Moment
- Status Bar:** Getting Started, Dashboard [Jenkins], Apache Tomcat/9.0...

7. Add Grafana repo

```
edureka@kmaster:~$ helm repo add grafana https://grafana.github.io/helm-charts
"grafana" has been added to your repositories
edureka@kmaster:~$
```

8. Update the helm repo

```
edureka@kmaster:~$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "grafana" chart repository
...Successfully got an update from the "prometheus-community" chart repository
...Successfully got an update from the "stable" chart repository
Update Complete. *Happy Helming!*
edureka@kmaster:~$
```

9. Install Grafana

```
edureka@kmaster:~$ helm install grafana stable/grafana
WARNING: This chart is deprecated
NAME: grafana
LAST DEPLOYED: Mon Nov 14 09:49:01 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
*****
****DEPRECATED****
*****
* The chart is deprecated. Future development has been moved to https://github.com/grafana/helm2-grafana

1. Get your 'admin' user password by running:
   kubectl get secret --namespace default grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo
2. The Grafana server can be accessed via port 80 on the following DNS name from within your cluster:
   grafana.default.svc.cluster.local
   Get the Grafana URL to visit by running these commands in the same shell:
   export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=grafana,app.kubernetes.io/instance=grafana" -o jsonpath=".items[0].metadata.name")
   kubectl --namespace default port-forward $POD_NAME 3000
3. Login with the password from step 1 and the username: admin
#####
##### WARNING: Persistence is disabled!!! You will lose your data when #####
#####           the Grafana pod is terminated. #####
#####
edureka@kmaster:~$
```

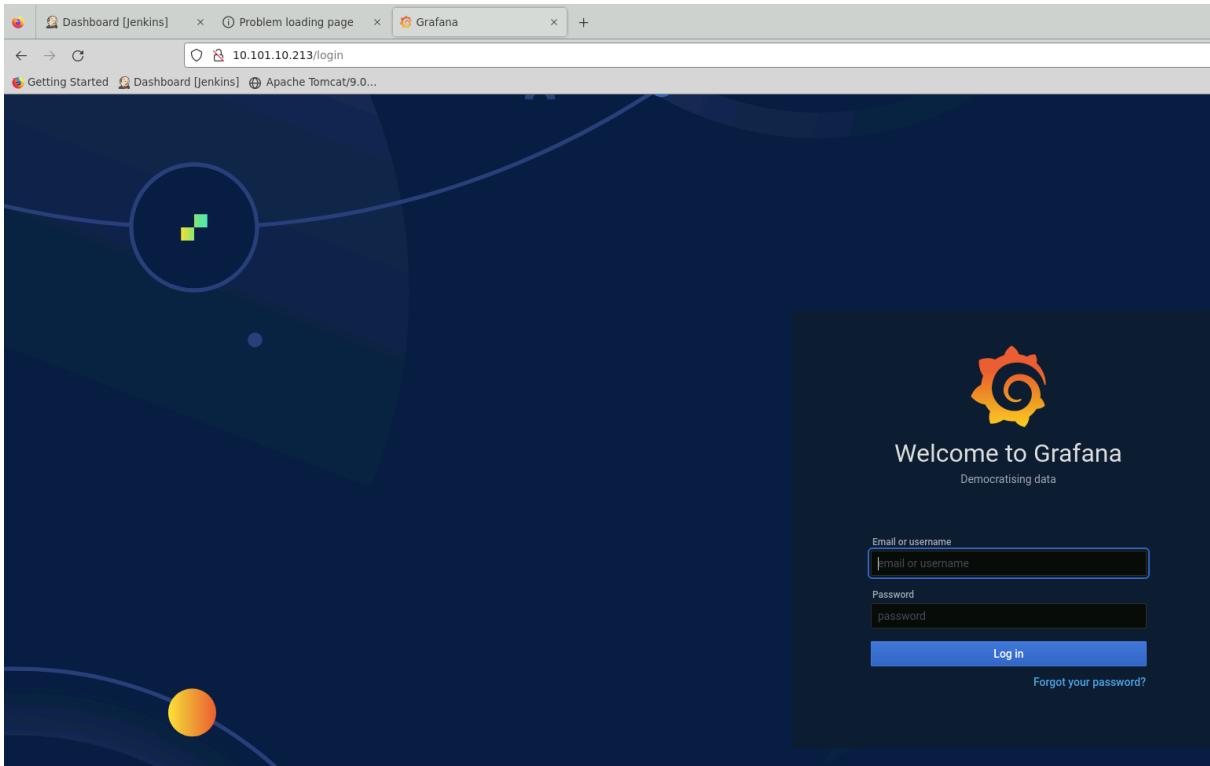
10. Check helm list

NAME	NAMESPACE	REVISION	UPDATED	STATUS	CHART	APP VERSION
abctechapp	default	1	2022-11-13 18:46:02.639324559 +0000 UTC	deployed	abctechapp-0.1.0	1.16.0
grafana	default	1	2022-11-14 09:49:01.997180416 +0000 UTC	deployed	grafana-5.5.7	7.1.1
prometheus	default	1	2022-11-14 09:26:44.887367267 +0000 UTC	deployed	prometheus-15.18.0	2.39.1

11. Create new service to access Grafana from outside

```
edureka@kmaster:~$ kubectl expose service grafana --type=NodePort --target-port=3000 --name=grafana-ext
service/grafana-ext exposed
edureka@kmaster:~$
```

12. Access Grafana using ip address



13. Get the Grafana login id and password

```
edureka@kmaster:~$ kubectl get secret --namespace default grafana -o yaml
apiVersion: v1
data:
  admin-password: RFR1MmJ5MFdIR1RtcFFidDVGbUVpNGlZZkZVUWNsUVd1cU04UnQ5Uw==
  admin-user: YWRtaW4=
  ldap-toml: ""
kind: Secret
metadata:
  annotations:
    meta.helm.sh/release-name: grafana
    meta.helm.sh/release-namespace: default
  creationTimestamp: "2022-11-14T09:49:02Z"
```

14. Admin-user and admin-password will be encrypted you need to decrypt it. Using following command

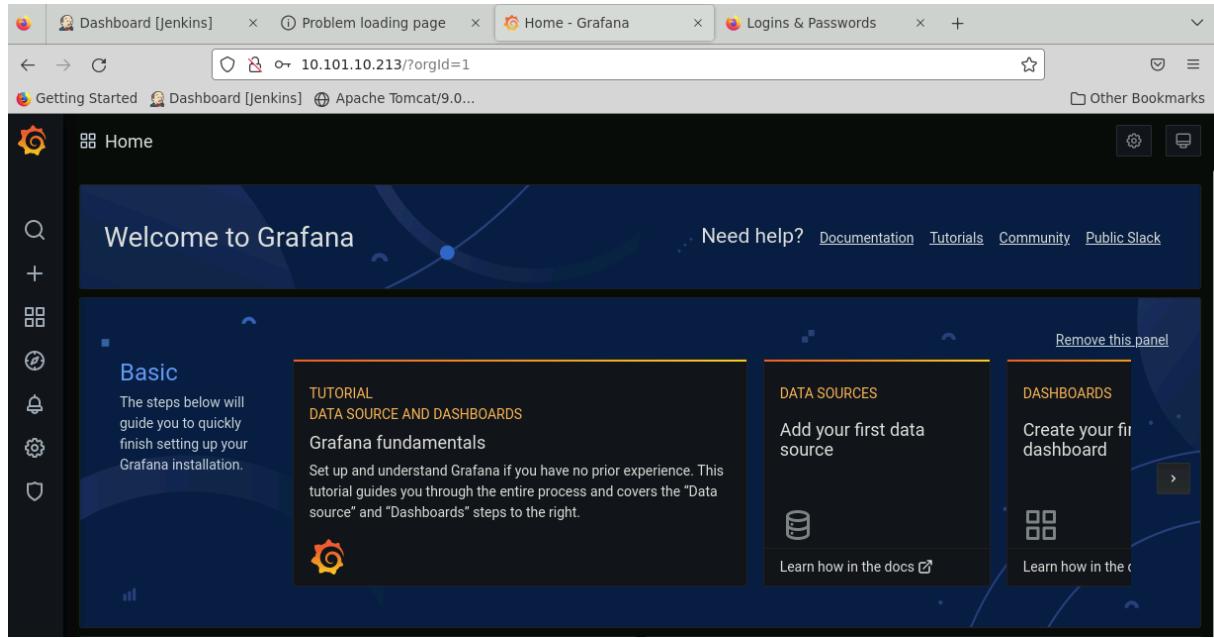
```
edureka@kmaster:~$ echo "YWRtaW4=" | openssl base64 -d ; echo
admin
edureka@kmaster:~$

edureka@kmaster:~$ echo "RFR1MmJ5MFdIR1RtcFFidDVGbUVpNGlZZkZVUWNsUVd1cU04UnQ5Uw==" | openssl base64 -d ; echo
DTu2by0WHGTmpQbt5FmEi4iYffUQclQWuqM8Rt9S
edureka@kmaster:~$
```

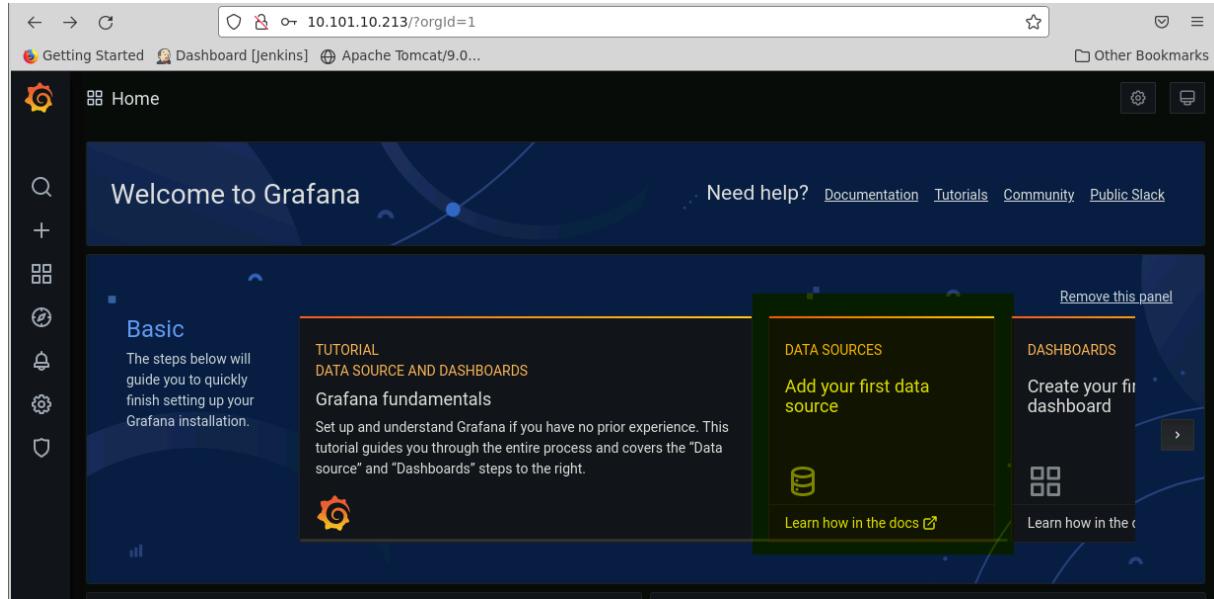
15. Command finds the password:

```
kubectl get secret --namespace default grafana -o jsonpath="{.data.admin-password}" |
base64 --decode ; echo
edureka@kmaster:~$ kubectl get secret --namespace default grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo
DTu2by0WHGTmpQbt5FmEi4iYffUQclQWuqM8Rt9S
edureka@kmaster:~$
```

16. Access the Grafana UI



17. On Grafana UI click on DATA SOURCES



18. Select Prometheus

The screenshot shows the 'Add data source' interface in Grafana. The URL in the browser bar is `10.101.10.213/datasources/new?utm_source=grafana_gettingstarted`. The main title is 'Add data source' with the subtitle 'Choose a data source type'. On the left, there's a sidebar with icons for Home, Getting Started, Dashboard [Jenkins], Apache Tomcat/9.0..., and a gear icon for settings. Below the sidebar is a search bar with the placeholder 'Filter by name or type'. A section titled 'Time series databases' contains two items: 'Prometheus' (selected, highlighted in green) and 'Graphite'. The Prometheus item has a description: 'Open source time series database & alerting' and a 'Core' button.

19. Add URL and Save & Test

The screenshot shows the Grafana configuration interface for a Prometheus data source. The URL in the browser is `10.101.10.213/datasources/edit/1/?utm_source=grafana_gettingstarted`. The main title is "Getting Started" and "Dashboard [Jenkins]". A green banner at the top right says "✓ Datasource updated". The configuration sections include:

- HTTP**: URL is set to `http://10.109.96.20:9090`, Access is "Server (default)", Whitelisted Cookies is "Add Name".
- Auth**: Basic auth is off, With Credentials is on, TLS Client Auth is off, With CA Cert is on, Skip TLS Verify is off, Forward OAuth Identity is off.
- Custom HTTP Headers**: Scrape Interval is 15s, Query timeout is 60s, HTTP Method is Choose.
- Misc**: Disable metrics lookup is off, Custom query parameters is "Example: max_source_resolution=5m&timeout=10".

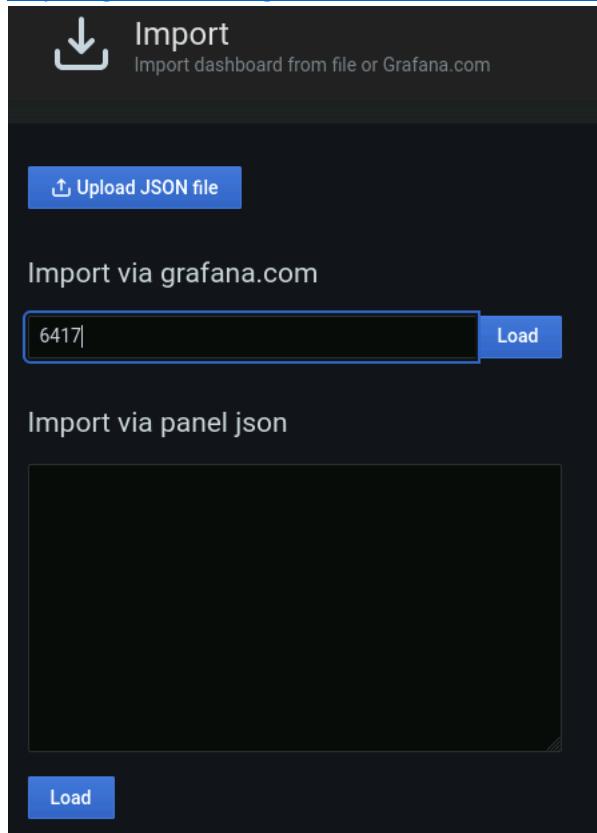
A green banner at the bottom left says "✓ Data source is working". At the bottom are buttons for "Save & Test" (blue), "Delete" (red), and "Back" (grey).

20. Create your dashboard → On Grafana home click + button → Import

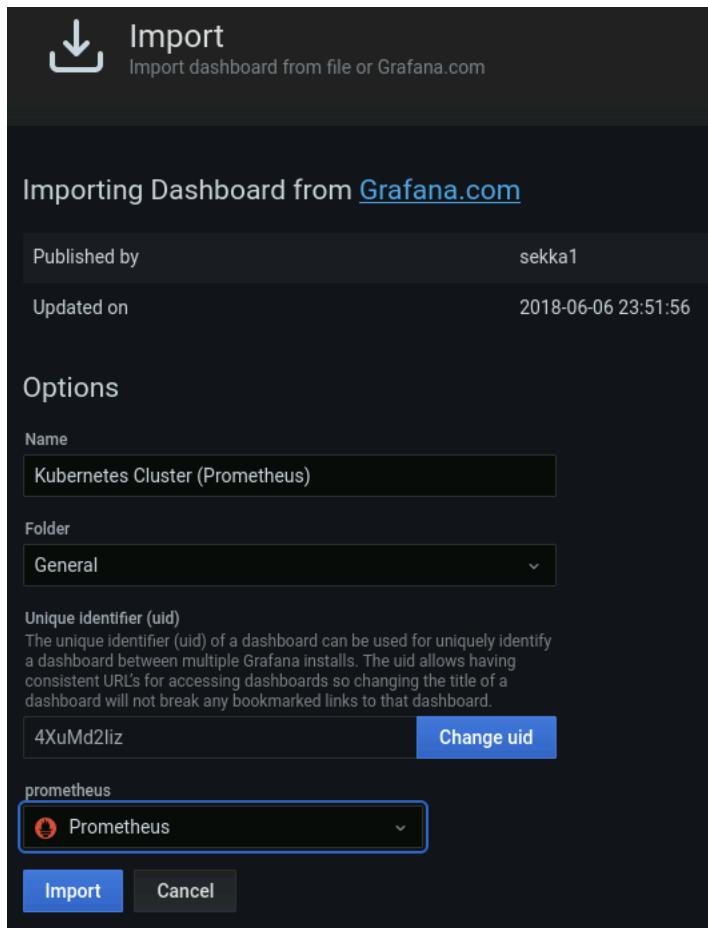
The screenshot shows the Grafana home page. The URL is `10.101.10.213/?orgId=1`. The main title is "Getting Started" and "Dashboard [Jenkins]". The sidebar on the left has a "+ Create" section with options: Dashboard, Folder, and Import (which is highlighted with a red border). A tooltip for Import says: "The steps below will guide you to quickly finish setting up your Grafana installation." To the right, there's a "TUTORIAL" section with links to "DATA SOURCE AND DASHBOARDS", "Grafana fundamentals", and "Set up and understand Grafana".

21. Use the 6417 template number and click on load. Reference url

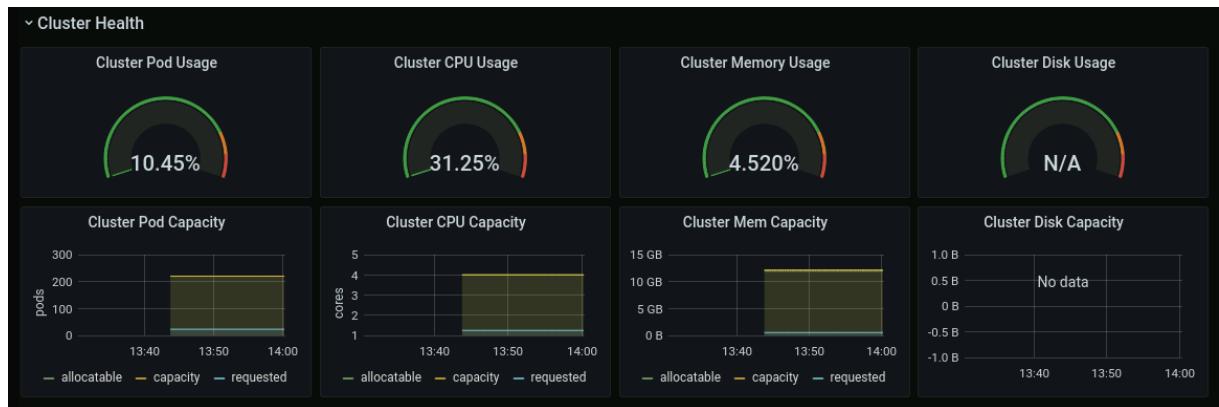
<https://grafana.com/grafana/dashboards/6417-kubernetes-cluster-prometheus/>

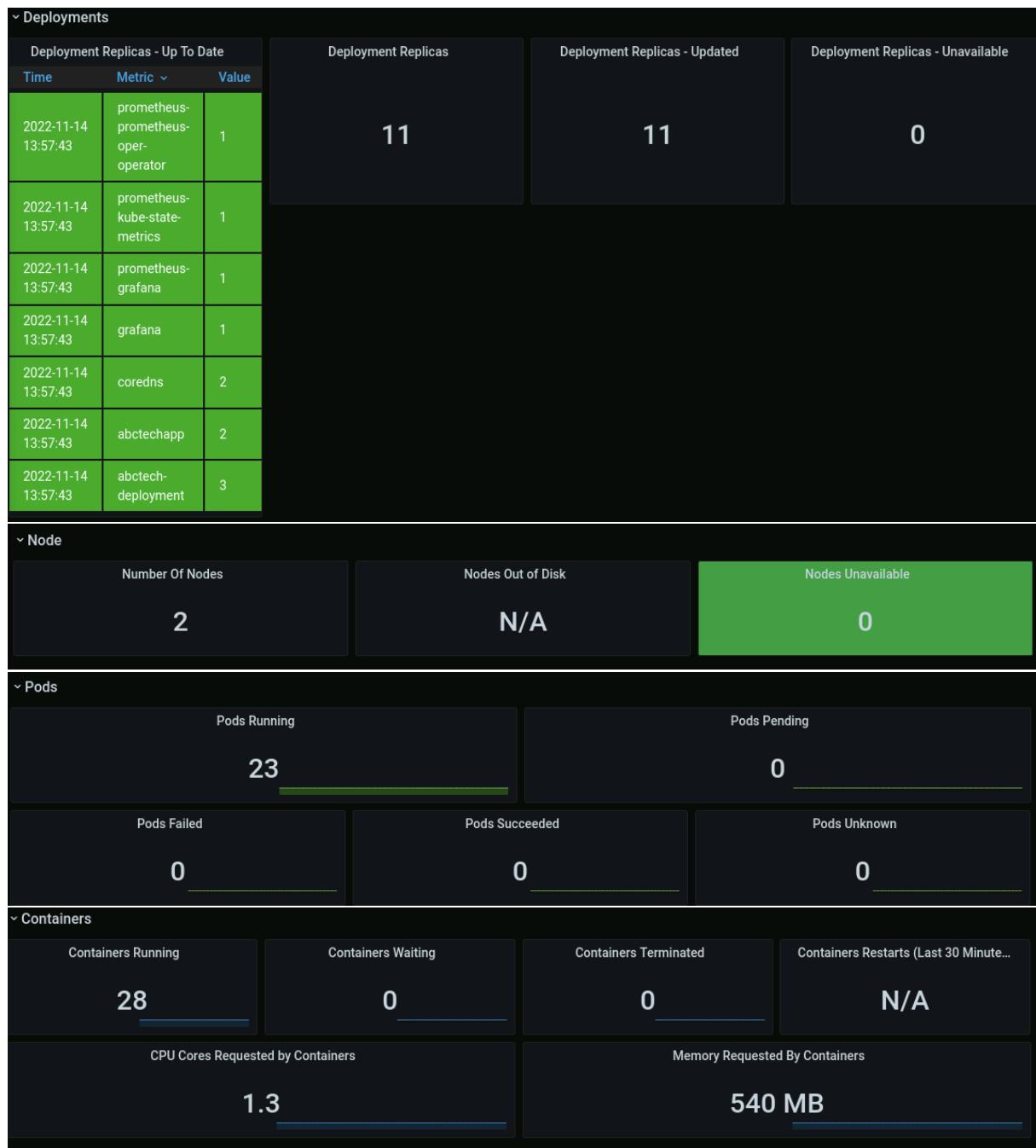


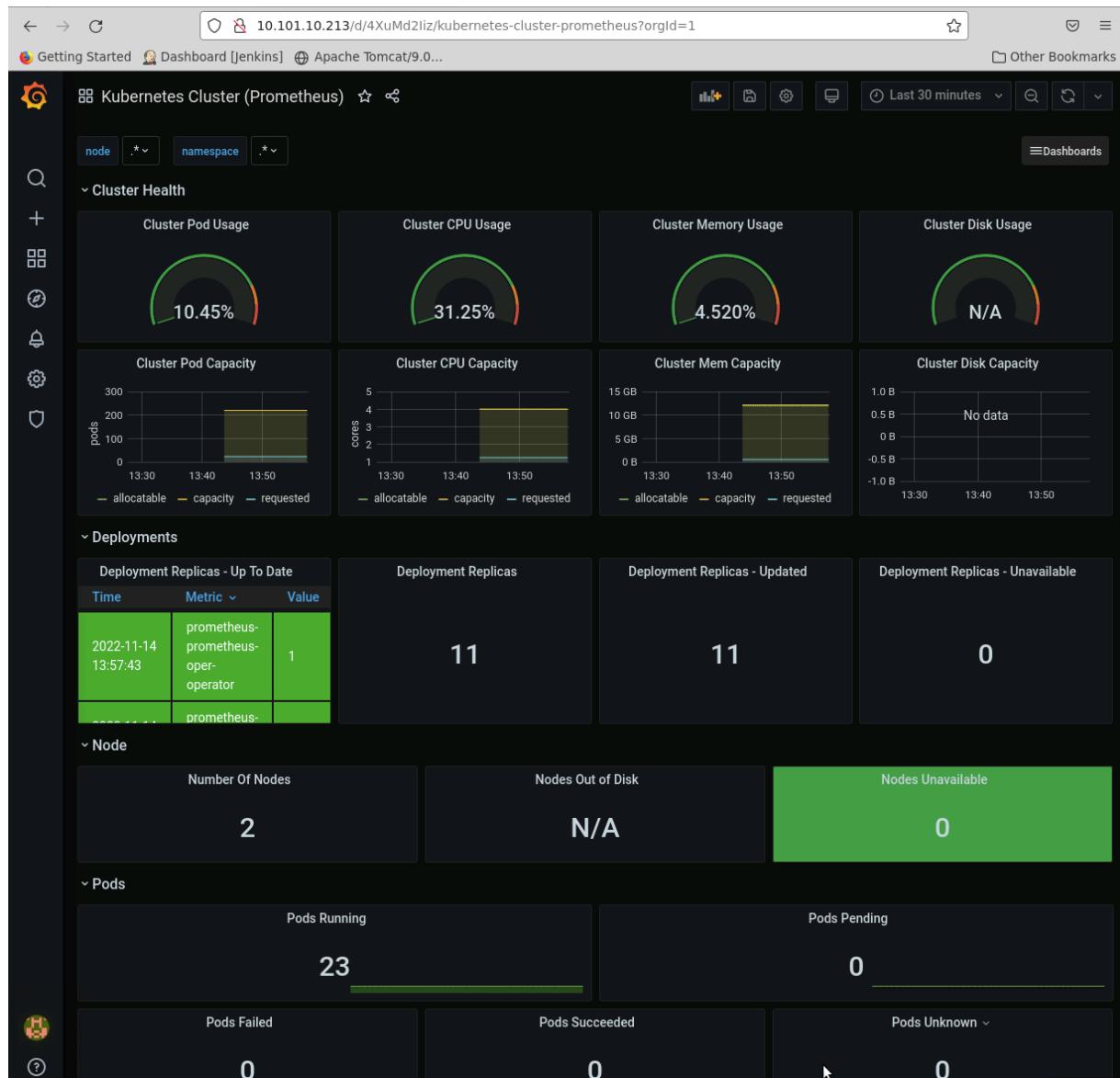
22. Select Prometheus and import



23. It will show the complete dashboard for the Kubernetes cluster and other monitoring activities







This completes the project

The end