

**Post Graduate Certification Program in DevOps**  
**Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V**  
**Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>**

**Business Challenge/Requirement:**

ABC Technologies is a leading online retail store, and it has recently acquired a large retail Offline business store. The business store has a large number of stores across the globe but is following the conventional pattern of development and deployment. As a result, it has landed at a great loss and is facing the following challenges.

- Low available
- Low scalable
- Low performance
- hard to build and maintain
- Developing and deploying are time-consuming

ABC will acquire the data from all these storage systems and plans to use it for analytics and prediction of the firm's growth and sales prospects. In the first phase, ABC has to create the servlets to add a product and display product details. Add servlet dependencies required to compile the servlets. Create an HTML page that will be used to add a product. The team is using Git to keep all the source code.

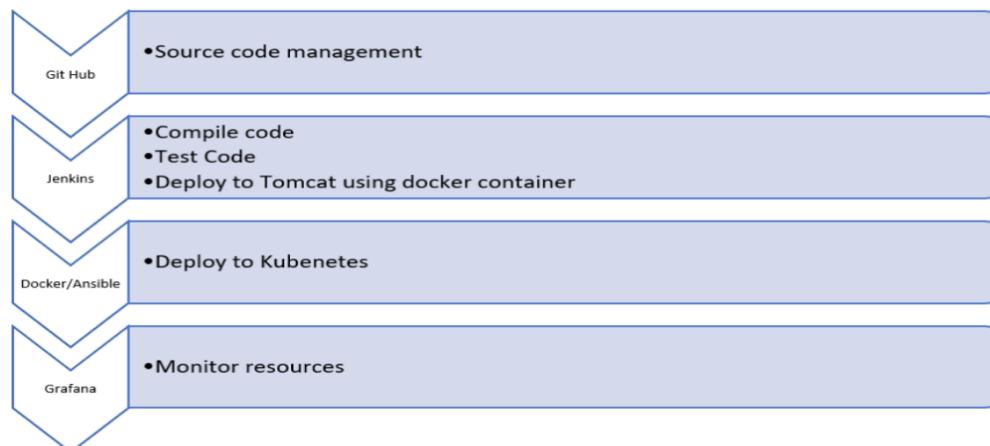
ABC has decided to use the DevOps model. Once source code is available in GitHub, we need to integrate it with Jenkins and provide continuous build generation for continuous delivery as well as integrate with Ansible and Kubernetes for deployment. Use Docker Hub to pull and push images between Ansible and Kubernetes.

**The goal of the Project**

Below are some of the high-level goals of this project:

- Implement CI/CD such that ABC Company is able to be
- highly available
- highly scalable
- highly performant
- easily built and maintained
- developed and deployed quickly

**Data Flow Architecture/Process Flow**



**Post Graduate Certification Program in DevOps**  
**Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V**  
**Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>**

---

## DataExplanationand Schema

A Sample Java project has been shared for usage. It is a Maven project and has src and test folders created in it. It has a POM.xml file that lists all the needed dependencies to execute this project

## ProblemStatements/Tasks

We need to develop a CI/CD pipeline to automate the software development, testing, packaging, and deployment, reducing the time to market the app and ensuring good quality service is experienced by end users. In this project, we need to—

- Push the code to our GitHub repository.
- Create a continuous integration pipeline using Jenkins to compile, test, and package the code present in GitHub.
- Write a Docker file to push the war file to the Tomcat server.
- Integrate Docker with Ansible and write the playbook.
- Deploy artifacts to the Kubernetes cluster
- Monitor resources using Grafana

Prerequisites Verify that the following software is installed in the working machine.

1. Java
2. Maven
3. Git
4. Jenkins
5. Docker
6. Ansible
7. Kubernetes
8. Grafana
9. Prometheus

## ApproachtoSolve:

### **Task 1: Clone the project from the GitHublink shared in resources to your local machine.**

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

- Create a new repository in GitHub and push the initial code to a repository
- <https://github.com/Kannan140494/newproject>
- Copy the industry-grade repo from the Edureka industry-grade project section
- Go to the directory and execute the following commands
  - i. ~\$ git init
  - ii. ~\$ git add .
  - iii. ~\$ git commit -m "adding remote repo"
- Configure git for the first time to GitHub
- Execute following commands
  - i. ~\$ git config --global user.name "Kannan140494/new-project"

# Post Graduate Certification Program in DevOps

## Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V

Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>

- ii. ~\$ git config --global user.email [sree140494@gmail.com](mailto:sree140494@gmail.com)
  - iii. Creating a personal access token to access GitHub – Personal access tokens (PAT) are an alternative to using passwords for authentication to GitHub when using the GitHub API or the command line.
- Push local repo to GitHub
    - i. ~\$ git remote add origin
    - ii. <https://github.com/Kannan140494/newproject>
    - iii. ~\$ git push -set-upstream origin master
  - Go back to GitHub and see that the repository has been updated.

### Setting up the remote git repository

The screenshot shows a GitHub repository page for 'Kannan140494 / newproject'. The repository has 24 commits from 'Seal2a1' made yesterday. The commits include creating .classpath, ansible, src, .project, ABCTech.yml, Ansible\_deployOnk8s.yml, Dockerfile, README.md, ansible.yml, deployment.yml, host, pom.xml, pom.xml.bak, and service.yml. The repository has 0 stars, 1 watching, and 0 forks. It also shows sections for Releases, Packages, and Languages (Java 89.4%, Dockerfile 10.6%).

\*\*\*\*\*

## Task 2: Integrating with Jenkins

**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 and we can send and execute files/commands over SSH.

### Installed Jenkins on the Master node:

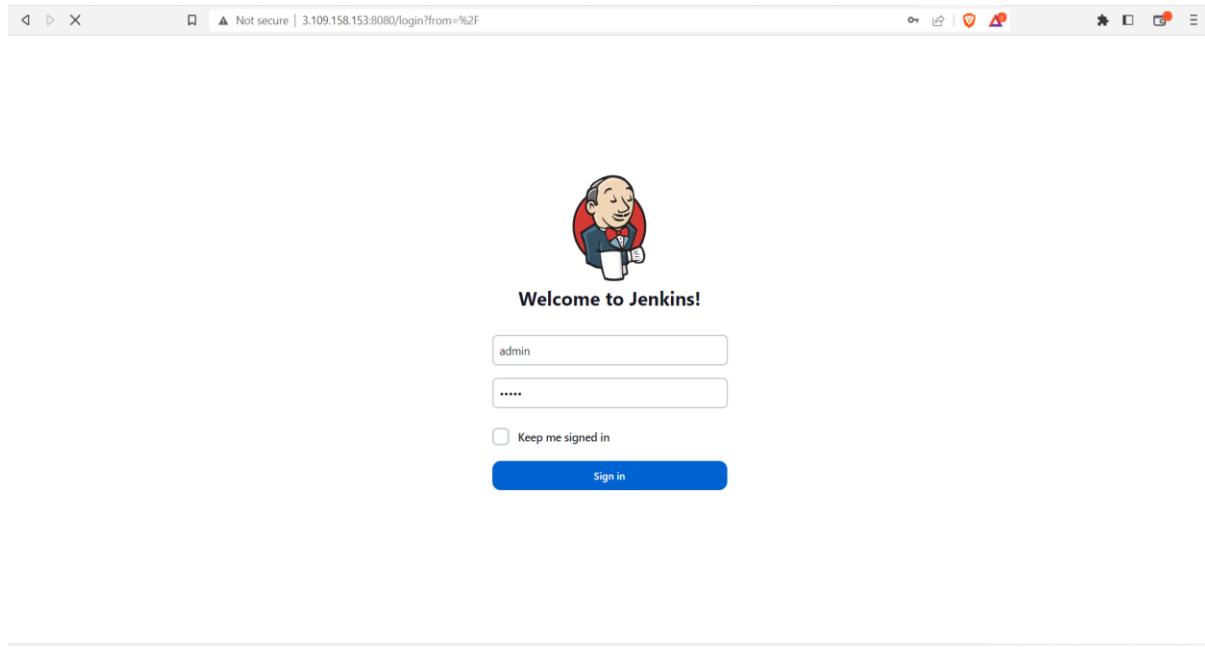
- i. sudo wget <https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installJenkins.sh> -P /tmp
- ii. sudo chmod 755 /tmp/installJenkins.sh
- iii. sudo bash /tmp/installJenkins.sh

**Post Graduate Certification Program in DevOps**  
**Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V**  
**Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>**

**Installed Maven on the Master node:**

- i. sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installMaven.sh -P /tmp
- ii. sudo chmod 755 /tmp/installMaven.sh
- iii. sudo bash /tmp/installMaven.sh

**Setup Jenkins on the Master node:**



**Installed required plugins (Git, Maven, Jacoco, Publish over SSH, etc)**

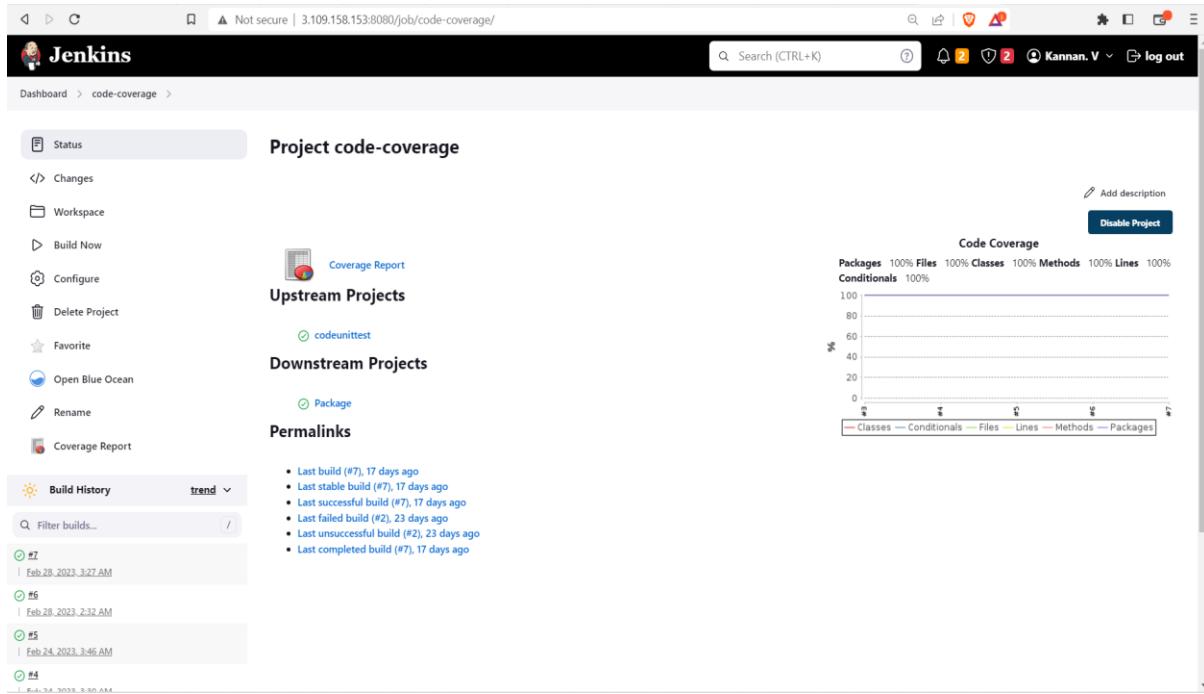
**Navigation: Dashboard >Manage Jenkins >Plugin Manager**

**Used Maven project to create CI and CD pipeline.**

A screenshot of the Jenkins dashboard. On the left, there's a sidebar with links like "New Item", "People", "Build History", "Project Relationship", "Check File Fingerprint", "Manage Jenkins", "My Views", and "Open Blue Ocean". The main area shows a table of build items. The table has columns: S (Status), W (Last Build), Name, Last Success, Last Failure, Last Duration, Coverage, Fav, and # Issues. The table lists several builds: "Code-Compile" (status green, last success 17 days ago, last failure 24 days ago, duration 7.3 sec), "code-coverage" (status green, last success 17 days ago, last failure 22 days ago, duration 14 sec), "codereview" (status green, last success 17 days ago, last failure 24 days ago, duration 11 sec), "codeunittest" (status green, last success 17 days ago, last failure 23 days ago, duration 8.1 sec), "Docker-Package" (status green, last success 3 days 18 hr ago, last failure 4 days 18 hr ago, duration 50 sec), "Package" (status green, last success 17 days ago, last failure N/A, duration 21 sec), and "Test" (status green, last success 29 days ago, last failure N/A, duration 0.15 sec). At the bottom, it shows "Icon legend" and links for "Atom feed for all", "Atom feed for failures", and "Atom feed for just latest builds". It also shows the REST API and Jenkins version (2.375.3).

**Post Graduate Certification Program in DevOps**  
**Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V**  
**Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>**

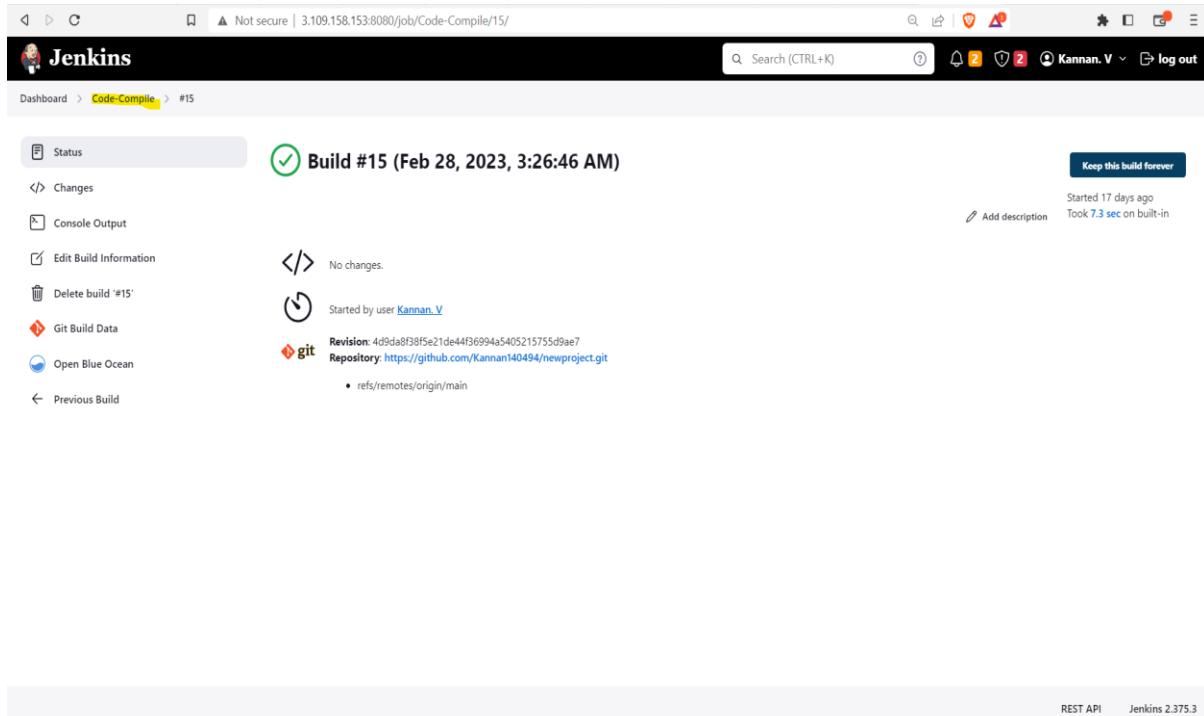
**I have configured upstream and downstream to the jobs which are dependent to create a pipeline.**



The screenshot shows the Jenkins Project code-coverage dashboard. On the left, there's a sidebar with various Jenkins management links like Status, Changes, Workspace, Build Now, Configure, Delete Project, Favorite, Open Blue Ocean, Rename, and Coverage Report. Below that is the Build History section, which lists the last seven builds: #7 (Feb 28, 2023, 3:27 AM), #6 (Feb 28, 2023, 2:32 AM), #5 (Feb 24, 2023, 3:46 AM), and #4 (Feb 24, 2023, 3:46 AM). The main content area is titled "Project code-coverage". It features sections for "Upstream Projects" (Coverage Report, codeunitest) and "Downstream Projects" (codeunitest, Package). There's also a "Permalinks" section. A large chart titled "Code Coverage" displays 100% coverage across Packages, 100% Files, 100% Classes, 100% Methods, and 100% Lines. The Y-axis ranges from 0 to 100. The X-axis has categories: Classes, Conditionals, Files, Lines, Methods, and Packages. Buttons for "Add description" and "Disable Project" are visible at the top right.

**Jobs:**

**Compile job:**

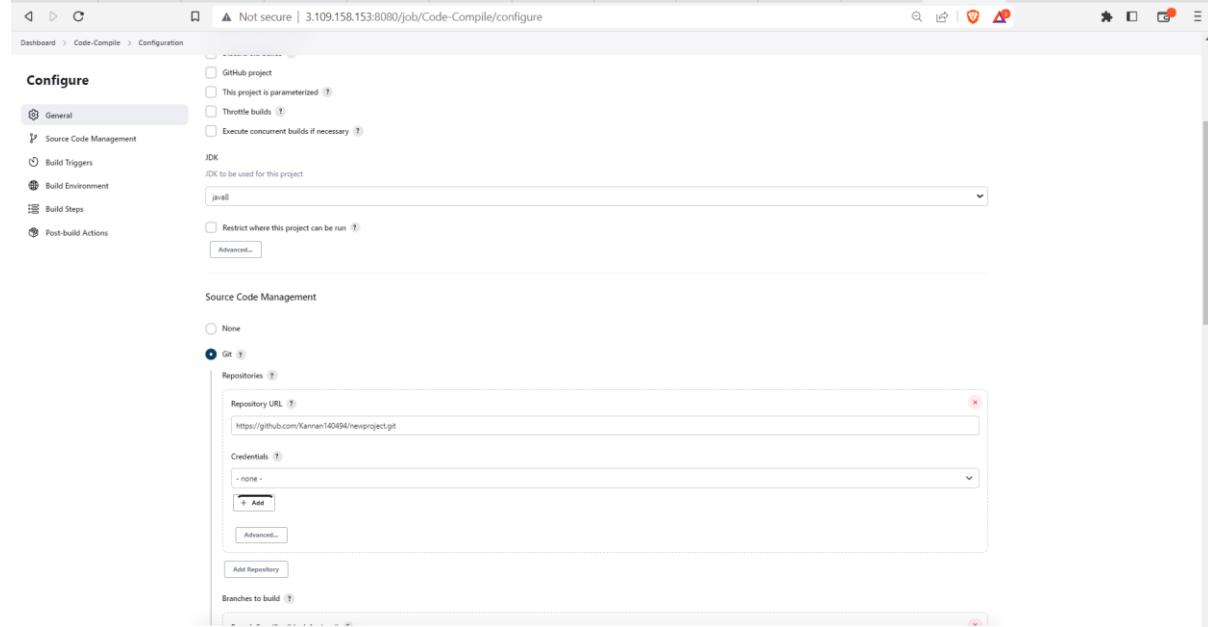


The screenshot shows the Jenkins Code Compile job details page for build #15. The build was successful, indicated by a green checkmark icon. The title is "Build #15 (Feb 28, 2023, 3:26:46 AM)". On the right, there are buttons for "Keep this build forever", "Add description", and status information: "Started 17 days ago" and "Took 7.3 sec on built-in". The left sidebar includes links for Status, Changes, Console Output, Edit Build Information, Delete build #15, Git Build Data, Open Blue Ocean, and Previous Build. The main content area shows the build log output, which is empty ("No changes"). It also displays information about the build: "Started by user Kannan. V", "Revision: 4d9da8f38f5e21de44f36994a5405215755d9ee", and "Repository: https://github.com/Kannan140494/newproject.git". The commit hash shown is 4d9da8f38f5e21de44f36994a5405215755d9ee. At the bottom right, there are links for "REST API" and "Jenkins 2.375.3".

**I have given my Git repo link and it is a public repo.**

**Post Graduate Certification Program in DevOps**  
**Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V**  
**Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>**

**In Configuration:**

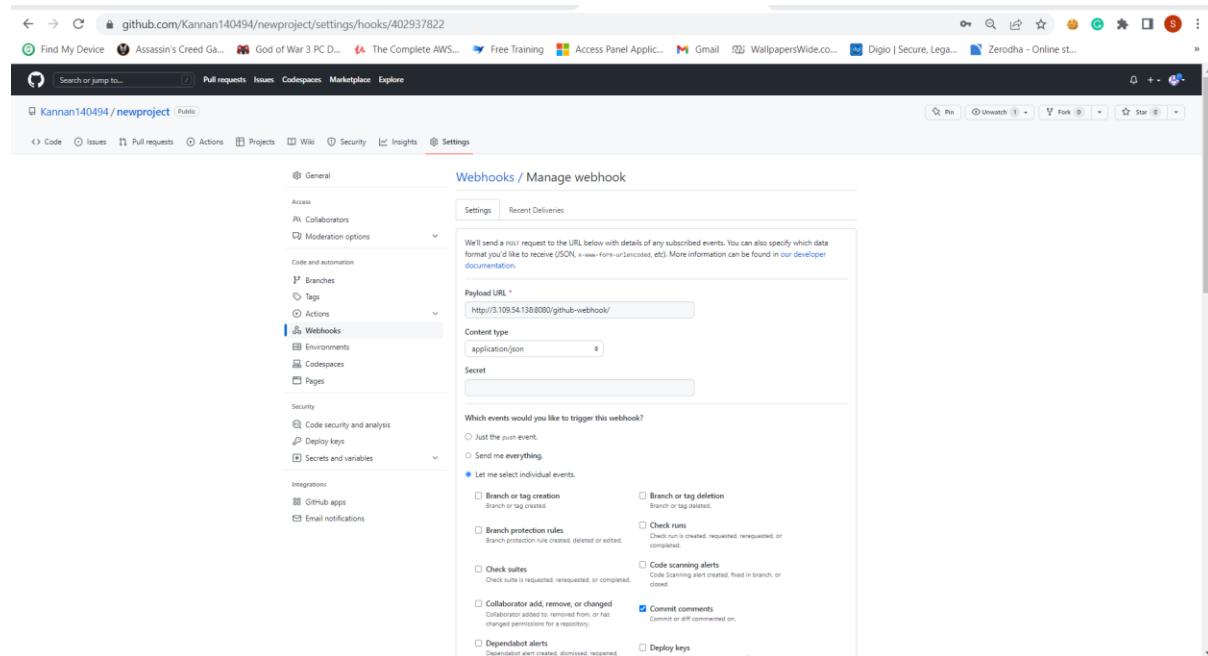


The screenshot shows the Jenkins job configuration page for a job named "Code-Compile". Under the "Source Code Management" section, the "Git" option is selected. The "Repository URL" field contains the value "https://github.com/Kannan140494/newproject.git". Other fields like "Credentials" and "Branches to build" are also visible.

**In Build triggers, I have chosen the GitHub hook trigger for GitSCM Polling. It will trigger the job if anything is pushed and committed to our GitHub repo.**

**We can set the GitHub hook by adding our payload URL/GitHub webhook/ in the setting of webhooks in GitHub**

**\*\*\*\* I have given this webhook trigger only for this compile Job. Later on, Jobs are triggered from Compile job\*\*\*\***



The screenshot shows the GitHub repository settings for "Kannan140494 / newproject". In the "Webhooks" section, a new webhook is being configured. The "Payload URL" is set to "http://3.109.154.138:8080/github-webhook/". The "Content type" is set to "application/json". Under "Which events would you like to trigger this webhook?", the "Commit comments" option is selected, while others like "Push event" and "Code scanning alerts" are disabled.

**As we have given our remote repo link, Jenkins will pull the pom.xml data from there.**

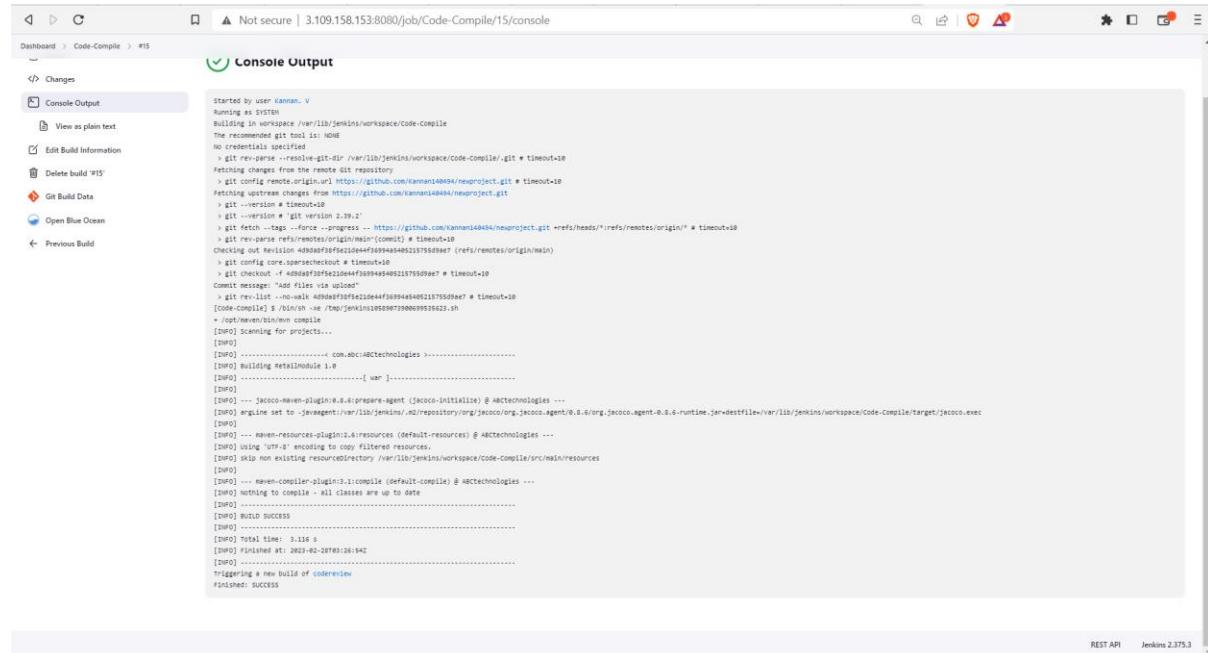
# Post Graduate Certification Program in DevOps

## Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V

---

Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>

### **Build Compile job:**

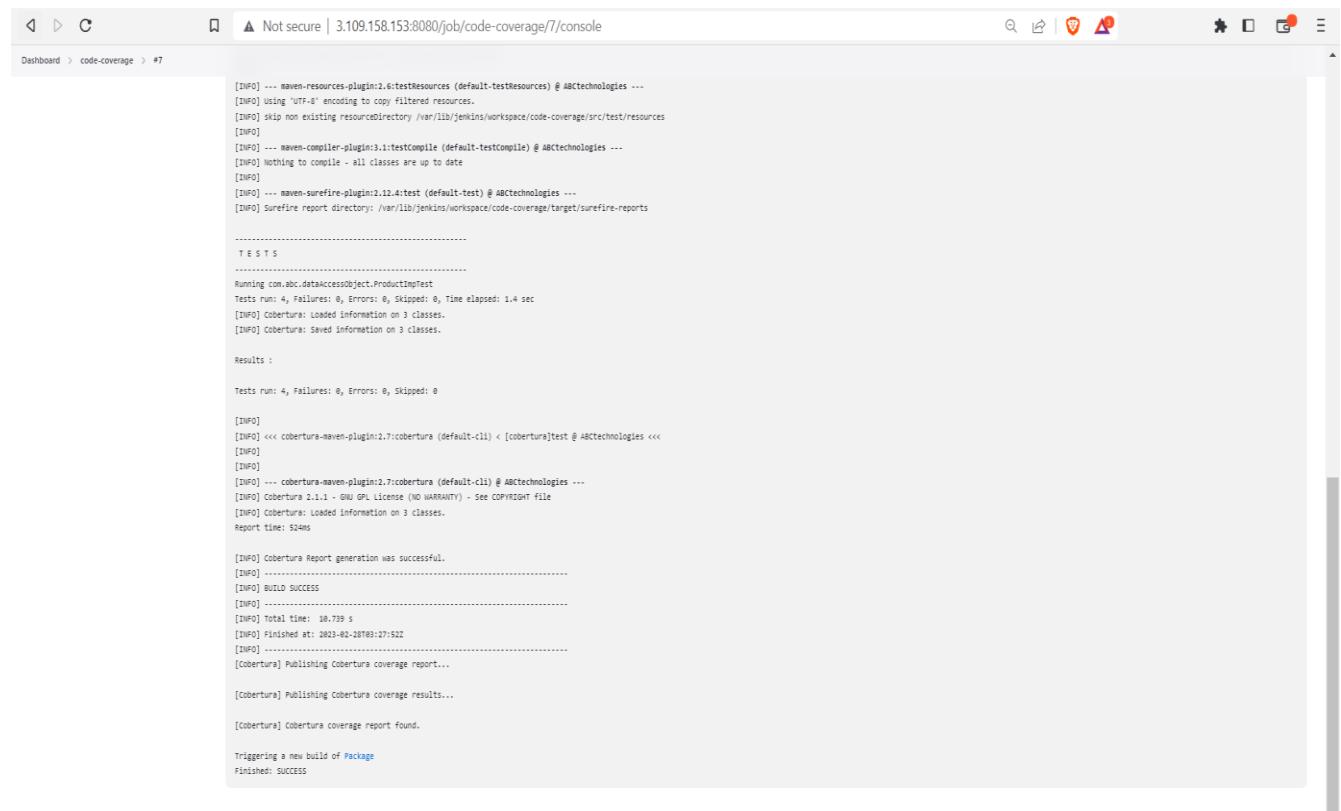


The screenshot shows the Jenkins console output for a build named 'Code-Compile'. The log starts with the command 'git rev-parse --resolve-git-dir' followed by several git fetch and clone operations. It then moves into the Maven directory and runs 'mvn clean install'. The output ends with a success message: 'Finished: SUCCESS'.

```
Started by user Kannan. V
Building in workspace /var/lib/jenkins/workspace/Code-Compile
The repository URL is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Code-Compile/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Kannan140494/newproject.git # timeout=10
Fetching upstream changes from https://github.com/Kannan140494/newproject.git
> git -version # timeout=10
> git fetch --tags --force --progress -- https://github.com/Kannan140494/newproject.git +refs/heads/*:refs/remotes/origin/*
> git reflog refs/remotes/origin/main#(commit) # timeout=10
Checking out Revision 4d0eefaf3ed4f4994a5452127559d8e7 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 4d0eefaf3ed4f4994a5452127559d8e7 # timeout=10
Commit message: "Add files via upload"
> git rev-list --no-walk 4d0eefaf3ed4f4994a5452127559d8e7 # timeout=10
[Code-Compile] $ /bin/sh -e /tmp/jenkins3508077390009935623.sh
* root@jenkins:~$ mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] ... <- com.abc:ABCTechnologies >.....
[INFO] Building RetailModule 1.0
[INFO] ...
[INFO] ... [ war ] .....
[INFO] ...
[INFO] ... Jaccos-maven-plugin:0.8.0:prepare-agent (Jaccos-Initialize) @ ABCTechnologies ...
[INFO] engine set to <plugin>/var/lib/jenkins/.m2/repository/org/jaccos/org.jaccos.agent/0.8.0/jaccos.agent-0.8.0-runtime.jar</plugin>
[INFO] ...
[INFO] ... maven-resources-plugin:1.0:resources (default-resources) @ ABCTechnologies ...
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non-existing resourceDirectory /var/lib/jenkins/workspace/Code-Compile/src/main/resources
[INFO]
[INFO] ... maven-compiler-plugin:3.1:compile (default-compile) @ ABCTechnologies ...
[INFO] nothing to compile - all classes are up to date
[INFO] ...
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 3.11s
[INFO] Finished at: 2023-02-28T03:26:54Z
[INFO] ...
Triggering a new build of codereview
Finished: SUCCESS
```

In the same way, I have Configured Jobs for Test, Code Coverage, and Package

### **Code Coverage:**



The screenshot shows the Jenkins console output for a build named 'code-coverage'. The log starts with Maven configuration for test resources and compiler. It then runs unit tests ('TESTS') using 'com.abc.dataaccessobject.ProductImplTest'. After tests, it generates code coverage reports using Cobertura. The output ends with a success message: 'Finished: SUCCESS'.

```
[INFO] ... maven-resources-plugin:2.6:testResources (default-testResources) @ ABCTechnologies ...
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/code-coverage/src/test/resources
[INFO]
[INFO] ... maven-compiler-plugin:3.1:testCompile (default-testCompile) @ ABCTechnologies ...
[INFO] nothing to compile - all classes are up to date
[INFO]
[INFO] ... maven-surefire-plugin:2.12.4:test (default-test) @ ABCTechnologies ...
[INFO] Surefire report directory: /var/lib/jenkins/workspace/code-coverage/target/surefire-reports
-----
TESTS
-----
Running com.abc.dataaccessobject.ProductImplTest
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.4 sec
[INFO] Cobertura: Loaded information on 3 classes.
[INFO] Cobertura: Saved information on 3 classes.

Results :

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

[INFO]
[INFO] *** cobertura-maven-plugin:2.7:cobertura (default-cli) < [cobertura]test @ ABCTechnologies ***
[INFO]
[INFO]
[INFO] ... cobertura-maven-plugin:2.7:cobertura (default-cli) @ ABCTechnologies ...
[INFO] Cobertura 2.1.1 - GNU GPL License (NO WARRANTY) - See COPYRIGHT file
[INFO] Cobertura: Loaded information on 3 classes.
Report time: 524ms

[INFO] Cobertura Report generation was successful.
[INFO] ...
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 18.739 s
[INFO] Finished at: 2023-02-28T03:27:51Z
[INFO] ...
[Cobertura] Publishing Cobertura coverage report...
[Cobertura] Publishing Cobertura coverage results...
[Cobertura] Cobertura coverage report found.

Triggering a new build of Package
Finished: SUCCESS
```

# Post Graduate Certification Program in DevOps

## Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V

Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>

### Code Review:

The screenshot shows the Jenkins job 'codereview' console output. The log details the execution of a Pmd (Programming Maintenance Delta) analysis. It starts with building the 'RetailModule 1.0' war file. A warning is issued about the 'metrics' profile being disabled because it does not exist. The log then proceeds through various stages of the Pmd analysis, including searching for files, traversing symbolic links, and successfully parsing the pmd.xml configuration. It identifies one issue related to duplicate code. The log ends with a success message.

```
[INFO] Building RetailModule 1.0
[INFO] .....[ war ].....
[INFO] --- maven-pmd-plugin:3.20.0:pmd (default-cli) @ ABCTechnologies ---
[WARNING] Unable to locate source Wef to link to - DISABLED
[INFO] PMD version: 6.53.0
[INFO] Rendering content with org.apache.maven.skins:maven-default-skin:jar:1.1 skin.
[INFO] .....
[INFO] BUILD SUCCESS
[INFO] .....
[INFO] Total time: 7.695 s
[INFO] Finished at: 2023-02-28T00:27:14Z
[INFO] .....
[WARNING] The requested profile "metrics" could not be activated because it does not exist.
[INFO] Searching for all files in '/var/lib/jenkins/workspace/codereview' that match the pattern '**/pmd.xml'
[INFO] Traversing of symbolic links: enabled
[INFO] -> Found 1 file
[INFO] Successfully parsed file /var/lib/jenkins/workspace/codereview/target/pmd.xml
[INFO] -> Found 1 issue (skipped 0 duplicates)
[INFO] Successfully processed file target/pmd.xml
[INFO] Post processing issues on 'Master' with source code encoding 'UTF-8'
[INFO] Creating SCM blamer to obtain author and commit information for affected files
[INFO] -> No blamer installed yet. You need to install the 'git-fingerprint' plugin to enable blaming for GIT.
[INFO] Resolving file names for all issues in workspace '/var/lib/jenkins/workspace/codereview'
[INFO] -> resolved paths in source directory (1 found, 0 not found)
[INFO] Resolving module names from module definition (build.xml, pom.xml, or Manifest.mf files)
[INFO] -> resolved module names for 1 issues
[INFO] Resolving package names (or namespaces) by parsing the affected files
[INFO] -> all affected files already have a valid package name
[INFO] no filter has been set, publishing all 1 issues
[INFO] Creating fingerprints for all affected code blocks to track issues over different builds
[INFO] -> created fingerprints for 1 issues (skipped 0 issues)
[INFO] Copying affected files to Jenkins' build folder '/var/lib/jenkins/jobs/codereview/builds/7/files-with-issues'
[INFO] -> 1 copied, 0 not in workspace, 0 not-found, 0 with I/O error
[INFO] Repository miner is not configured, skipping repository mining
[INFO] Reference build recorder is not configured
[INFO] Obtaining reference build from same job (codereview)
[INFO] Using reference build 'codereview #6' to compute new, fixed, and outstanding issues
[INFO] Issues delta (vs. reference build): outstanding: 1, new: 0, fixed: 0
[INFO] No quality gates have been set - skipping
[INFO] Health report is disabled - skipping
[INFO] Created analysis result for 1 issues (Found 0 new issues, Fixed 0 issues)
[INFO] Attaching ResultWith ID 'pmd' to build 'codereview #7'.
[Checks API] No suitable checks publisher found.
Triggering a new build of codeunittest
Finished: SUCCESS
```

REST API Jenkins 2.375.3

### Code unit test:

The screenshot shows the Jenkins job 'codeunittest' console output. The log details the execution of a Jacoco (Java Code Coverage) analysis and a Surefire (Java Test Framework) test run. It starts with scanning for projects, building the 'RetailModule 1.0' war file, and running the Jacoco agent. The log then moves on to the test phase, where it runs tests for the 'com.abcd:ABCTechnologies' project. It lists various Maven plugins used for compilation and resource handling. The test results show 4 tests run, 0 failures, 0 errors, and 0 skipped. The log concludes with a success message.

```
[INFO] Scanning for projects...
[INFO]
[INFO] .....< com.abcd:ABCTechnologies >.....
[INFO] Building RetailModule 1.0
[INFO] .....[ war ].....
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.6:prepare-agent (jacoco-initialize) @ ABCTechnologies ---
[INFO] agent set to -javaagent:/var/lib/jenkins/.m2/repository/org/jacoco/org.jacoco.agent/0.8.6/org.jacoco.agent-0.8.6-runtime.jar=destfile=/var/lib/jenkins/workspace/codeunittest/target/jacoco.exec
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ ABCTechnologies ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/codeunittest/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ ABCTechnologies ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ ABCTechnologies ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/codeunittest/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ ABCTechnologies ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:xtend (default-test) @ ABCTechnologies ---
[INFO] Surefire report directory: /var/lib/jenkins/workspace/codeunittest/target/surefire-reports
.....
[T E S T S ]
.....
Running com.abcd:datasourceobject:ProductTest
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.129 sec
Results :

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

[INFO] .....
[INFO] BUILD SUCCESS
[INFO] .....
[INFO] Total time: 4.262 s
[INFO] Finished at: 2023-02-28T00:27:31Z
[INFO] .....
Recording test results
[Checks API] No suitable checks publisher found.
Triggering a new build of code-coverage
Finished: SUCCESS
```

REST API Jenkins 2.375.3

# Post Graduate Certification Program in DevOps

## Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V

Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>

### Package:

The screenshot shows the Jenkins dashboard for the 'Package' project. The left sidebar includes options like Status, Changes, Workspace, Build Now, Configure, Delete Maven project, Modules, Favorite, Open Blue Ocean, Rename, and a dropdown for Build History. The main area shows the 'Maven project Package' configuration. It lists 'Last Successful Artifacts' (ABCTechnologies-1.0.war, 6.00 MB) and 'Latest Test Result' (no failures). Below that is an 'Upstream Projects' section with a 'code-coverage' link. A 'Test Result Trend' chart is present, showing a single bar at level 4 across four time points (x1 to x5). The bottom of the page includes links for 'down feed for all' and 'down feed for failures', and footer text 'REST API Jenkins 2.375.3'.

### Package Output:

```
[INFO] [INFO] --- maven-resources-plugin:3.6:testResources (default-testResources) @ ABCtechnologies ---
[INFO] [INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] [INFO] skip non-existing resourceDirectory /var/lib/jenkins/workspace/Package/src/test/resources
[INFO] [INFO] --- maven-compiler-plugin:3.1:compile (default-testCompile) @ ABCtechnologies ---
[INFO] [INFO] Nothing to compile - all classes are up to date
[INFO] [INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ ABCtechnologies ---
[INFO] [INFO] Surefire report directory: /var/lib/jenkins/workspace/Package/target/surefire-reports
-----
[T E S T S ]
-----
Running com.abc.DataAccessObject.ProductImplTest
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.161 sec
Results :
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0

[JENKINS] Recording test results
[INFO]
[INFO] --- maven-war-plugin:3.2.2:war (default-war) @ ABCtechnologies ---
[INFO] Packaging webapp
[INFO] Assembling webapp [ABCtechnologies] in [/var/lib/jenkins/workspace/Package/target/ABCtechnologies-1.0]
[INFO] Processing war project
[INFO] Copying webapp resources [/var/lib/jenkins/workspace/Package/src/main/webapp]
[INFO] Webapp assembled in [138 msec]
[INFO] Building war: [/var/lib/jenkins/workspace/Package/target/ABCtechnologies-1.0.war]
[INFO] --- jacoco-maven-plugin:0.8.4:report (jacoco-site) @ ABCtechnologies ---
[INFO] Loading execution data file: /var/lib/jenkins/workspace/Package/target/jacoco.exec
[INFO] Analyzing bundle 'RetailModule' with 2 classes
[INFO]
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 10.381 s
[INFO] Finished At: 2023-01-25T09:55:12Z
[INFO] -----
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/lib/jenkins/workspace/Package/pom.xml to com.abc/ABCtechnologies/1.0/ABCtechnologies-1.0.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/Package/target/ABCtechnologies-1.0.war to com.abc/ABCtechnologies/1.0/ABCtechnologies-1.0.war
channel stopped
Archiving artifacts
Finished: SUCCESS
```

As I have given upstream and downstream to these jobs, once any commit is pushed to the GitHub repo, webhook will trigger the compile job, and later compile job will trigger the test and package respectively according to the stream. This acts as a pipeline.

\*\*\*\*\*

Post Graduate Certification Program in DevOps  
Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V

---

Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>

### **Task 3: Deployment with Docker**

**Docker:** Docker is a set of platforms as a service product that use OSlevel 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.

**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. Here I'm using the Tomcat container to deploy our /.war file.

#### **Write a Docker file:**

- FROM docker.io/library/ubuntu:18.04
- RUN apt-get -y update && apt-get -y upgrade
- RUN apt-get -y install openjdk-8-jdk wget
- RUN mkdir /usr/local/tomcat
- ADD https://downloads.apache.org/tomcat/tomcat-9/v9.0.72/bin/apache-tomcat-9.0.72.tar.gz /tmp/apache-tomcat-9.0.72.tar.gz
- RUN cd /tmp && tar xvfz apache-tomcat-9.0.72.tar.gz
- RUN cp -Rv /tmp/apache-tomcat-9.0.72/\* /usr/local/tomcat/
- ADD \*/\*.war /usr/local/tomcat/webapps
- EXPOSE 8080
- CMD /usr/local/tomcat/bin/catalina.sh run

#### **Install Docker:**

- i. sudo wget <https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installDocker.sh> -P /tmp
- ii. sudo chmod 755 /tmp/installDocker.sh
- iii. sudo bash /tmp/installDocker.sh

#### **Deploying .war file generated from package job into the tomcat server**

**Issue 1:** As part of Tomcat deployment I have tried to deploy using shell commands by copying the .war file to the web apps/ folder but it did not work and I got the below error at that time.

#### **Error message:**

- "sudo: no tty present and no askpass program specified
- Build step 'Execute shell' marked build as a failure
- Finished: FAILURE"

I thought to copy the SSH public file to the slave machine but it not worked out. So finally I followed a new way to deploy the .war file.

**Solution:** Added the 'Deploy to container' plugin and modified the packaging job and configured the post-build action to deploy to the tomcat server.

**Post Graduate Certification Program in DevOps**  
**Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V**

---

Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>

**Issue 2:** After running the job I got an issue and was not able to get into the tomcat so I have configured the tomcat conf/tomcat-users.xml file with the below added to the end like I have created a user as a tomcat and a password as an admin for the role of manager-GUI and manager\_script and later it allowed me to deployed to the tomcat successfully.

**Solution:**

- conf/tomcat-users.xml
- <role rolename="manager-gui"/>
- <role rolename="manager-script"/>
- <user username="tomcat" password="admin" roles="manager-gui,manager-script"/>

**Issue 3:** While running the build I got an issue that Jenkins was not able to deploy to the tomcat and figure out that whenever we restarted Lab tomcat will not start automatically so I have added one more build stage as Execute shell to start the tomcat and also got error even after this because tomcat will not initialize immediate start so I have included sleep 10 commands as well for it to initialize totally.

**Solution:**

- sudo bash /opt/tomcat/bin/startup.sh
- sleep 10

**Step 2:** Create the docker build and docker container with the above .war file generated from the package command and upload the docker image to the docker hub and run this docker image as a container.

- As part of building docker images and uploading to the docker hub and running in the docker container.

I tried to build the docker image manually locally and it got successful as part of that I created a Dockerfile and added the plugins "cloud bees docker build and publish", "docker pipeline", "docker plugin", " docker build step".

After installation of plugins, you should be able to see the build/publish docker image in the build section of Jenkins job and one more option "docker build and publish" if you see that your plugin is installed successfully.

For telling Jenkins to prepare the docker image we need to build the DockerFile, now we need to give instructions to Jenkins but it will read this Dockerfile in the home directory so I have placed this file in the home directory and after build, I have pushed the docker image to the docker hub for that I need to give my docker hub credentials to Jenkins so for giving credentials I have configured the job in the build section "Docker build and publish" and validated in local and docker hub after build the job.

**Issue 4:** Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get <http:///%2Fvar%2Frun%2Fdocker.sock/v1.40/containers/json>: dial unix /var/run/docker.sock: connect: permission denied

# Post Graduate Certification Program in DevOps

## Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V

---

**Project Remote Git Repository Link:** <https://github.com/Kannan140494/newproject>

**Solution:** I have added the below commands in the newly created build section Execute shell before starting to build the docker image

- sudo chmod 666 /var/run/docker.sock
- sudo chmod 777 /var/lib/jenkins/.docker/config.json
- Sudo chmod 777 .docker

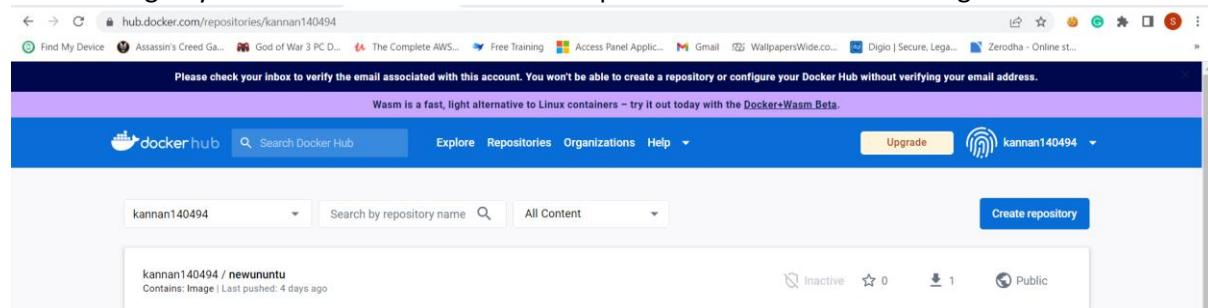
And also added the below additions in the /etc/sudoers file(sudo file)

Jenkins ALL=(ALL) NOPASSWD: ALL

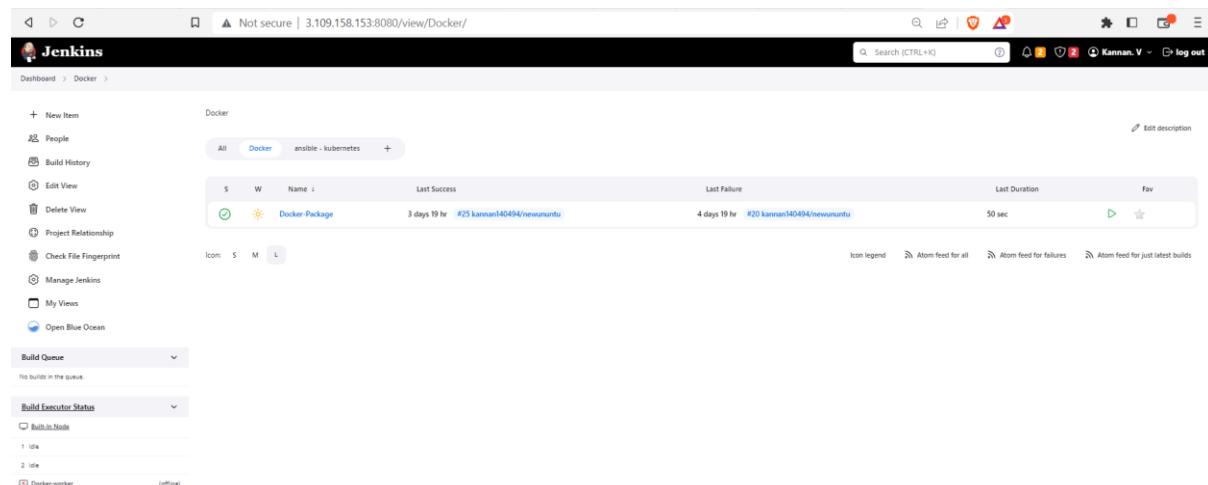
**Issue 5:** While building a docker image with Dockerfile it is not recognizing the .war file path that I have mentioned in Dockerfile and got the below error

- ((ADD /var/lib/jenkins/workspace/package\_kubernetes2/target/ABCtechnologies-1.0.war
- /usr/local/tomcat/webapps ADD failed: file not found in build context or excluded by
- .dockerignore: stat
- var/lib/jenkins/workspace/package\_kubernetes2/target/ABCtechnologies-1.0.war))

**Solution:** So I modified the Dockerfile with the below changes \*\*/\*.war instead of target/\*.war  
Steps to configure the job for docker integration after installing the above-mentioned plugins Create the build tab with the maven package and then the ‘docker build and publish’ plugin added and go to configure In build tab -> select ‘docker build and publish’ -> Repository name(kannan140494) -> and add Registry credentials with ‘username and password’- and that’s it it will get success.



**Below is the screenshot for the above tasks:**



**Post Graduate Certification Program in DevOps**  
**Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V**  
**Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>**

The image consists of three vertically stacked screenshots of a Jenkins job configuration page, specifically for a 'Docker-Package' job named 'Docker'.

**Screenshot 1: Configuration - General Tab**

This screen shows the 'General' tab selected. It includes fields for 'JDK' (set to 'Java 11'), 'Label Expression' (set to 'Devops'), and 'Source Code Management' (set to 'Git' with a single repository URL: <https://github.com/Kannan140494/newproject.git>). Other tabs visible include 'Build Triggers', 'Build Environment', 'Pre Steps', 'Build', 'Post Steps', 'Build Settings', and 'Post-build Actions'.

**Screenshot 2: Configuration - Build Tab**

This screen shows the 'Build' tab selected. It includes fields for 'Root POM' (set to 'pom.xml') and 'Goals and options' (set to 'package'). Under 'Post Steps', the 'Run regardless of build result' option is selected. A 'Execute shell' step is defined with the command:

```
#whoami  
#sudo chmod 666 /var/run/docker.sock  
#sudo chmod 777 /var/lib/jenkins/.docker/config.json
```

**Screenshot 3: Configuration - Post Steps Tab**

This screen shows the 'Post Steps' tab selected. It includes fields for 'Repository Name' (set to 'kannan140494/neununtu'), 'Tag', 'Docker Host URI', 'Server credentials' (set to 'none'), 'Docker registry URL', 'Registry credentials' (set to 'kannan140494\*\*\*\*\* (DOCKERHUB\_CREDS)'), and an 'Execute shell' step with the command:

```
echo "Successfully build and uploaded to dockerhub"  
sudo docker run -d -P kannan140494/neununtu  
echo "Successfully deployed to the docker container by pulling the image from dockerhub"
```

# Post Graduate Certification Program in DevOps

Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V

Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>

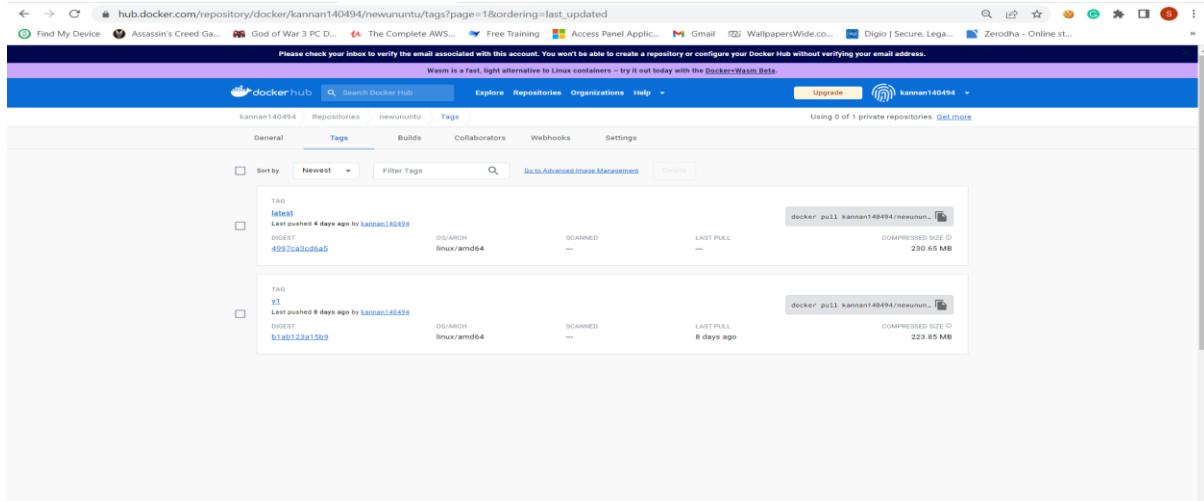
The screenshot shows three browser tabs related to a Jenkins Docker Pipeline job:

- Plugins**: A list of available Jenkins plugins, with the "Docker" plugin selected. It includes details about the Docker plugin, its version (1.279), and its functionality for building Dockerfile-based projects and publishing built images to a registry.
- Console Output**: The Jenkins log for the Docker Pipeline job. It shows the Maven build process, including the execution of `mvn clean package`, the creation of a Docker image named `kannan140494/neununtu`, and the successful deployment of the image to a Docker container.
- Console Output**: Another view of the Jenkins log for the same job, showing the Maven build and Docker image creation steps.

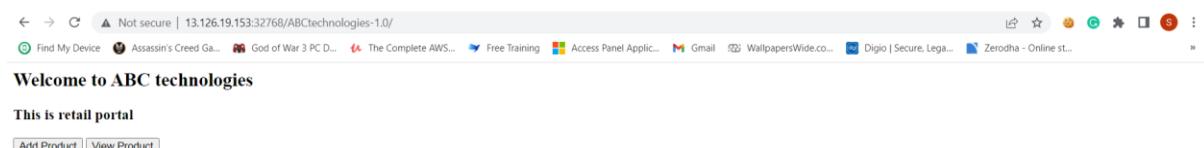
# Post Graduate Certification Program in DevOps

## Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V

### Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>



A screenshot of the Apache Tomcat 9.0.38 documentation homepage. The main header includes links for Home, Documentation, Configuration, Examples, Wiki, and Mailing Lists, along with a Find Help button. The title 'Apache Tomcat/9.0.38' is prominently displayed. A central green banner says 'If you're seeing this, you've successfully installed Tomcat. Congratulations!' Below it is a cartoon cat icon. To the right are buttons for Server Status, Manager App, and Host Manager. The page is divided into sections: 'Developer Quick Start' (Tomcat Setup, First Web Application), 'Realms &amp; AAA', 'Examples', 'Servlet Specifications', 'Tomcat Versions', 'Managing Tomcat' (with instructions for restricted access), 'Documentation' (links to Tomcat 9.0 Documentation, Configuration, and Wiki), and 'Getting Help' (links to FAQ and Mailing Lists like tomcat-announce, tomcat-users, and taglibs-user).



Created the Jenkins pipeline by adding this job downstream to the packaging job. Once any new package is built a new Image is built over the docker host and build the required container with a \*.war file in it.

\*\*\*\*\*

# Post Graduate Certification Program in DevOps

## Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V

Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>

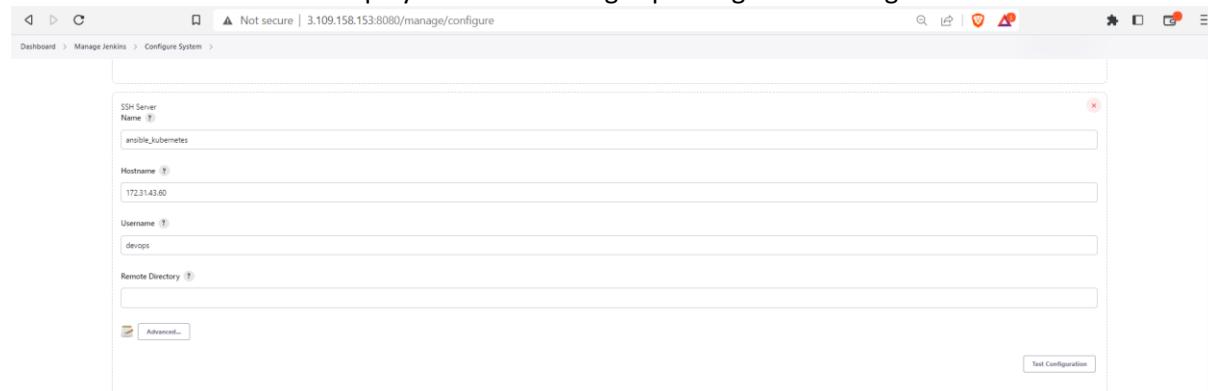
**Task 4:** Integrate the Docker host with Ansible. Write an Ansible playbook to create an image and create a continuer. Integrate Ansible with Jenkins. Deploy Ansible-playbook. CI/CD job to build code on ansible and deploy it on the docker container

- Deploy Artifacts on Kubernetes
- Write pod, service, and deployment manifest file
- Integrate Kubernetes with Ansible
- Ansible playbook to create deployment and service

**Ansible:** Ansible is an open-source IT automation engine that automates provisioning, configuration management, application deployment, orchestration, and many other IT processes.

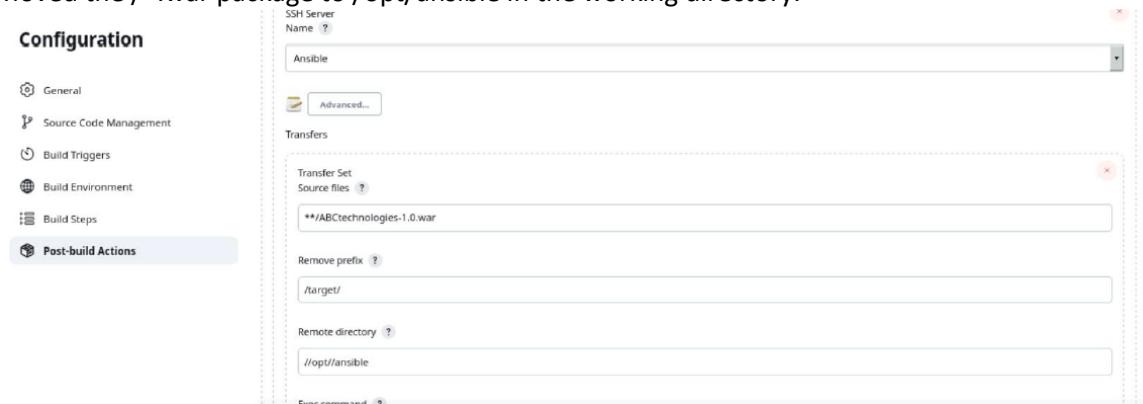
**Kubernetes:** Kubernetes is an open-source container orchestration system for automating software deployment, scaling, and management. Google originally designed Kubernetes, but the Cloud Native Computing Foundation now maintains the project.

on Master-node and added a playbook for creating & pushing docker images to the docker hub.



Used Package job as upstream to configure this job.

Moved the /\*.war package to /opt/ansible in the working directory.



Created the Dockerfile to create an Image with /\*.war package in it and run on the tomcat container.  
Create : path: cd /opt/ansible #: cat Dockerfile

```
FROM tomcat:9.0-alpine

ADD ./*.war /usr/local/tomcat/webapps/

EXPOSE 8080

CMD ["catalina.sh", "run"]
```

# Post Graduate Certification Program in DevOps

## Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V

Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>

Added Ansible Playbook to create and push docker Image to docker hub.

Create : path: cd /opt/ansible #: Cat [ABCtech.yml](#)

```
---
- hosts: ansible
  tasks:
    - name: creating docker image
      command: docker build -t tomy:v1 .
      args:
        chdir: /opt/ansible
    - name: creating tag on docker image
      command: docker tag newununtu:v1 kannan140494/tomy:v1
    - name: pushing on to docker hub
      command: docker push kannan140494/newununtu:v1
```

In Jenkins Job, I have given commands to execute this ansible playbook only after the Package job is executed successfully.

The screenshot shows the Jenkins job configuration interface. On the left, there's a sidebar with options like General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The Post-build Actions section is selected and expanded. It contains fields for '/target/' (Remote directory), '//opt/ansible' (value for Remote directory), and 'ansible-playbook /opt/ansible/ABCtech.yml' (value for Exec command). Below these fields is a note: 'All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables'. At the bottom of the section are 'Advanced...' and 'Add Transfer Set' buttons.

### Output:

```
SSH: EXEC: STDOUT/STDERR from command [ansible-playbook /opt/ansible/ABCtech.yml]
] ...
SSH: EXEC: connected
PLAY [ansible] ****
TASK [Gathering Facts] ****
[DEPRECATION WARNING]: Distribution ubuntu 18.04 on host 172.31.3.226 should
use /usr/bin/python3, but is using /usr/bin/python for backward compatibility
with prior Ansible releases. A future Ansible release will default to using the
discovered platform python for this host. See https://docs.ansible.com/ansible
/2.9/reference_appendices/interpreter_discovery.html for more information. This
feature will be removed in version 2.12. Deprecation warnings can be disabled
by setting deprecation_warnings=False in ansible.cfg.
ok: [172.31.3.226]

TASK [creating docker image] ****
changed: [172.31.3.226]

TASK [creating tag on docker image] ****
changed: [172.31.3.226]

TASK [pushing on to docker hub] ****
changed: [172.31.3.226]

PLAY RECAP ****
172.31.3.226 : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

SSH: EXEC: completed after 49.052 ms
SSH: Disconnecting configuration [Ansible] ...
SSH: Transferred 1 file(s)
Triggering a new build of ABC_Technologies_Package_deployment_on_Kubernetes_cluster
Finished: SUCCESS
```

Using Ansible, I have created the docker Image with /\*.war in it and pushed it to the docker hub.

Created Kubernetes service and deployment manifest files in /opt/k8s work directory.

Create : path: cd /opt/k8s #: Cat Deployment.yml

**Post Graduate Certification Program in DevOps**  
**Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V**  
**Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>**

---

```
kind: Deployment
apiVersion: apps/v1
metadata:
  name: abctechnologies-dep
spec:
  replicas: 2
  minReadySeconds: 45
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 2
  selector:
    matchLabels:
      app: abc-tech-app
  template:
    metadata:
      labels:
        app: abc-tech-app
    spec:
      containers:
        - image: kannan140494/newununtu
          name: app
---
kind: Service
apiVersion: v1
metadata:
  name: abc-tech-service
spec:
  type: NodePort
  selector:
    app: abc-tech-app
  ports:
    - port: 80 #cluster port
      targetPort: 8080 #container image port
```

Created ansible playbook to deploy Kubernetes resources

Execute ansible-playbook to deploy on K8's Cluster only if the upstream job is stable.

**Post Graduate Certification Program in DevOps**  
**Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V**  
**Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>**



**Configuration**

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions**

Remove prefix ?  
Remote directory ?  
Exec command ?  
ansible-playbook /opt/k8s/ABC-ansible-k8s.yml

All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)

Advanced...

[↑ Back to Project](#)

**Console Output**

Started by user ranjith  
Running as SYSTEM  
Building on the built-in node in workspace /var/lib/jenkins/workspace/ABC\_Technologies\_Package\_deployment\_on\_Kubernetes\_cluster  
SSH: Connecting from host [kmaster]  
SSH: Connecting with configuration [Ansible] ...  
SSH: EXEC: completed after 7,005 ms  
SSH: Disconnecting configuration [Ansible] ...  
SSH: Transferred 0 file(s)  
Finished: SUCCESS

[View as plain text](#)

[Edit Build Information](#)

[Delete build #9](#)

[Timings](#)

[Open Blue Ocean](#)

10.244.1.70:8080/ABCte x | 10.244.1.72:8080/ABCte x | 10.244.1.71:8080/ABCte x | +  
10.244.1.70:8080/ABCtechnologies-1.0/

**Welcome to ABC technologies**

**This is retail portal**

Add Product View Product

\*\*\*Kubernetes cluster is up and running with our deployed /\*.war file\*\*\*  
\*\*\*\*\*

### **Task 5:** Monitoring using Prometheus and Grafana

**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.

Post Graduate Certification Program in DevOps  
Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V  
Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>

---

1. Update the System Update the apt package list to prepare the system for further installations.

**\$ sudo apt update**

2. Download and Install Prometheus

Prometheus installation files are packaged as precompiled binaries. To download your preferred binaries, you can visit the official Prometheus download page.

If you decide to install a different version of Prometheus, please note the version numbers in the following examples when downloading and extracting the archives.

Download the Prometheus release package.

**\$ wget <https://github.com/prometheus/prometheus/releases/download/v2.37.0/prometheus-2.37.0.linux-amd64.tar.gz>**

Extract the downloaded archive.

**\$ tar xvf prometheus-2.37.0.linux-amd64.tar.gz**

Change directory to the extracted archive.

**\$ cd prometheus-2.37.0.linux-amd64**

Create the configuration file directory.

**\$ sudo mkdir -p /etc/Prometheus**

Create the data directory.

**\$ sudo mkdir -p /var/lib/Prometheus**

Move the binary files Prometheus and promtool to /usr/local/bin/.

**\$ sudo mv prometheus promtool /usr/local/bin/**

Move console files in the console directory and library files in the console\_libraries directory to /etc/prometheus/ directory.

**\$ sudo mv consoles/ console\_libraries/ /etc/prometheus/**

Move the template configuration file prometheus.yml to /etc/prometheus/ directory

**\$ sudo mv prometheus.yml /etc/prometheus/prometheus.yml**

Verify the installed version of Prometheus.

**\$ prometheus --version**

Verify the installed version of promtool.

**\$ promtool --version**

3. Configure the System Group and User

Post Graduate Certification Program in DevOps  
Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V  
Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>

---

Create a Prometheus group.

```
$ sudo groupadd --system Prometheus
```

Create a user Prometheus and assign it to the created Prometheus group.

```
$ sudo useradd -s /sbin/nologin --system -g prometheus Prometheus
```

Set the ownership of Prometheus files and data directories to the Prometheus group and user.

```
$ sudo chown -R prometheus:prometheus /etc/prometheus/ /var/lib/prometheus/
```

```
$ sudo chmod -R 775 /etc/prometheus/ /var/lib/prometheus/
```

#### 4. Configure Systemd Service

Create a system service file for Prometheus to start at boot time.

```
$ sudo vi /etc/systemd/system/prometheus.service
```

Add the following lines to the file and save it:

**[Unit]**

**Description=Prometheus**

**Wants=network-online.target**

**After=network-online.target**

**[Service]**

**User=prometheus**

**Group=prometheus**

**Restart=always**

**Type=simple**

**ExecStart=/usr/local/bin/prometheus \**

**--config.file=/etc/prometheus/prometheus.yml \**

**--storage.tsdb.path=/var/lib/prometheus/ \**

**--web.console.templates=/etc/prometheus/consoles \**

**--web.console.libraries=/etc/prometheus/console\_libraries \**

**--web.listen-address=0.0.0.0:9090**

**[Install]: WantedBy=multi-user.target**

Start the Prometheus service.

Post Graduate Certification Program in DevOps  
Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V  
Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>

---

**Graph:**

For example, enter the following expression to graph the per-second HTTP request rate returning status code 200 happening in the self-scraped Prometheus:

```
rate(promhttp_metric_handler_requests_total{code="200"}[1m])
```

You can experiment with the graph range parameters and other settings.

**MONITORING LINUX HOST METRICS WITH THE NODE EXPORTER:**

```
wget https://github.com/prometheus/node_exporter/releases/download/v1.3.1/node_exporter-1.3.1.linux-amd64.tar.gz
tar xvfz node_exporter-1.3.1.linux-amd64.tar.gz
cd node_exporter-1.3.1.linux-amd64
./node_exporter
```

**Install Grafana:**

```
sudo apt-get install -y apt-transport-https
sudo apt-get install -y software-properties-common wget
sudo wget -q -O /usr/share/keyrings/grafana.key https://apt.grafana.com/gpg.key
echo "deb [signed-by=/usr/share/keyrings/grafana.key] https://apt.grafana.com stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
sudo apt-get update
sudo apt-get install grafana
sudo systemctl start grafana-server
sudo systemctl enable grafana-server
sudo systemctl status grafana-server
```

The Grafana dashboard can be accessed from localhost:3000 in your browser

The default username and password are admin and admin. You can change it after the first login

# Post Graduate Certification Program in DevOps

Industry Grade Project 1: ABC Technologies; Submission By: Kannan. V

Project Remote Git Repository Link: <https://github.com/Kannan140494/newproject>

after completing all the steps I was getting Fail to start the Prometheus then I restarted the VM after the restart Prometheus is up and running

## OUTPUT:



## **OUTPUT:**



**Submission By:** Kannan. V

**Email ID:** Sree140494@gmail.com

\*\*\* End of the Project \*\*\*