

Karthik Pradeep Hegadi

**2KE20CS032**

### **Assignment 25**

*Understood. To follow the provided instructions and create the files/directory using the same name and case as provided in the task steps, please provide me with the specific names and case instructions for the files/directory you want to create.*

---

## **Jenkins**

---

### **Assignment 1 : Jenkins assignment 1**

**Note: I have added only main steps.**

---

#### **1. Install jenkins**

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.433s)
sudo yum install jenkins -y

Last metadata expiration check: 0:01:17 ago on Tue 10 Oct 2023 10:46:45 PM IST.
Package jenkins-2.414.2-1.1.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (1.083s)
sudo systemctl status jenkins

● jenkins.service - Jenkins Continuous Integration Server
    Loaded: loaded (/usr/lib/systemd/system/jenkins.service; disabled; preset:
    Active: inactive (dead)

Oct 10 22:47:01 localhost.localdomain jenkins[16697]: 2023-10-10 17:17:01.089+00
```

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (4.064s)
sudo systemctl start jenkins

lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.192s)
sudo systemctl enable jenkins

Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /u
```

```
lousy@localhost.localdomain ~ git:(HEAD) ±11
sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: d
  Active: active (running) since Tue 2023-10-10 22:48:41 IST; 40s ago
    Main PID: 29729 (java)
      Tasks: 44 (limit: 10685)
     Memory: 333.9M
        CPU: 6.908s
       CGroup: /system.slice/jenkins.service
             └─29729 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java
```

## 2. Create new job and run it

The screenshot shows the Jenkins Project the\_test dashboard. At the top, there's a navigation bar with the Jenkins logo and the project name 'Project the\_test'. Below the navigation bar, there's a sidebar with various project management options like Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. The main content area is titled 'Project the\_test' and displays a 'Build History' section. It shows four builds: Last build (#1), Last stable build (#1), Last successful build (#1), and Last completed build (#1), all of which occurred 7 min 58 sec ago. There's also a 'trend' dropdown menu. On the right side, there's a 'Permalinks' section with links to the last four builds.

The screenshot shows the Jenkins Console Output for build #1. At the top, there's a navigation bar with the Jenkins logo and the project name 'the\_test'. Below the navigation bar, there's a sidebar with options like Status, Changes, Console Output (which is selected and highlighted in a grey box), View as plain text, Edit Build Information, and Delete build '#1'. The main content area is titled 'Console Output' and contains a large green circle with a white checkmark. To the right of the checkmark, the output of the build is displayed. The output shows the build was started by user 'lousy' and running as SYSTEM. It shows the command being run: [the\_test] \$ /bin/sh -xe /tmp/jenkins1841829317364151931.sh + echo 'Hello World!!!!'. The output ends with 'Hello World!!!!' and 'Finished: SUCCESS'.

Karthik Pradeep Hegadi

2KE20CS032

## Assignment 26

*Understood. To follow the provided instructions and create the files/directory using the same name and case as provided in the task steps, please provide me with the specific names and case instructions for the files/directory you want to create.*

### Jenkins -2

## Assignment 2 :Jenkins Maven Project

```
lusy@localhost.localdomain /etc/nginx/conf.d (3m 27.04s)
sudo yum update
[sudo] password for lusy:
Last metadata expiration check: 1:50:07 ago on Sat 14 Oct 2023 10:23:16 AM IST.
Dependencies resolved.
=====
 Package                                         Architecture
=====
 Installing:
  kernel                                         aarch64
  kernel-headers                                 aarch64
  kernel-tools                                   aarch64
  kernel-tools-headers                           aarch64
  kernel-tools-libelf                            aarch64
  kernel-tools-libelf-devel                     aarch64
  kernel-tools-libelf-devel-devel               aarch64
  kernel-tools-libelf-devel-devel-devel         aarch64
  kernel-tools-libelf-devel-devel-devel-devel   aarch64
  kernel-tools-libelf-devel-devel-devel-devel-devel aarch64
```

## 1. Download and install apache maven and set the environmental variables for apache-maven

```
lusy@localhost.localdomain /etc/nginx/conf.d (0.532s)
sudo yum install maven -y
Last metadata expiration check: 1:53:47 ago on Sat 14 Oct 2023 10:23:16 AM IST.
Package maven-1:3.6.3-15.el9.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

```
lusy@localhost.localdomain /etc/nginx/conf.d (0.188s)
mvn -version
Apache Maven 3.6.3 (Red Hat 3.6.3-15)
Maven home: /usr/share/maven
Java version: 11.0.18, vendor: Red Hat, Inc., runtime: /usr/lib/jvm/java-11-open
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "5.14.0-370.el9.aarch64", arch: "aarch64", family: "u
```

```

lousy@localhost.localdomain /etc/nginx/conf.d (44.073s)
vi ~/.bashrc

lousy@localhost.localdomain /etc/nginx/conf.d (0.07s)
sudo systemctl start jenkins

lousy@localhost.localdomain /etc/nginx/conf.d (1.035s)
sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
    Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: d
    Active: active (running) since Sat 2023-10-14 10:08:42 IST; 2h 9min ago
      Main PID: 712 (java)

```

## 2. Create new project ex: maven \_project

The screenshot shows the Jenkins dashboard at [10.211.55.20:8080](http://10.211.55.20:8080). The sidebar on the left includes links for 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'My Views'. The main area displays a table of recent builds:

| S | W | Name          | Last Success    |
|---|---|---------------|-----------------|
|   |   | maven_project | 2 min 25 sec #2 |
|   |   | the_test      | 3 days 13 hr #1 |

## 4. Select Freestyle project and proceed 'Next'

Source Code Management

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#)  
https://github.com/maraks-gradious/maven-test1

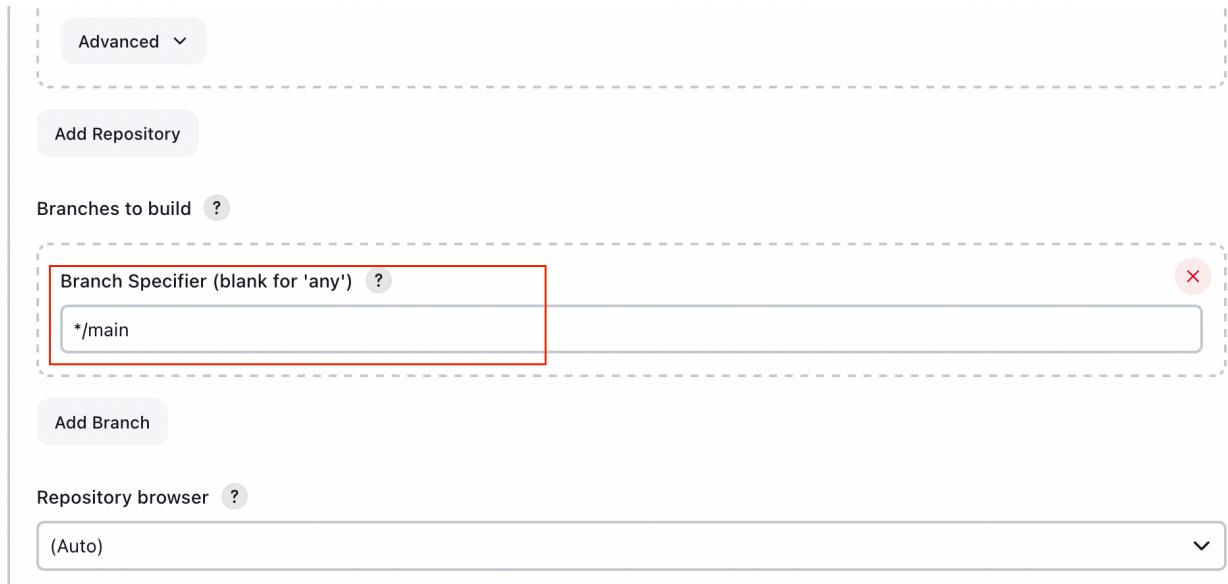
Credentials [?](#)  
- none -

Add [▼](#)

Select the source code management as git, provide the repository url as Below

<https://github.com/maraks-gradious/maven-test1>

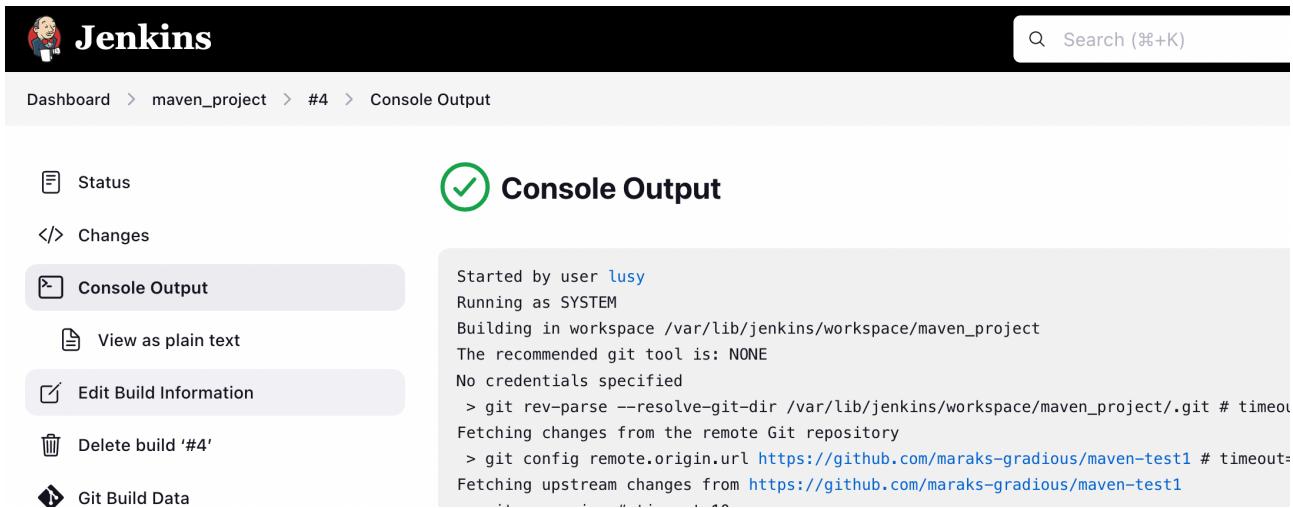
6. In the branches to build, modify master to main



7. Modify the build step as below



## 8. You can see the console output as below



The screenshot shows the Jenkins interface for a build named 'maven\_project' (Build #4). The left sidebar contains links for Status, Changes, Console Output (which is selected and highlighted in grey), View as plain text, Edit Build Information (also highlighted), Delete build '#4', and Git Build Data. The main content area is titled 'Console Output' with a green checkmark icon. It displays the build logs:

```
Started by user lusy
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/maven_project
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/maven_project/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/maraks-gradious/maven-test1 # timeout=10
Fetching upstream changes from https://github.com/maraks-gradious/maven-test1
... (truncated)
```

## 9. You can verify the output of your builds in your jenkins home directory which is in the path

*/var/lib/jenkins/workspace*

```
lusy@localhost.localdomain:/etc/nginx/conf.d (0.014s)
cd /var/lib/jenkins/workspace

lusy@localhost.localdomain /var/lib/jenkins/workspace (0.026s)
ls
maven_project  sona  SonarQube  the_test
```

Karthik Pradeep Hegadi

2KE20CS032

## Assignment 26

*Understood. To follow the provided instructions and create the files/directory using the same name and case as provided in the task steps, please provide me with the specific names and case instructions for the files/directory you want to create.*

Jenkins

---

---

## Assignment 2 : SonarQube - Docker installation

---

---

### 1. Edit the system configuration file vi /etc/sysctl.conf

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.02s)
cd /etc/
```

```
lusy@localhost.localdomain /etc (2m 3.22s)
sudo vi sysctl.conf
```

### 2. Insert the following lines at the end of this file

1. m.max\_map\_count=262144
2. fs.file-max=65536

```
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).i
vm.max_map_count=262144
fs.file-max=65536
~
```

### **3. Enable the system configuration.**

1.sysctl-p

```
lousy@localhost.localdomain /etc (0.056s)
sudo sysctl -p
vm.max_map_count = 262144
fs.file-max = 65536
```

### **4. Create a configuration file named 99-sonarqube.conf.**

vi /etc/security/limits.d/99-sonarqube.conf

```
lousy@localhost.localdomain /etc (25.941s)
sudo vi /etc/security/limits.d/99-sonarqube.conf
```

### **5. Paste the following lines in the vi editor**

root- nofile 65536

root- nproc 7610

```
#root soft nofile 65536
#root hard nofile 65536
root soft nproc 7610
root hard nproc 7610
```

~

### **6. Reboot the system**

```
lousy@localhost.localdomain /etc (0.073s)
sudo reboot
```

```
Connection to 10.211.55.20 closed by remote host.
Connection to 10.211.55.20 closed.
```

## 7. Start the docker service using systemctl start docker

(Note : you need to have docker installed in your system)

```
base ~ (15.404s)
ssh lusy@10.211.55.20

lusy@localhost.localdomain ~ git:(HEAD) ±11 (2.783s)
sudo systemctl start docker
[sudo] password for lusy:

lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.909s)
sudo systemctl status docker
● docker.service - Docker Application Container Engine
    Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset:
    Active: active (running) since Tue 2023-10-10 22:58:25 IST; 18s ago
      TriggeredBy: ● docker.socket
      Process: 1144 ExecStart=/bin/systemctl --no-pid --quiet --root /
      Main PID: 1144 (dockerd)
         CPU: 0.000 CPU(s) [idle]
          Tasks: 0 (limit: 12282)
          Memory: 0.0M
             CPU: 0.000 CPU(s) [idle]
```

## 8. Execute the command

```
docker pull sonarqube
```

```
lusy@localhost.localdomain ~ git:(HEAD) ±11
docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
6ea603f1df5e: Pull complete
5ba2aefbab33: Pull complete
869876930041: Pull complete
481239d57792: Pull complete
0707fa9330d3: Pull complete
e2a21872f107: Downloading [=====] 137.6MB/382MB
0b5f8e022df6: Download complete
```

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (5.259s)
docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:45c205d63a3341c88618528a0e0245620633f20c98832033712acf6f8536f1a4
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest
```

## 9. List the docker images

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.047s)
docker images
REPOSITORY           TAG      IMAGE ID      CREATED       SIZE
sonarqube            latest   be5948f93c5d  7 days ago   710MB
httpd_test           latest   834e577d6152  3 weeks ago  195MB
kartik404/mysql     arm     e248da1c6d57  3 weeks ago  507MB
kartik404/mysql     first_dhub  e248da1c6d57  3 weeks ago  507MB
mysql               latest   e248da1c6d57  3 weeks ago  507MB
```

## 10. Create Docker volumes to store the SonarQube persistent data.

```
docker volume create sonarqube-conf  
docker volume create sonarqube-data  
docker volume create sonarqube-logs  
docker volume create sonarqube-extensions
```

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.04s)
```

```
docker volume create sonarqube-conf
```

```
sonarqube-conf
```

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.043s)
```

```
docker volume create sonarqube-data
```

```
sonarqube-data
```

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.04s)
```

```
docker volume create sonarqube-logs
```

```
sonarqube-logs
```

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.043s)
```

```
docker volume create sonarqube-extensions
```

```
sonarqube-extensions
```

## 11. Verify the persistent data directories

```
docker volume inspect sonarqube-conf  
docker volume inspect sonarqube-data  
docker volume inspect sonarqube-logs  
docker volume inspect sonarqube-extensions
```

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.037s)
docker volume inspect sonarqube-conf
[{"Created": "2023-10-10T23:36:29+05:30", "Driver": "local", "Labels": null, "Mountpoint": "/var/lib/docker/volumes/sonarqube-conf/_data", "Name": "sonarqube-conf", "Options": null, "Scope": "local"}]
```

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.036s)
docker volume inspect sonarqube-data
[{"Created": "2023-10-10T23:36:35+05:30", "Driver": "local", "Labels": null, "Mountpoint": "/var/lib/docker/volumes/sonarqube-data/_data", "Name": "sonarqube-data", "Options": null, "Scope": "local"}]
```

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.036s)
docker volume inspect sonarqube-logs
[{"Created": "2023-10-10T23:36:41+05:30", "Driver": "local", "Labels": null, "Mountpoint": "/var/lib/docker/volumes/sonarqube-logs/_data", "Name": "sonarqube-logs", "Options": null, "Scope": "local"}]
```

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.039s)
docker volume inspect sonarqube-extensions
[{"Created": "2023-10-10T23:36:46+05:30", "Driver": "local", "Labels": null, "Mountpoint": "/var/lib/docker/volumes/sonarqube-extensions/_data", "Name": "sonarqube-extensions", "Options": null, "Scope": "local"}]
```

## 12. create symbolic links to an easier access location.

mkdir/sonargube

In -s /var/lib/docker/volumes/sonarqube-conf/\_data /sonarqube/conf

In -s /var/lib/docker/volumes/sonarqube-data/\_data/sonarqube/data

In -s /var/lib/docker/volumes/sonarqube-logs/\_data/sonarqube/logs

In -s /var/lib/docker/volumes/sonarqube-extensions/ data

/sonarqube/extensions

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (4.16s)
```

```
sudo mkdir /sonarqube
```

```
[sudo] password for lusy:
```

(*verify*)

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.023s)
```

```
cd /sonarqube
```

```
lusy@localhost.localdomain /sonarqube (0.02s)
```

```
cd
```

(NEXT PAGE)

### **13. Start a SonarQube container with persistent data storage.**

```
docker run -d -name sonarqube -p 9000:9000 -p 9092:9092 -V  
sonarqube-conf:/opt/sonarqube/conf-vsonarqube-data:/opt/sonarqube/data-V  
sonarqube-logs:/opt/sonarqube/logs-v  
sonarqube-extensions:/opt/sonarqube/extensions sonarqube
```

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.056s)  
sudo ln -s /var/lib/docker/volumes/sonarqube-conf/_data /sonarqube/conf  
  
lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.057s)  
sudo ln -s /var/lib/docker/volumes/sonarqube-data/_data /sonarqube/data  
  
lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.059s)  
sudo ln -s /var/lib/docker/volumes/sonarqube-logs/_data /sonarqube/logs  
  
lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.058s)  
sudo ln -s /var/lib/docker/volumes/sonarqube-extensions/_data /sonarqube/extensions
```

### **14. Open your browser by the url IP address of you server plus:9000**

```
lusy@localhost.localdomain ~ git:(HEAD) ±11 (0.392s)  
docker run -d --name sonarqube -p 9000:9000 -p 9092:9092 \  
-v sonarqube-conf:/opt/sonarqube/conf \  
-v sonarqube-data:/opt/sonarqube/data \  
-v sonarqube-logs:/opt/sonarqube/logs \  
-v sonarqube-extensions:/opt/sonarqube/extensions \  
sonarqube  
543d21224c962ea2cd8551b2ea201a3671402510c19203301c1741dacf1639dd
```

### **15. The SonarQube dashboard will be presented.**

### **16. Click on the Login button and use the Sonarqube default username and**

**password.**

- Default Username: admin
- Default Password: admin

Log in to SonarQube

admin

•••••

Log in Cancel

the\_test #1 Console [Jenkins]

How do you want to create your project?

Getting Started

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More

## How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)?  
Create your project from your favorite DevOps platform.

First, you need to set up a DevOps platform configuration.

|                             |                       |
|-----------------------------|-----------------------|
| Import from Azure DevOps    | <a href="#">Setup</a> |
| Import from Bitbucket Cloud | <a href="#">Setup</a> |
| Import from GitHub          | <a href="#">Setup</a> |
| Import from GitLab          | <a href="#">Setup</a> |

Are you just testing or have an advanced use-case? Create a project manually.

Karthik Pradeep Hegadi

**2KE20CS032**

### **Assignment 27**

*Understood. To follow the provided instructions and create the files/directory using the same name and case as provided in the task steps, please provide me with the specific names and case instructions for the files/directory you want to create.*

---

**Jenkins -3**

---

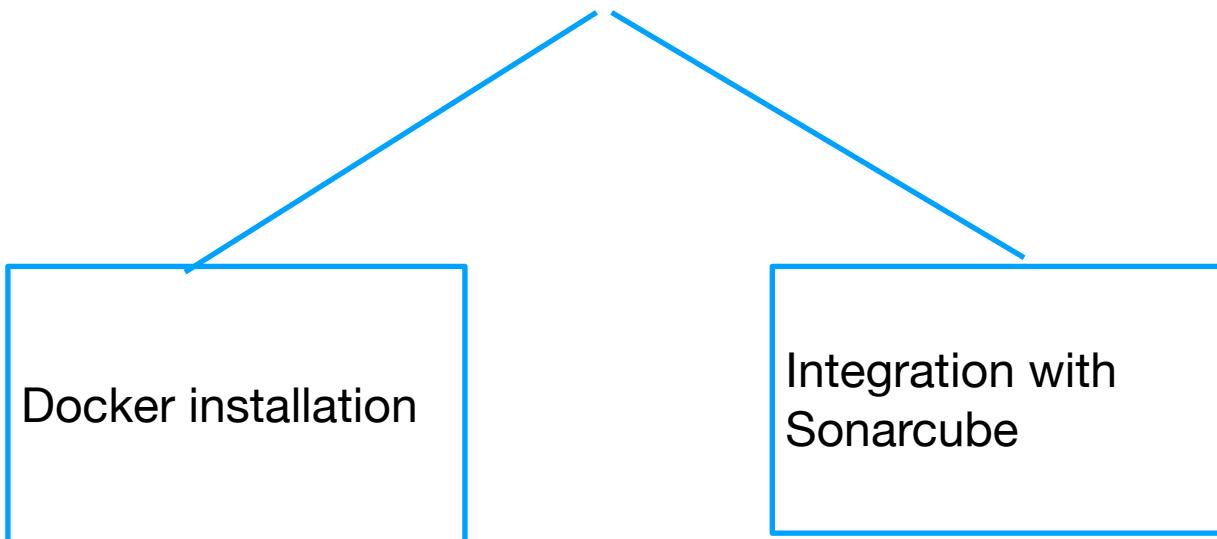
### **Assignment 3 :- Integrate SonarQube scanner into Pipeline**

---

**What is sonarqube?**

**SonarQube is an open-source platform developed by SonarSource for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells on 29 programming languages**

### **Integrate SonarQube scanner into Pipeline**



**Docker Installation**

## Integration with Sonarcube

1. In the jenkins, install the following plugins sonarqube scanner for jenkins and pipeline maven integration

The screenshot shows the Jenkins Plugins page. A search bar at the top contains the text "scanne". Below it, a list of installed and enabled plugins is shown:

- SonarQube Scanner for Jenkins 2.16.1**: Enabled. Description: This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality. Status: Enabled. Action buttons: a green hexagonal icon with a checkmark and a red hexagonal icon with a minus sign.
- Maven Integration plugin 3.23**: Enabled. Description: This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTs as well as the automated configuration of various Jenkins publishers such as Junit. Status: Enabled. Action buttons: a green hexagonal icon with a checkmark and a red hexagonal icon with a minus sign.
- Pipeline Maven Integration Plugin 1345.va\_0ef5530a\_5ca\_**: Enabled. Description: This plugin provides integration with Pipeline, configures maven environment to use within a pipeline job by calling sh mvn or bat mvn. The selected maven installation will be configured and prepended to the path. Status: Enabled. Action buttons: a green hexagonal icon with a checkmark and a red hexagonal icon with a minus sign.

3. Login to your sonarqube server -> Administration -> User -> Security -> tokens

4. Generate a token for jenkins and save it

The screenshot shows the SonarQube Security Tokens page for the "Administrator" user. The title is "Tokens of Administrator".

**Generate Tokens**

| Name             | Expires in |
|------------------|------------|
| Enter Token Name | 30 days    |

**Actions**

| Name    | Type | Project | Last use     | Created          | Expiration       | Actions                |
|---------|------|---------|--------------|------------------|------------------|------------------------|
| jenkins | User |         | < 1 hour ago | October 14, 2023 | January 12, 2024 | <a href="#">Revoke</a> |

[Done](#)

Administration

Configuration ▾ Security ▾ Projects ▾ System Marketplace

Users Create and administer individual users.

Search by login or name...  All users

| Name                  | SCM Accounts | Last connection | Last SonarLint connection | Groups | Tokens   | Actions |
|-----------------------|--------------|-----------------|---------------------------|--------|--|---------|
| A Administrator admin |              | < 1 hour ago    | Never                     | 2      | 1 <input type="button" value="Update tokens"/> |         |

1 of 1 shown

Jenkins

Dashboard > Manage Jenkins > Credentials

Credentials KEY Credentials

| T | P | Store ↓ | Domain   | ID            | Name          |
|---|---|---------|----------|---------------|---------------|
|   |   | System  | (global) | sonarcube_key | sonarcube_key |

Stores scoped to Jenkins

## 5. Navigate to Manage jenkins -> configure system ->sonarqube servers

Dashboard > Manage Jenkins > System > path

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables

Environment variables

SonarQube installations

List of SonarQube installations

| Name  |
|-------|
| sonar |

Server URL

Default is http://localhost:9000

http://10.211.55.20:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

sonarcube\_key

Add Advanced

The same name which we have mentioned while creating KEY

**In above step Provide the name sonar, your sonarqube server url and port, in the dropdown select**

**the secret text that is generated from your server**

**(Note : You need to add sonarqube credentials in Manage jenkins -> credentials -> system -> Global credentials. Select kind and copy sonarqube secret key, provide any id and description and add it)**

## 7. Now create a pipeline using the script placed in the link

Enter an item name

SonarQube\_pipeline\_script|  
» A job already exists with the name 'SonarQube\_pipeline\_script'

**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.

Dashboard > SonarQube\_pipeline\_script > Configuration path

**Configure** **Pipeline**

General Advanced Project Options Pipeline

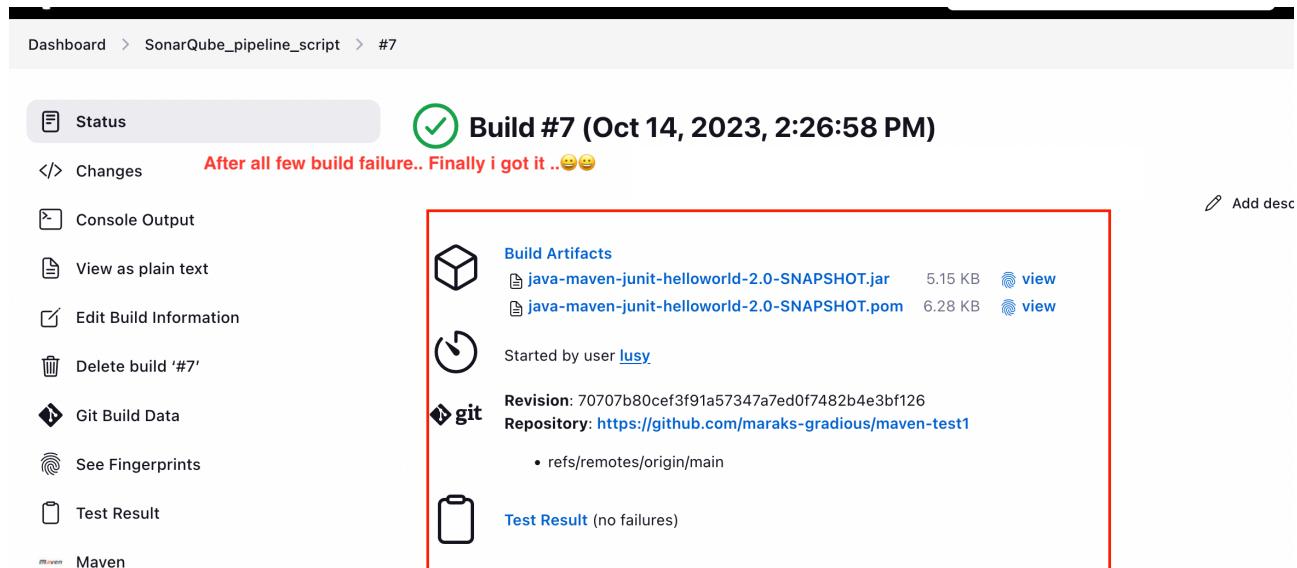
Definition Pipeline script

Script ?

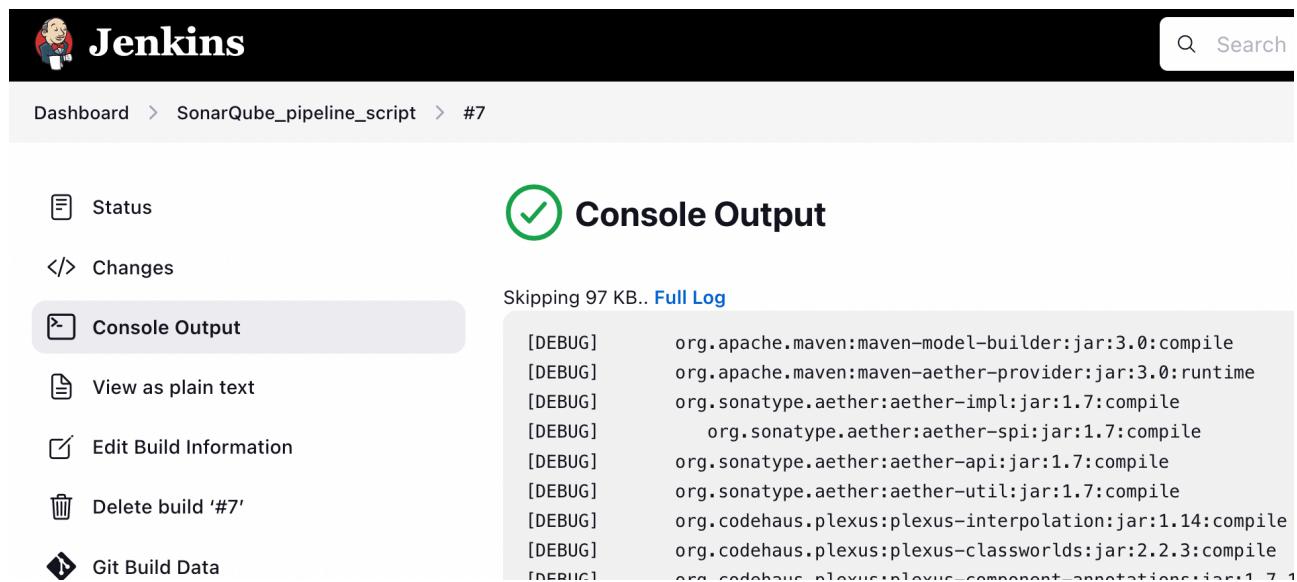
```
1 * pipeline {
2   agent any
3   stages{
4     stage('Source') {
5       steps{
6         git branch: 'main', url: 'https://github.com/maraks-gradious/maven-test1'
7       }
8     }
9     stage('Build') {
10    steps {
11      withSonarQubeEnv('sonar'){
12        // Optionally use a Maven environment you've configured already
13        withMaven{
14          sh "mvn clean verify sonar:sonar -X"
15        }
16      }
17    }
18  }
19}
```

**note: I have -X at last line**

## 8. Build the pipeline and verify the build status



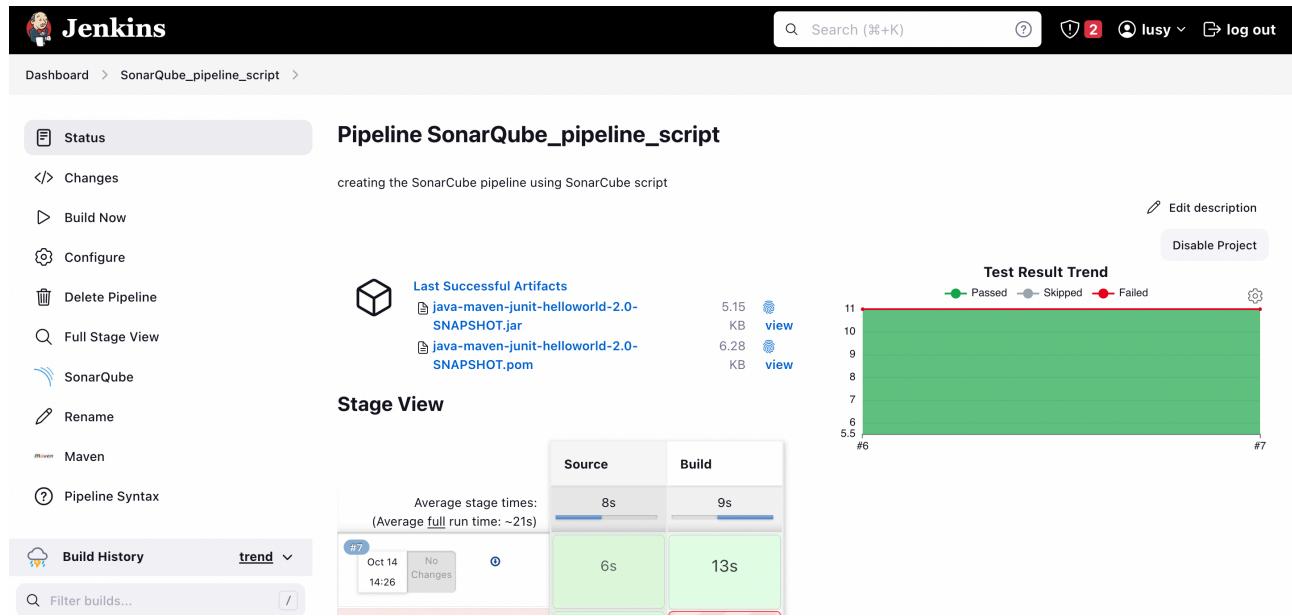
The screenshot shows the SonarQube Pipeline interface. At the top, it says "Build #7 (Oct 14, 2023, 2:26:58 PM)". Below that, a message reads "After all few build failure.. Finally i got it ..😊😊". A red box highlights the "Build Artifacts" section, which lists two files: "java-maven-junit-helloworld-2.0-SNAPSHOT.jar" (5.15 KB) and "java-maven-junit-helloworld-2.0-SNAPSHOT.pom" (6.28 KB). It also shows the revision (70707b80cef3f91a57347a7ed0f7482b4e3bf126) and repository (<https://github.com/marak-s-gradius/maven-test>). The "Test Result" section indicates no failures.



The screenshot shows the Jenkins interface. At the top, it says "Console Output". Below that, a message reads "Skipping 97 KB.. Full Log". A red box highlights the log output, which shows Maven dependency resolution for various org.apache.maven modules and components.

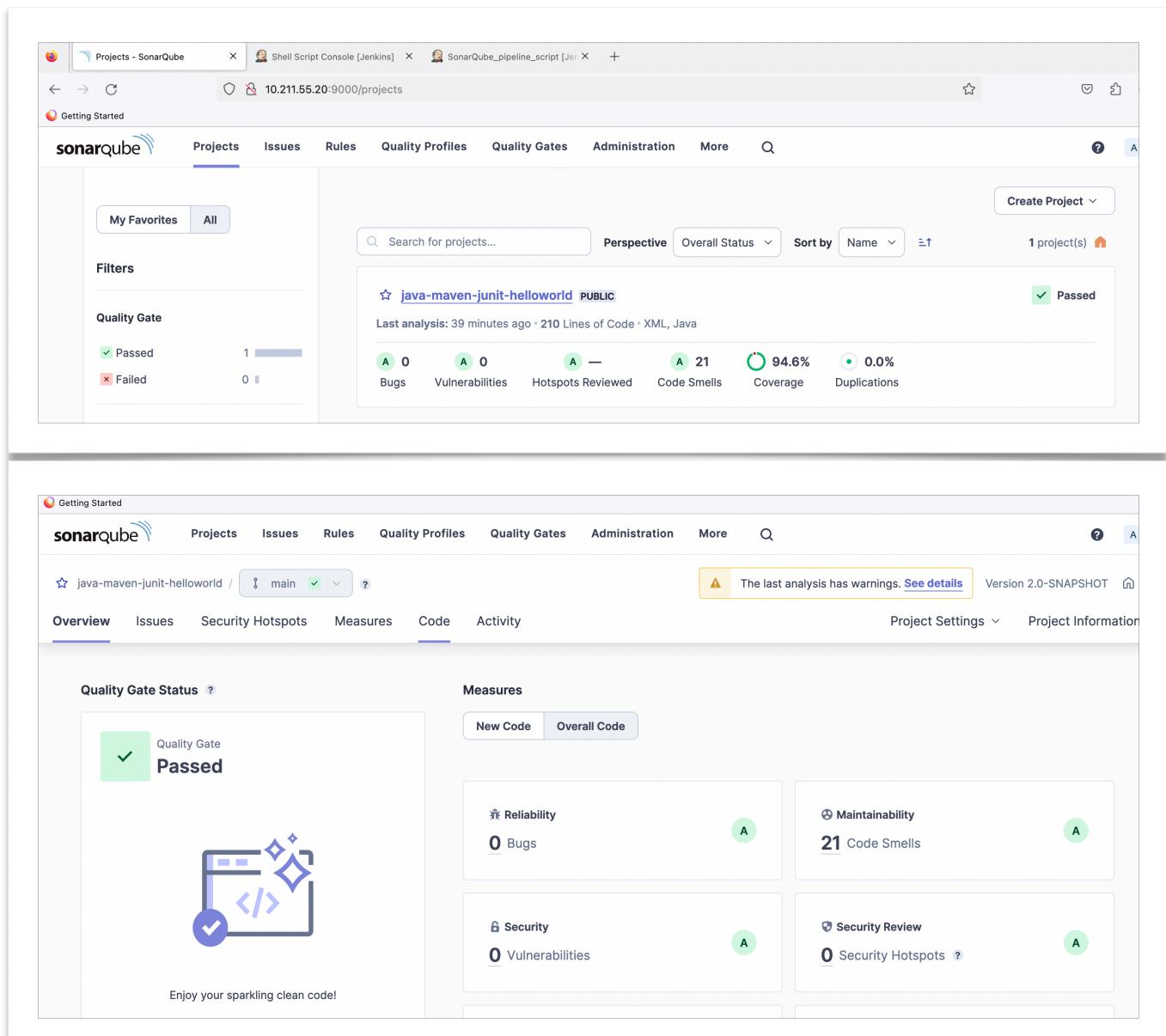
```
[DEBUG] org.apache.maven:maven-model-builder:jar:3.0:compile
[DEBUG] org.apache.maven:maven-aether-provider:jar:3.0:runtime
[DEBUG] org.sonatype.aether:aether-impl:jar:1.7:compile
[DEBUG] org.sonatype.aether:aether-spi:jar:1.7:compile
[DEBUG] org.sonatype.aether:aether-api:jar:1.7:compile
[DEBUG] org.sonatype.aether:aether-util:jar:1.7:compile
[DEBUG] org.codehaus.plexus:plexus-interpolation:jar:1.14:compile
[DEBUG] org.codehaus.plexus:plexus-classworlds:jar:2.2.3:compile
[DEBUG] org.codehaus.plexus:plexus-component-annotations:jar:1.7.1
```

## 9. Now navigate to sonarqube server to check the code quality results



The screenshot shows the Jenkins Pipeline SonarQube pipeline script dashboard. On the left, there's a sidebar with various project management and pipeline-related links. The main area is titled "Pipeline SonarQube\_pipeline\_script". It displays the status of the pipeline, which is currently "creating the SonarCube pipeline using SonarCube script". Below this, there's a section for "Last Successful Artifacts" showing two Java Maven JAR files. To the right is a "Test Result Trend" chart showing a single green bar from index 6 to 7, indicating all tests passed. Under "Stage View", it shows the average stage times for Source (8s) and Build (9s), with a total average full run time of approximately 21 seconds. A "Build History" section shows a single build (#7) from Oct 14 at 14:26 with "No Changes".

## 10. You can click your project and see the code test results,



The screenshot shows the SonarQube interface for the "java-maven-junit-helloworld" project. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. The main content area shows the project details: "Last analysis: 39 minutes ago · 210 Lines of Code · XML, Java". It displays quality gate status as "Passed" with a green checkmark. Below this, there are metrics for Bugs (0), Vulnerabilities (0), Hotspots Reviewed (1), Code Smells (21), Coverage (94.6%), and Duplications (0.0%). Further down, the "Overview" tab is selected, showing the "Quality Gate Status" as "Passed" with a green checkmark. The "Measures" section includes four cards: Reliability (0 Bugs), Maintainability (21 Code Smells), Security (0 Vulnerabilities), and Security Review (0 Security Hotspots). A message at the bottom says "Enjoy your sparkling clean code!".

Karthik Pradeep Hegadi

2KE20CS032

## Assignment 28

*Understood. To follow the provided instructions and create the files/directory using the same name and case as provided in the task steps, please provide me with the specific names and case instructions for the files/directory you want to create.*

### Jenkins -4

## Assignment 3 :- Build a war file Deploy into the tomcat container using pipeline script

### 1. Use this repo link for pipeline project

[https://github.com/maheshgradious/java\\_project](https://github.com/maheshgradious/java_project)

Fork the project

The screenshot shows a GitHub repository page for 'java\_project'. The repository is public and was forked from 'maheshgradious/java\_project'. It has 1 branch and 0 tags. The main branch is 2 commits ahead of the upstream. The commit history shows a recent update to the Dockerfile by 'kartikhegadi' (bf9122e) 3 days ago, along with other files like dist, src/main/webapp, README.md, and pom.xml. A file named 'Dockerfile' is highlighted with a red underline.

| File            | Description          | Time Ago      |
|-----------------|----------------------|---------------|
| Dockerfile      | Update Dockerfile    | 3 days ago    |
| dist            | Add files via upload | 10 months ago |
| src/main/webapp | Add files via upload | 10 months ago |
| README.md       | Add files via upload | 10 months ago |
| pom.xml         | Add files via upload | 10 months ago |

## Create Docker file

A screenshot of a GitHub repository page titled "java\_project / Dockerfile". The page shows a commit by "kartikhegadi" with the message "Update Dockerfile". Below the commit, there are tabs for "Code", "Blame", and "3 lines (3 loc) · 104 Bytes". A note says "Code 55% faster with GitHub Copilot". The Dockerfile content is:

```
1 FROM tomcat
2 COPY target/hello-world-war-1.0.0.war /usr/local/tomcat/webapps/
3 CMD ["catalina.sh", "run"]
```

## Use project for pipeline (choose pipeline project)

A screenshot of a Jenkins pipeline configuration page. The title bar says "Jenkins". The navigation bar shows "Dashboard > Docker-tomcat > Configuration". The main area has two tabs: "Configure" and "General". The "General" tab is selected. It includes a "Description" field containing the text "Assignment 4 build a war file using docker and deploy in tomcat." and a "Plain text" link.

## 2. Build a warfile using maven

## 3. Create a docker image of tomcat container and copy the .war build into it

## 4. Run the image as a container in docker(next page)

A screenshot of the Jenkins Pipeline script editor. The title bar says "Pipeline". The main area shows a "Script" section with the following Groovy code:

```
1 pipeline {
2     agent any
3     stages {
4         stage ('source') {
5             steps {
6                 git branch:'main', url:'https://github.com/kartikhegadi/java_project.git'
7             }
8         }
9         stage('Build') {
10            steps {
11                sh 'mvn clean verify'
12            }
13        }
14    }
15    stage ('Docker_build') {
```

Pipeline script

```
Script ?  
11 -  
12     steps {  
13         sh 'mvn clean verify'  
14     }  
15     }  
16     stage ('Docker_build') {  
17         steps {  
18             sh 'docker build . -t tomcat_imagee'  
19         }  
20     }  
21     stage ('Running_container') {  
22         steps {  
23             sh 'docker run -d -p 9090:8080 --name tomcat_containere tomcat_imagee'  
24         }  
25     }  
26 }
```

Full script ::

```
pipeline {  
    agent any  
    stages {  
        stage ('source') {  
            steps {  
                git branch:'main', url:'https://github.com/kartikhegadi/  
java_project.git'  
            }  
        }  
        stage('Build') {  
            steps {  
                sh 'mvn clean verify'  
            }  
        }  
        stage ('Docker_build') {  
            steps {  
                sh 'docker build . -t tomcat_imagee'  
            }  
        }  
    }  
}
```

```

stage ('Running_container') {
    steps {
        sh 'docker run -d -p 9090:8080 --name tomcat_container \
tomcat_imagee'
    }
}
}
}
}

```

## Build it

**Pipeline Docker-tomcat**

- </> Changes assignment 4
- ▷ Build Now
- ⚙ Configure
- trash Delete Pipeline
- 🔍 Full Stage View
- edit Rename
- info Pipeline Syntax

**Stage View**

| source                    | Build | Docker_build | Running_container |
|---------------------------|-------|--------------|-------------------|
| 6s                        | 1s    | 2s           | 157ms             |
| #4 Oct 28 19:10 2 commits | 6s    | 1s           | 4s                |
|                           | 287ms |              |                   |

Average stage times:  
(Average full run time: ~13s)

Build History trend ▾

## Verify

Dashboard > Docker-tomcat > #4

**Status** **Build #4 (Oct 28, 2023, 7:10:17 PM)**

- </> Changes
- [-] Console Output
- edit Edit Build Information
- trash Delete build '#4'
- git Git Build Data
- restart Restart from Stage
- replay Replay
- grid Pipeline Steps

**</>** Changes

1. Update Dockerfile ([details](#) / [githubweb](#))
2. Update Dockerfile ([details](#) / [githubweb](#))

Started by user [lousy](#)

Revision: [7cd51da532e01f55ff1e96c6c47be7a59c10a8f8](#)  
Repository: [https://github.com/kartikhegadi/java\\_project.git](https://github.com/kartikhegadi/java_project.git)

- refs/remotes/origin/main

```

#5 DONE 0.0s

#6 [2/2] COPY target/hello-world-war-1.0.0.war /usr/local/tomcat/webapps/
#6 DONE 0.0s

#7 exporting to image
#7 exporting layers done
#7 writing image sha256:93cc46f487c650a72e1848abac412a61784ccc0350679d33a661ee67896de862 done
#7 naming to docker.io/library/tomcat_imagee done
#7 DONE 0.0s
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Running_container)
[Pipeline] sh
+ docker run -d -p 9090:8080 --name tomcat_container tomcat_imagee
4f5da7b4146ab7776ab0774a4693998a69423486b618e67491d21dfa136b40cb
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

## Verify container

```

lusi@localhost.localdomain ~ git:(a57470d) ±14 (0.059s)
sudo docker ps
CONTAINER ID   IMAGE      COMMAND       CREATED      STATUS      PORTS          NAMES
4f5da7b4146a   tomcat_imagee   "catalina.sh run"   7 minutes ago   Up 7 minutes   0.0.0.0:9090->8080/tcp, :::9090->8080/tcp   tomcat_containere

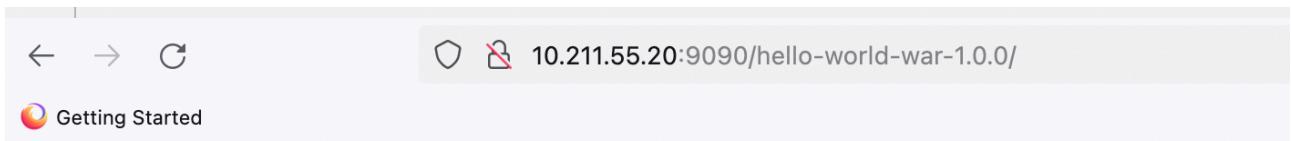
```

## 5. Access the tomcat server using the url to display the web page

```

lusi@localhost.localdomain ~/java_project/target git:(main) ±1 (0.06s)
curl 10.211.55.20:9090/hello-world-war-1.0.0/
<html>
<head>
<title>Hello World!</title>
</head>
<body>
    <h1>Hello World!</h1>
    <p>
        It is now
        Sat Oct 28 14:17:05 UTC 2023</p>
    <p>
        You are coming from
        10.211.55.20</p>
</body>

```



It is now Sat Oct 28 14:17:23 UTC 2023

You are coming from 10.211.55.2

Karthik Pradeep Hegadi

**2KE20CS032**

### **Assignment 29**

*Understood. To follow the provided instructions and create the files/directory using the same name and case as provided in the task steps, please provide me with the specific names and case instructions for the files/directory you want to create.*

### **Jenkins -5**

---

---

### **Assignment - 5 - create a simple php project pipeline /deploy to any web server & trigger the job using command line remotely**

---

#### **METHOD\_1**

**(USE THIS REPO LINK FOR php project ->**

<https://github.com/maheshgradious/php.git>

**U should have PHP server i.e httpd**

```
lousy@localhost.localdomain ~ git:(a57470d) ±14 (19.806s) $_
sudo yum install tomcat
Last metadata expiration check: 3:29:58 ago on Sat 28 Oct 2023 03:53:35 PM IST.
Package tomcat-1:9.0.62-16.el9.noarch is already installed.
Dependencies resolved.
=====
Package           Architecture   Version        Repository
=====
Upgrading:
tomcat           noarch        1:9.0.62-37.el9    appstream
tomcat-el-3.0-api noarch        1:9.0.62-37.el9    appstream
tomcat-jsp-2.3-api noarch        1:9.0.62-37.el9    appstream
tomcat-lib        noarch        1:9.0.62-37.el9    appstream
tomcat-servlet-4.0-api noarch        1:9.0.62-37.el9    appstream
Transaction Summary
=====
Upgrade 5 Packages

Total download size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
```

```
[sudo] password for lousy:
Sorry, try again.
[sudo] password for lousy:
Last metadata expiration check: 0:34:51 ago on Sat 28 Oct 2023 07:27:45 PM IST.
Package php-8.0.30-1.el9.aarch64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

```
lousy@localhost.localdomain /opt
```

## Deploy php code in apache or nginx web server) (use APACHE)

1. To run the job remotely, on creating a new project enable the following option

[V] Trigger builds remotely (e.g., from scripts)?

### Authentication Token

Use the following URL to trigger build remotely: JENKINS\_URL/job/test/build?token=TOKEN\_NAME or /buildWithParameters?token= TOKEN\_NAME

Optionally append &cause=Cause+Text to provide text that will be included in the recorded build cause.

2. Input your Auth token in this field, copy the url below

3. Save the project

The screenshot shows the Jenkins General configuration page for a project named "php\_deployment". The "General" tab is selected. In the "Description" section, there is a plain text preview of the assignment description: "Assignment - 5 - create a simple php project pipeline /deploy to any web server & trigger the job using command line remotely". Below this, under "Post-build Actions", four checkboxes are listed: "Discard old builds", "GitHub project", "This project is parameterized", and "Throttle builds". At the bottom of the page, the "Branches to build" section is shown, with a single branch specifier "/main" entered. There is also an "Add Branch" button and a "Repository browser" link.

Dashboard > php\_deployment > Configuration

**Configure**

**General**

Description

Assignment - 5 - create a simple php project pipeline /deploy to any web server & trigger the job using command line remotely

Plain text [Preview](#)

Post-build Actions

Discard old builds ?

GitHub project

This project is parameterized ?

Throttle builds ?

Branches to build ?

Branch Specifier (blank for 'any') ?

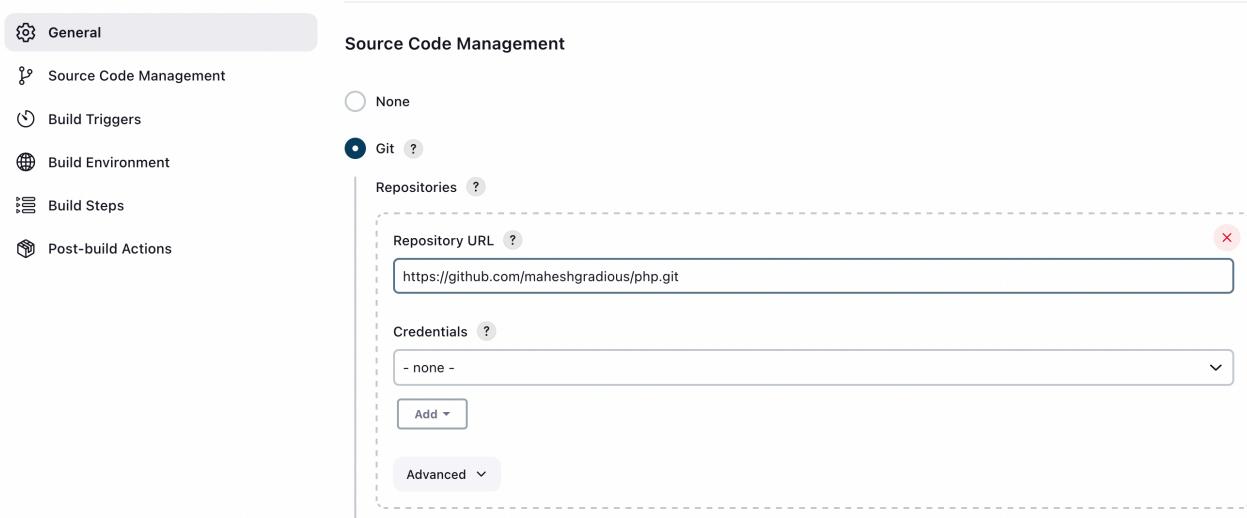
\*/main

Add Branch

Repository browser ?

#### 4. From your linux command prompt, run the following

**curl -J -u user\_name your\_copied url**



**(Note : provide your user\_name, make sure you gave the TOKEN\_NAME)**

#### API Token

This screenshot shows the 'API Token' section of the Jenkins configuration. It lists a single token named 'lousy\_token' which was created 0 days ago and has been used 16 times. A red trash can icon is next to the token name. At the bottom, there's a button labeled 'Add new Token' and a red link at the bottom that says 'Create token here use in link to auto trigger'.

## Build Triggers

- Trigger builds remotely (e.g., from scripts) [?](#)

Authentication Token

lousy@123

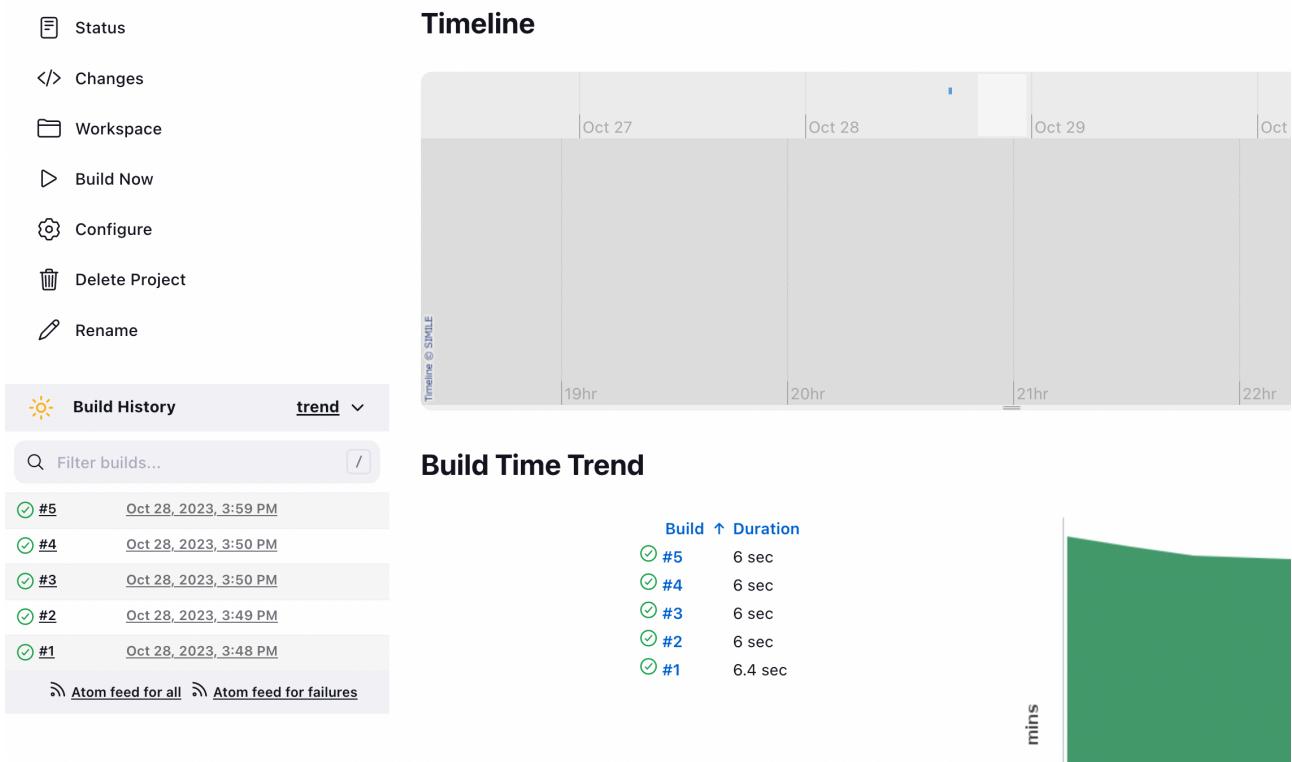
Use the following URL to trigger build remotely: JENKINS\_URL/job/php\_deployment/build?token=TOKEN\_NAME or /buildWithParameters?token=TOKEN\_NAME

Optionally append &cause=Cause+Text to provide text that will be included in the recorded build cause.

- Build after other projects are built [?](#)
- Build periodically [?](#)
- GitHub hook trigger for GITScm polling [?](#)
- Poll SCM [?](#)

```
lousy@localhost.localdomain ~ git:(a57470d) ±18 (0.055s)
curl -u lousy:1196266ea5651bdd30d5bf2da90578a751 -X POST http://10.211.55.20:8080/job/deploy_php/build?token=lousy@123
<html>
<head>
```

Dashboard > php\_deployment > Build Time Trend



Karthik Pradeep Hegadi

2KE20CS032

### Assignment 29

*Understood. To follow the provided instructions and create the files/directory using the same name and case as provided in the task steps, please provide me with the specific names and case instructions for the files/directory you want to create.*

### Jenkins -5

---

---

### Assignment - 5 - create a simple php project pipeline /deploy to any web server & trigger the job using command line remotely

---

---

### METHOD \_2

```
lusy@localhost.localdomain /var/www/html (0.073s)
php -v

PHP 8.0.30 (cli) (built: Aug 3 2023 17:13:08) ( NTS gcc aarch64 )
Copyright (c) The PHP Group
Zend Engine v4.0.30, Copyright (c) Zend Technologies
    with Zend OPcache v8.0.30, Copyright (c), by Zend Technologies
```

```
lusy@localhost.localdomain /var/www/html (3.818s)
sudo systemctl status httpd

[sudo] password for lusy:
● httpd.service - The Apache HTTP Server
    Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/httpd.service.d
              └─php-fpm.conf
    Active: active (running) since Sat 2023-10-28 18:53:30 IST; 2h 3min ago
      Docs: man:httpd.service(8)
   Main PID: 158425 (httpd)
     Status: "Total requests: 4; Idle/Busy workers 100/0;Requests/sec: 0.00054; Bytes served/se...
       Tasks: 213 (limit: 10684)
      Memory: 30.6M
        CPU: 7.171s
      CGroup: /system.slice/httpd.service
              ├─158425 /usr/sbin/httpd -DFOREGROUND
              ├─158426 /usr/sbin/httpd -DFOREGROUND
```



# Jenkins

Search (⌘+K)

Dashboard > php\_dep > Configuration

## Configure

### General

Enabled

#### General

Advanced Project Options

Pipeline

#### Description

Assignment - 5 - create a simple php project pipeline /deploy to any web server & trigger the job using command line remotely

Plain text [Preview](#)

Dashboard > php\_dep > Configuration

## Configure

Build whenever a SNAPSHOT dependency is built ?

GitHub hook trigger for GITScm polling ?

Poll SCM ?

Quiet period ?

Trigger builds remotely (e.g., from scripts) ?

#### Authentication Token

lusi@123

Use the following URL to trigger build remotely: JENKINS\_URL/job/php\_dep/build?token=TOKEN\_NAME or  
/buildWithParameters?token=TOKEN\_NAME

Optionally append &cause=Cause+Text to provide text that will be included in the recorded build cause.

Dashboard > php\_dep > Configuration

## Configure

### Definition

Pipeline script

#### General

Advanced Project Options

Pipeline

#### Script

```
1- pipeline {
2   agent any
3   stages {
4     stage('Source') {
5       steps {
6         git branch: 'main', url: 'https://github.com/maheshgradious/php.git'
7       }
8     }
9     stage('Copy') {
10    steps {
11      sh 'sudo cp /var/lib/jenkins/workspace/php_dep/index.php /var/www/html/'
12    }
13  }
14}
15}
```

 Jenkins

Dashboard > php\_dep > #4

Status  Build #4 (Oct 28, 2023, 8:49:45 PM) Keep this build

</> Changes  
Console Output  
Edit Build Information  
Delete build '#4'  
Git Build Data  
Restart from Stage

Started by remote host 10.211.55.20 with note: null

git Revision: c812e96cd7b1b3575494bb9ab108ee0fef3715d8 Repository: <https://github.com/maheshgradious/php.git>

refs/remotes/origin/main



lousy@localhost.localdomain /var/www/html (0.034s)

**curl 10.211.55.20/index.php**

```
<html>
<head>
  <title>PHP Test</title>
</head>
<body>
  <p>Hello World</p>
</body>
</html>
```

**lousy@localhost.localdomain /var/www/html**

The screenshot shows the Jenkins dashboard for a pipeline named "php\_dep". The left sidebar includes links for Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Rename, and Pipeline Syntax. The main area displays the "Build History" with five builds listed from #5 to #1. Build #5 is highlighted with a red arrow pointing to the terminal log on the right, which shows the command "curl -u lusy:11dbd888f088ee7060b737942b56789c5f http://10.211.55.20:8080/job/php\_dep/build?token=lusy@123" and its output.

```

lusy@localhost.localdomain /var/www/html (0.059s)
sudo cat index.php
<html>
<head>
<title>PHP Test</title>
</head>
<body>
<?php echo '<p>Hello World</p>'; ?>
</body>
</html>

lusy@localhost.localdomain /var/www/html (0.057s)
sudo chmod 777 /var/www/html/index.php

lusy@localhost.localdomain /var/www/html (0.021s)
cat index.php
<html>
<head>
<title>PHP Test</title>
</head>
<body>
<?php echo '<p>Hello World</p>'; ?>
</body>
</html>

lusy@localhost.localdomain /var/www/html (0.034s)
curl 10.211.55.20/index.php
<html>
<head>
<title>PHP Test</title>
</head>
<body>
<p>Hello World</p>
</body>
</html>

lusy@localhost.localdomain /var/www/html (0.059s)
curl -u lusy:11dbd888f088ee7060b737942b56789c5f http://10.211.55.20:8080/job/php_dep/build?token=lusy@123

lusy@localhost.localdomain /var/www/html

```

The screenshot shows a browser window displaying the Jenkins build output. The URL is "10.211.55.20/index.php". The page content is "Hello World". The right side of the screen shows the terminal logs corresponding to the build output.

```

lusy@localhost.localdomain /var/www/html (0.059s)
sudo cat index.php
<html>
<head>
<title>PHP Test</title>
</head>
<body>
<?php echo '<p>Hello World</p>'; ?>
</body>
</html>

lusy@localhost.localdomain /var/www/html (0.057s)
sudo chmod 777 /var/www/html/index.php

lusy@localhost.localdomain /var/www/html (0.021s)
cat index.php
<html>
<head>
<title>PHP Test</title>
</head>
<body>
<?php echo '<p>Hello World</p>'; ?>
</body>
</html>

lusy@localhost.localdomain /var/www/html (0.034s)
curl 10.211.55.20/index.php
<html>
<head>
<title>PHP Test</title>
</head>
<body>
<p>Hello World</p>
</body>
</html>

lusy@localhost.localdomain /var/www/html (0.059s)
curl -u lusy:11dbd888f088ee7060b737942b56789c5f http://10.211.55.20:8080/job/php_dep/build?token=lusy@123

```

Karthik Pradeep Hegadi

2KE20CS032

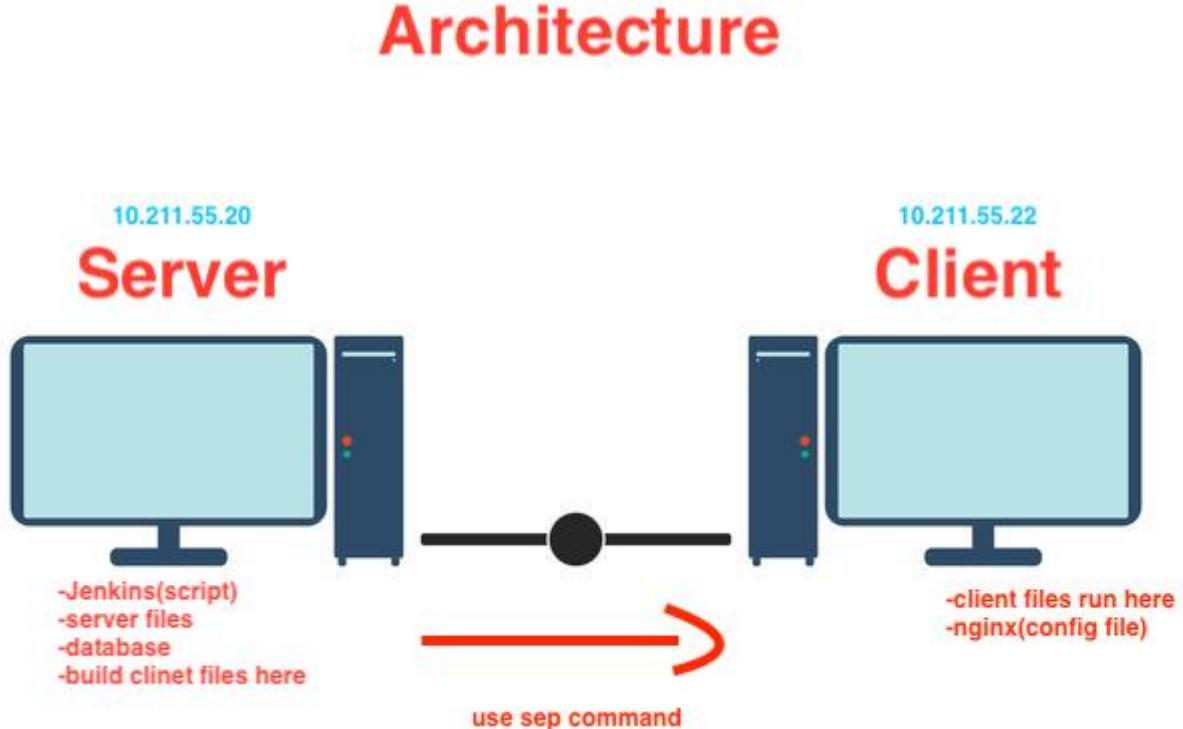
### Assignment 30

*Understood. To follow the provided instructions and create the files/directory using the same name and case as provided in the task steps, please provide me with the specific names and case instructions for the files/directory you want to create.*

Jenkins -6

### Assignment - Assignment - 6 - Build Jenkins Pipeline for 3 tier architecture involving Nodejs, ReactS & mysql

(You can refer the manual setup of server & client side, and you need to implement this in pipeline script)



## Server Building (manually)

Clone the server repo using given link

### 1) install nodejs

```
lusy@localhost.localdomain ~/nodejs_server git:(main) ±3 (10.666s)
sudo yum install nodejs
[sudo] password for lusy:
Sorry, try again.
[sudo] password for lusy:
Last metadata expiration check: 2:25:50 ago on Sun 29 Oct 2023 05:18:11 AM IST.
Package nodejs-1:16.20.1-2.el9.aarch64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

### 2) Install mysqld

```
lusy@localhost.localdomain ~/nodejs_server git:(main) ±3 (0.502s)
sudo yum install mysql
Last metadata expiration check: 2:26:40 ago on Sun 29 Oct 2023 05:18:11 AM IST.
Package mysql-8.0.32-1.el9.aarch64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

### 3) Create database

```
7 rows in set (0.10 sec)

mysql> create database doctor_appointment;
ERROR 1007 (HY000): Can't create database 'doctor_appointment'; database exists
mysql> █
```

```
Type 'help;' or '\h' for help. Type '\c' to clear

mysql> show databases;
+-----+
| Database      |
+-----+
| doctor_appointment |
| information_schema |
| mattermost       |
| mysql           |
| performance_schema |
| socka          |
| sys            |
+-----+
7 rows in set (0.10 sec)
```

#### 4) Import data base

```
lusy@localhost.localdomain ~/nodejs_server git:(main) ±3 (2.789s)
mysql -u root -p doctor_appointment <doctor_appointment.sql
Enter password:
```

#### 5) Verify it

```
mysql> show tables;
+-----+
| Tables_in_doctor_appointment |
+-----+
| patients
+-----+
1 row in set (0.02 sec)

mysql> select * from patients;
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | name      | age | gender | status | time   | date    | phone   | doctor  |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | John Doe  | 28  | Male   | Consult | 06:00 PM | 2 Feb 2021 | +91 987654321 | Dr. Ananth |
| 2  | Mukul Rao | 28  | Male   | Revisit | 06:00 PM | 2 Feb 2021 | +91 987654321 | Dr. Ananth |
| 3  | Neeraj Sharma | 28 | Male   | Consult | 06:00 PM | 2 Feb 2021 | +91 987654321 | Dr. Ananth |
| 11 | Manjunath Arun Naik | 22 | Male   | Revisit | 06:00 PM | 2 Feb 2021 | +91 932228949 | Dr. Ananth |
| 12 | Subbu Raja | 22  | Male   | Consult | 10:00 AM  | 10-12-2022 | 9012345678  | Dr. Raman  |
| 13 | Raj Kanna  | 31  | Male   | Consult | 13:00 PM  | 12-12-2022 | 8360285211 | Dr. Viswa |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

#### 6) change the credentials of app.js file

```
/ server to handle ajax requests to mockapi.io
const express = require('express');
const app = express();
const axios = require('axios');
const mysql = require('mysql2');

const pool = mysql.createPool({
  host: 'localhost',
  user: 'root',
  password: 'password',
  database: 'doctor_appointment'
});
```

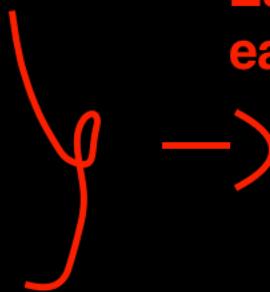
Edit accordingly

## 7) start it using npm command

```
lusy@localhost.localdomain ~/nodejs_server git:(main) ±3
npm start

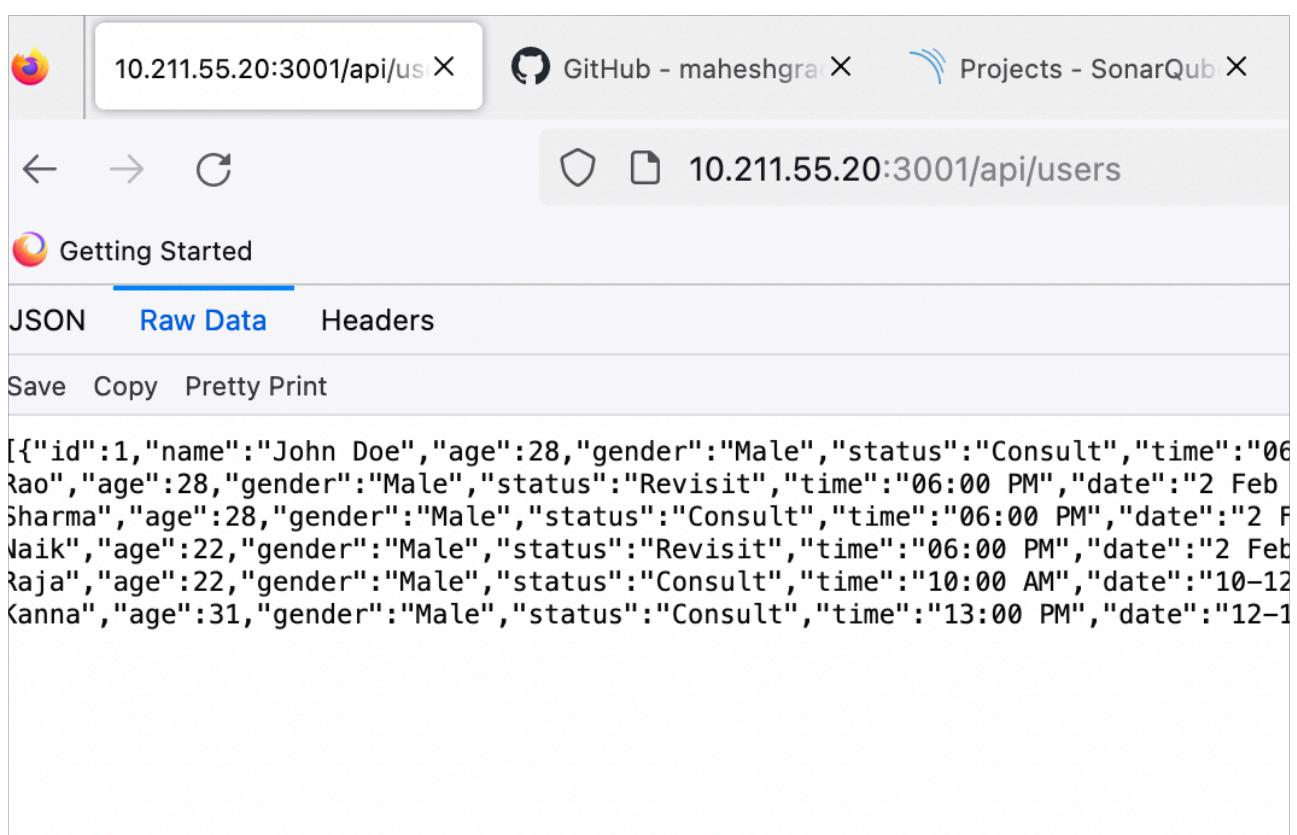
> server-mysql@1.0.0 start
> node app.js

listening on port 3001
GET /api/users
```



Each Time you trigger each instance you get

## 8) Final verification



| ID | Name     | Age | Gender | Status  | Time     | Date        |
|----|----------|-----|--------|---------|----------|-------------|
| 1  | John Doe | 28  | Male   | Consult | 06:00 PM | 2 Feb       |
| 2  | Rao      | 28  | Male   | Revisit | 06:00 PM | 2 Feb       |
| 3  | Sharma   | 28  | Male   | Consult | 06:00 PM | 2 Feb       |
| 4  | Naik     | 22  | Male   | Revisit | 06:00 PM | 2 Feb       |
| 5  | Raja     | 22  | Male   | Consult | 10:00 AM | 10-12 Kanna |
| 6  | Kanna    | 31  | Male   | Consult | 13:00 PM | 12-1        |

## NOTE

Note in client server the nginx should be upend running

Config files should we placed in right manner

Reverse Proxy should we created to trigger the server

```
lusy@localhost.localdomain:/etc (0.02s)
cd nginx

lusy@localhost.localdomain /etc/nginx (0.028s)
ls
conf.d          fastcgi.conf.default    koi-utf      mime.types.default  scgi_params        uwsgi_params.default
default.d       fastcgi_params          koi-win     nginx.conf          scgi_params.default  win-utf
fastcgi.conf    fastcgi_params.default mime.types  nginx.conf.default uwsgi_params
```

```
lusy@localhost.localdomain /etc/nginx/conf.d (0.022s)
cat reactjs.conf
server {
    listen 80;
    listen [::]:80;
    root /usr/share/nginx/html/build;

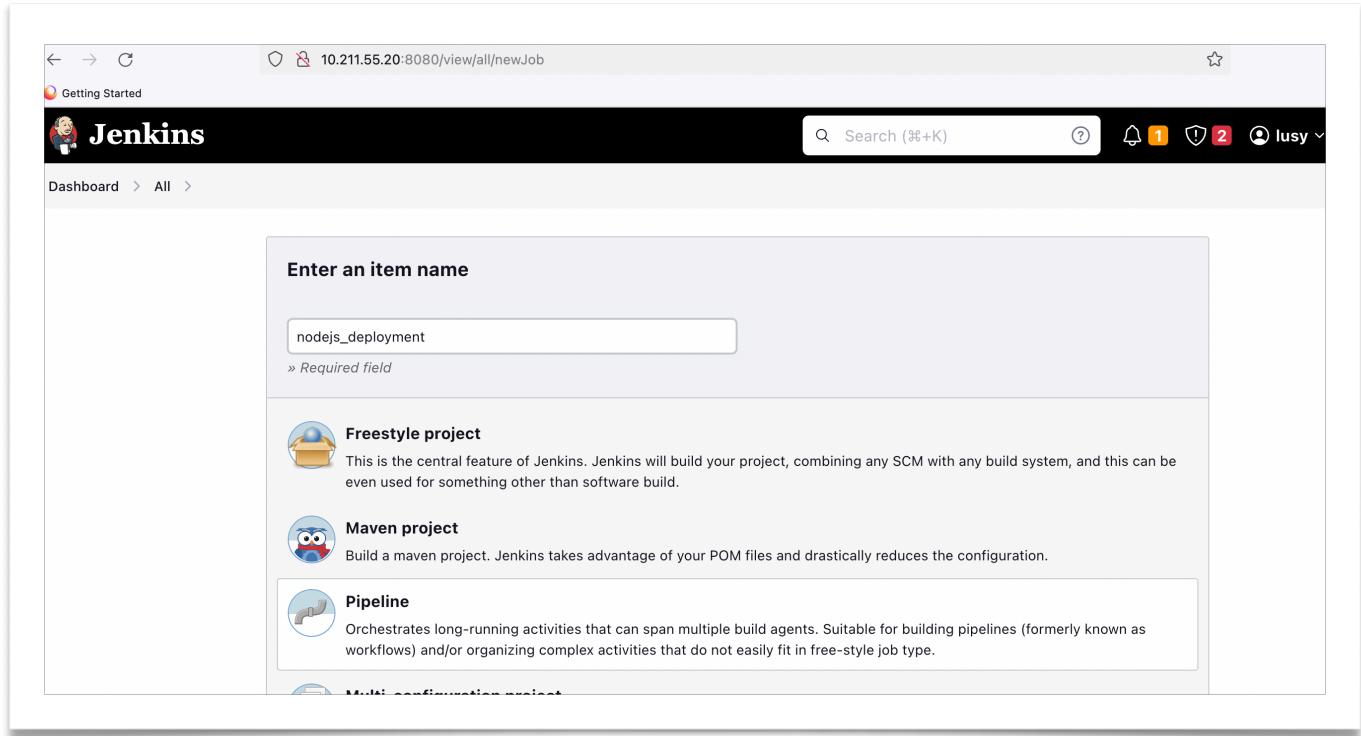
    location /api/users {
        proxy_pass http://10.211.55.20:3001/api/users;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }

    location /api/ {
        proxy_pass http://10.211.55.20:3001/api/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }

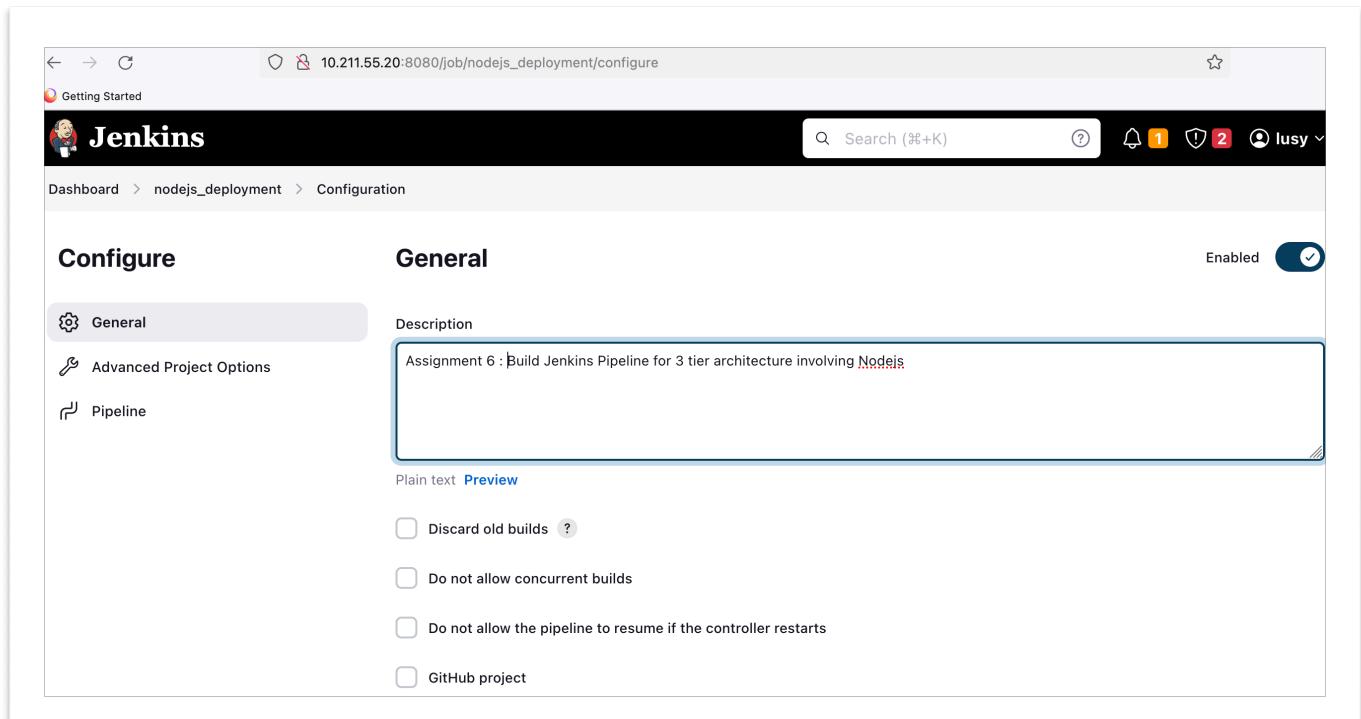
    location / {
        try_files $uri $uri/ /index.html;
    }
}
```

reverse proxy to trigger  
server in which DB is  
setup

# Jenkins setup



The screenshot shows the Jenkins 'New Job' creation interface. At the top, there's a search bar with 'Search (⌘+K)' and a user icon for 'lusy'. Below the header, the URL is 10.211.55.20:8080/view/all/newJob. The main area has a title 'Enter an item name' with a required field 'nodejs\_deployment'. It lists three project types: 'Freestyle project' (selected), 'Maven project', and 'Pipeline'. Each type has a brief description and an icon.



The screenshot shows the Jenkins Pipeline configuration page for the job 'nodejs\_deployment'. The left sidebar has tabs for 'General', 'Advanced Project Options', and 'Pipeline'. The 'General' tab is selected. It shows a 'Description' field containing 'Assignment 6 : Build Jenkins Pipeline for 3 tier architecture involving Nodejs'. There are also checkboxes for 'Plain text' and 'Preview'. On the right, there's an 'Enabled' toggle switch which is checked. Other options shown include 'Discard old builds', 'Do not allow concurrent builds', 'Do not allow the pipeline to resume if the controller restarts', and 'GitHub project'.

## Config Jenkins for nodejs and jdk11

JDK installations ^    Edited

Add JDK

JDK

Name

jdk11

Install automatically ?

≡ Install Oracle Java SE Development Kit from the website ?

Version

Java SE Development Kit 9.0.4

NodeJS installations

NodeJS installations ^    Edited

# My Script

```
pipeline {  
    agent any  
    tools {  
        nodejs "node"  
        jdk "jdk11"  
    }  
    environment {  
        SCANNER_HOME = tool 'sonar-scanners'  
    }  
    stages {  
        stage('git checkout') {  
            steps {  
                git branch: 'master', url: 'https://github.com/kartikhegadi/nodejs_client.git'  
            }  
        }  
        stage('npm install') {  
            steps {  
                sh 'npm install'  
            }  
        }  
        stage('npm with SonarQube') {  
            steps {  
                withSonarQubeEnv('sonarqube') {  
                    sh """  
                        $SCANNER_HOME/bin/sonar-scanner \  
                        -Dsonar.projectName=npmlclient \  
                        -Dsonar.java.binaries=\. \  
                        -Dsonar.projectKey=npmsclient \  
                        -Dsonar.source=. \  
                    """  
                }  
            }  
        }  
        stage('quality gate') {  
            steps {  
                timeout(time: 5, unit: 'MINUTES') {  
                    waitForQualityGate abortPipeline: true, credentialsId: 'sonarqube'  
                }  
            }  
        }  
        stage('npm run build') {  
            steps {  
                sh 'sudo npm run build'  
            }  
        }  
    }  
}
```

```
stage('npm run build') {  
    steps {  
        sh 'sudo npm run build'  
    }  
}  
stage('replace file .js' ) {  
    steps {  
        sh 'sudo mv -f /home/lusy/server_react_js/reactjs.conf /home/lusy/nodejs_client/build/static/js/'  
    }  
}  
stage('copy build file') {  
    steps {  
        sh 'sudo scp -r /home/jenkins/workspace/nodejs_client/build/ lusy10.211.55.22:/usr/share/nginx/html/'  
        sh 'echo "Files copied"'  
    }  
}  
post {  
    always {  
        emailext body: 'SMTP mail sent)', subject: 'Test Email', to: ' 2ke20cs086_t@kleit.ac.in'  
    }  
}
```

I have remotley in server i.e in 10.211.55.20  
i have saved that i am just moving to other machine

# Using the script build it

The screenshot shows the Jenkins interface for the 'php\_deployment' project. On the left, there's a sidebar with options like Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. The main area is titled 'Project php\_deployment' with a description: 'Assignment - 5 - create a simple php project pipeline /deploy to any web server & trigger the job using command line remotely'. Below this is a 'Permalinks' section showing recent builds. A 'Build History' tab is selected, showing build #5 from Oct 28, 2023, at 3:59 PM. The build status is green.

## Fetch the page

The screenshot shows a browser window with multiple tabs open. The active tab is 'Appointment Booking' at the URL '10.211.55.22/index.html/api/user'. The page displays a form for booking an appointment with fields for Patient Name, Phone Number, Doctor Name, Gender, Date, Age, Time, and a 'Book Appointment' button. Below the form is a table listing patient appointments:

| Patient                              | Status  | Appointment            | Phone                                    | Doctor     | Actions              |
|--------------------------------------|---------|------------------------|--|------------|----------------------|
| User Img<br>John Doe<br>28 yrs, Male | Consult | 06:00 PM<br>2 Feb 2021 | +91 987654321<br><a href="#">Contact</a> | Dr. Ananth | <a href="#">More</a> |
| User Img<br>Mukul Rao<br>yrs, Male   | Revisit | 2 Feb 2021             | +91 987654321<br><a href="#">Contact</a> | Dr. Ananth | <a href="#">More</a> |

## Email:

## Plugins

The screenshot shows the Jenkins plugin manager interface at the URL [10.211.55.20:8080/manage/pluginManager/installed](http://10.211.55.20:8080/manage/pluginManager/installed). The title bar includes the Jenkins logo, a search bar with placeholder "Search (⌘+K)", and various status icons. The main navigation bar has links for "Manage Jenkins" and "Plugins". On the left, there's a sidebar with "Updates" (21), "Available plugins", "Installed plugins" (selected), "Advanced settings", and "Download progress". A search bar at the top of the main content area contains the text "email". The main table lists installed plugins:

| Name  | Enabled  |
|---|--|
| Email Ext Recipients Column Plugin 27vb_9404db_b_018d | <input checked="" type="checkbox"/> <input type="checkbox"/> |
| Email Extension Template Plugin 1.5                   | <input checked="" type="checkbox"/> <input type="checkbox"/> |
| Validating Email Parameter Plugin 1.10                | <input checked="" type="checkbox"/> <input type="checkbox"/> |

Details for the "Email Extension Template Plugin" are shown:

Email Extension Template Plugin 1.5  
This plugin allows administrators to create global templates for the Extended Email Publisher.  
[Report an issue with this plugin](#)

A yellow callout box states: "This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information."

Details for the "Validating Email Parameter Plugin" are shown:

Validating Email Parameter Plugin 1.10  
Jenkins Validating Email Parameter Plugin  
[Report an issue with this plugin](#)

A pink callout box displays a security warning: "Warning: The currently installed plugin version may not be safe to use. Please review the following security notices:

- [Stored XSS vulnerability](#)

"

# Configurations

The screenshot shows the Jenkins System configuration page under the 'Manage Jenkins' section. The 'Content Token Reference' section is visible at the top. Below it, there's a form for SMTP settings:

- SMTP Port**: 465
- Reply-To Address**: (empty field)
- Charset**: UTF-8

Under the configuration, there's a checked checkbox for "Test configuration by sending test e-mail". The "Test e-mail recipient" field contains "karthikhegdi143@gmail.com". A success message "Email was successfully sent" is displayed next to the recipient field, with the word "testing" highlighted in red. A "Test configuration" button is also present.

The screenshot shows a Gmail inbox. The search bar at the top has "Search in emails". The toolbar includes Back, Archive, Spam, Delete, Mark as unread, Snooze, Add to Tasks, Move to, Labels, More, Active, Help, Settings, and a three-dot menu. The inbox list shows one email:

**Test email #5** (Inbox) 19

address not configured yet <karthikhegdi143@gmail.com>  
to me ▾ 01:47 (2 minutes ago) ⭐ ⓘ Reply

This is test email #5 sent from Jenkins

At the bottom of the email view, there are "Reply" and "Forward" buttons.

## Post-build Actions

### Editable Email Notification ?

Allows the user to disable the publisher, while maintaining the settings

Disable Extended Email Publisher ?

#### Project From

#### Project Recipient List ?

Comma-separated list of email address that should receive notifications for this project.

2ke20cs086\_t@kleit.ac.in

#### Project Reply-To List ?

Comma-separated list of email address that should be in the Reply-To header for this project.

\$DEFAULT\_REPLYTO

#### Content Type ?

HTML (text/html)

#### Default Subject ?

\$DEFAULT SUBJECT

#### Default Content ?

\$DEFAULT CONTENT

```
JDK installation skipped: Unknown CPU architecture: aarch64
Checking out Revision c812e96cd7b1b3575494bb9ab108ee0fef3715d8 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f c812e96cd7b1b3575494bb9ab108ee0fef3715d8 # timeout=10
Commit message: "Add files via upload"
> git rev-list --no-walk c812e96cd7b1b3575494bb9ab108ee0fef3715d8 # timeout=10
No emails were triggered.
Email was triggered for: Always
Email was triggered for: Success
Sending email for trigger: Always
JDK installation skipped: Unknown CPU architecture: aarch64
An attempt to send an e-mail to empty list of recipients, ignored.
Finished: SUCCESS
```

## E-mail Notification

SMTP server

smtp.gmail.com

Default user e-mail suffix [?](#)

[Advanced](#) ^  Edited

Use SMTP Authentication [?](#)

This is APP password

User Name

karthikhegdi143@gmail.com

Password

.....