

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
```

```
lusy@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 of the 'the_test' project. The top navigation bar includes the Jenkins logo and the project name 'the_test'. The sidebar on the left lists options: 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 features a large green circle with a white checkmark icon. To the right of the icon, the output of the build is displayed. It shows the build was started by user 'lusy' and running as SYSTEM. The workspace path is /var/lib/jenkins/workspace/the_test. The command executed was [the_test] \$ /bin/sh -xe /tmp/jenkins1841829317364151931.sh + echo 'Hello World!!!!'. The output concludes 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. 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

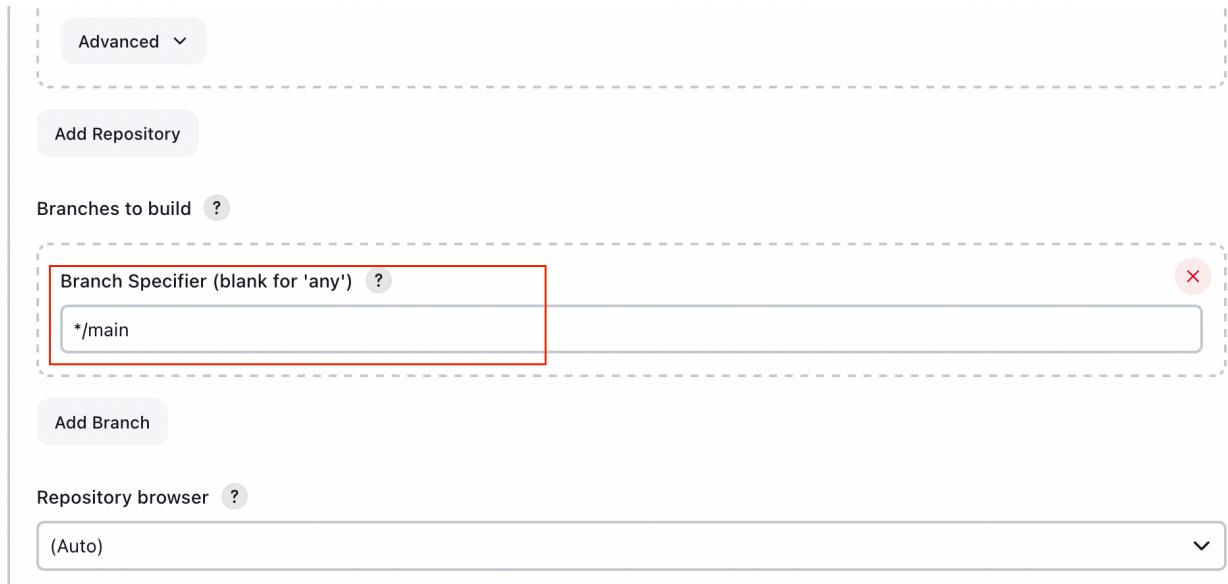
Credentials [?](#)
- none -

Add [▼](#)

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

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

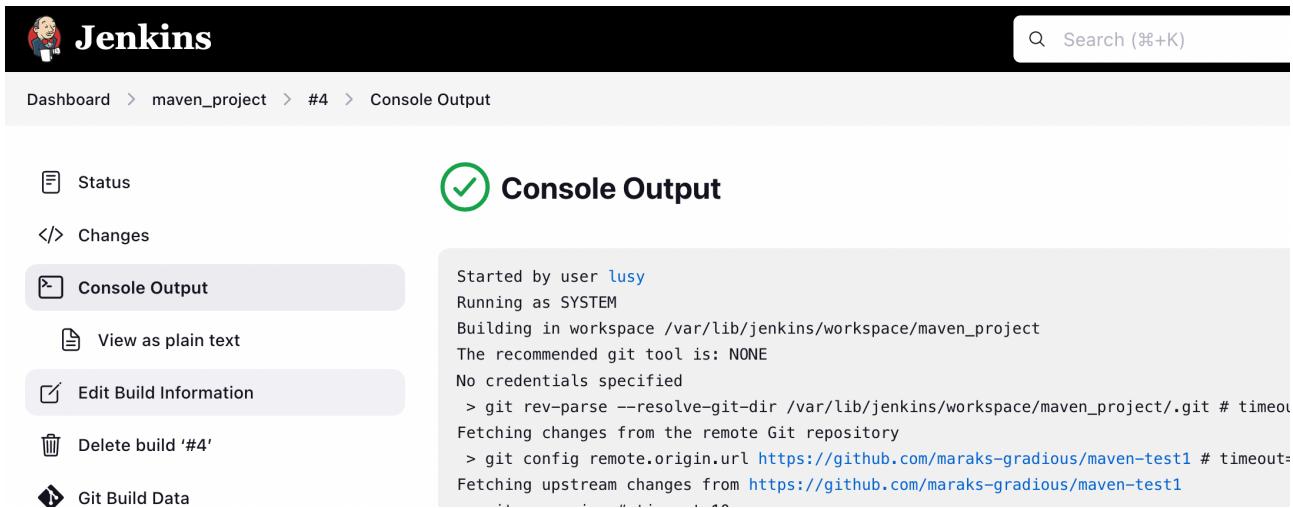
6. In the branches to build, modify master to main



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 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.

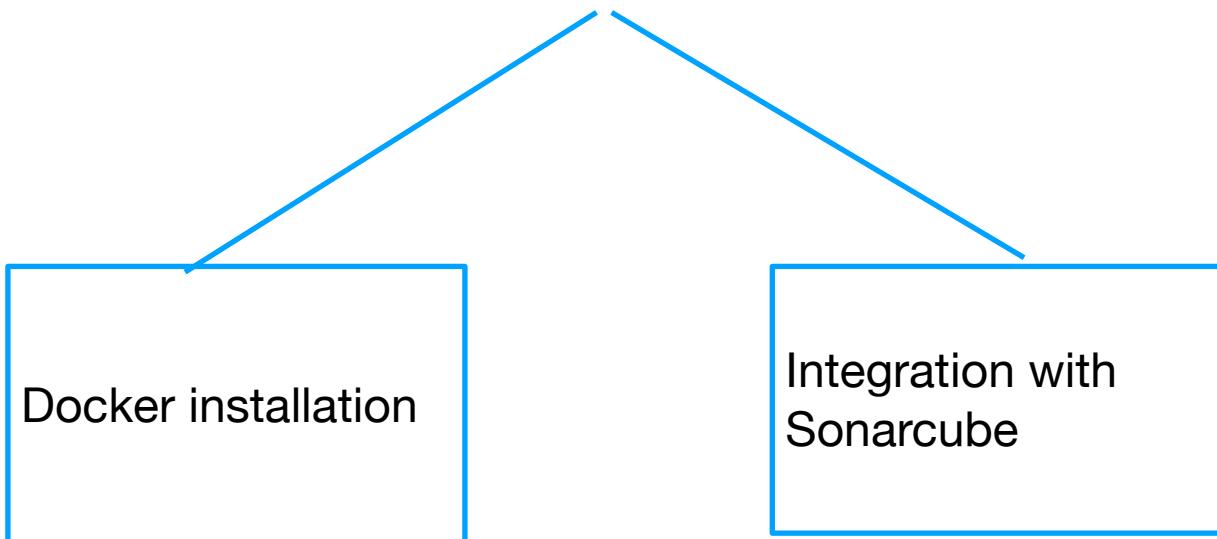
Jenkins -3

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

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

```

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: sha256:45c205d63a3341c88618528a0e0245620633f20c9883203371acf6f8536f1a4
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/mysqlld  arm     e248da1c6d57  3 weeks ago  507MB
kartik404/mysqlld  first_dhub  e248da1c6d57  3 weeks ago  507MB
mysqlld             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	Setup
Import from Bitbucket Cloud	Setup
Import from GitHub	Setup
Import from GitLab	Setup

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

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	Revoke

[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

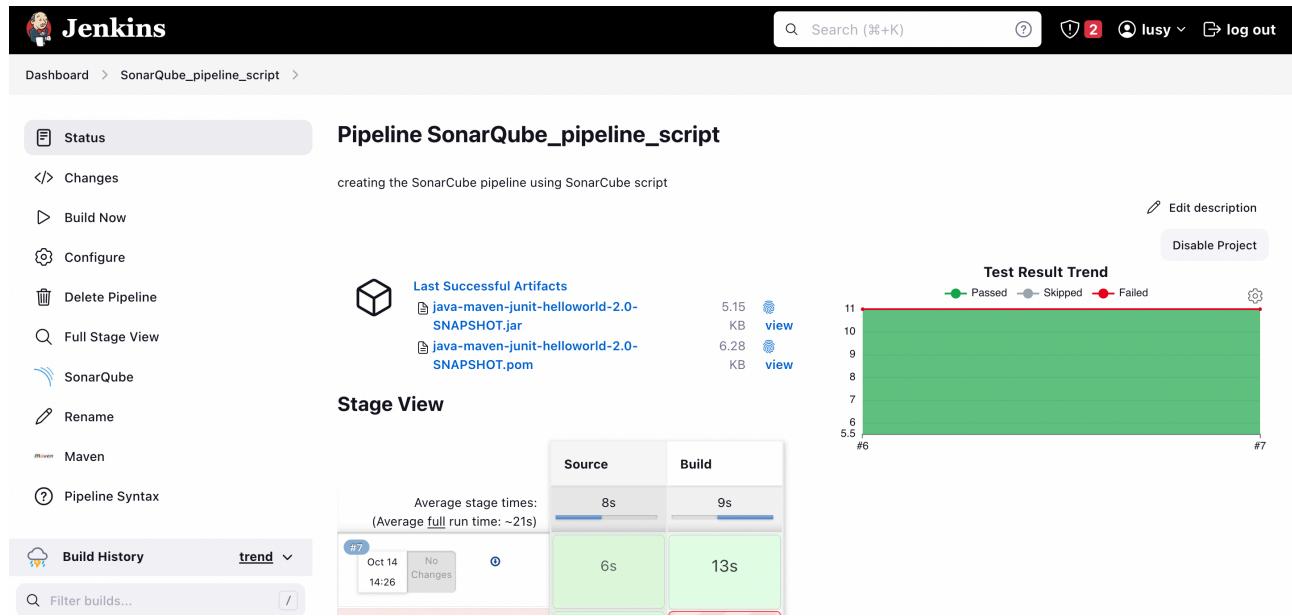
8. Build the pipeline and verify the build status

The screenshot shows the SonarQube Pipeline build status for build #7, which was successful on October 14, 2023, at 2:26:58 PM. A note in red text says "After all few build failure.. Finally i got it ..😊😊". The build artifacts section shows 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). The repository information indicates it was started by user "lusy" from the "https://github.com/marak-s-gradious/maven-test1" repository, specifically at revision 70707b80cef3f91a57347a7ed0f7482b4e3bf126, pointing to the "refs/remotes/origin/main" branch. There were no test failures.

The screenshot shows the Jenkins build status for build #7, which was successful. The "Console Output" tab is selected, showing the log output. The log starts with "[DEBUG]" entries for various Maven dependencies being resolved, such as org.apache.maven:maven-model-builder:jar:3.0:compile and org.sonatype.aether:aether-impl:jar:1.7:compile. The log then continues with "[INFO]" entries for the build process itself, including the execution of the SonarQube pipeline script.

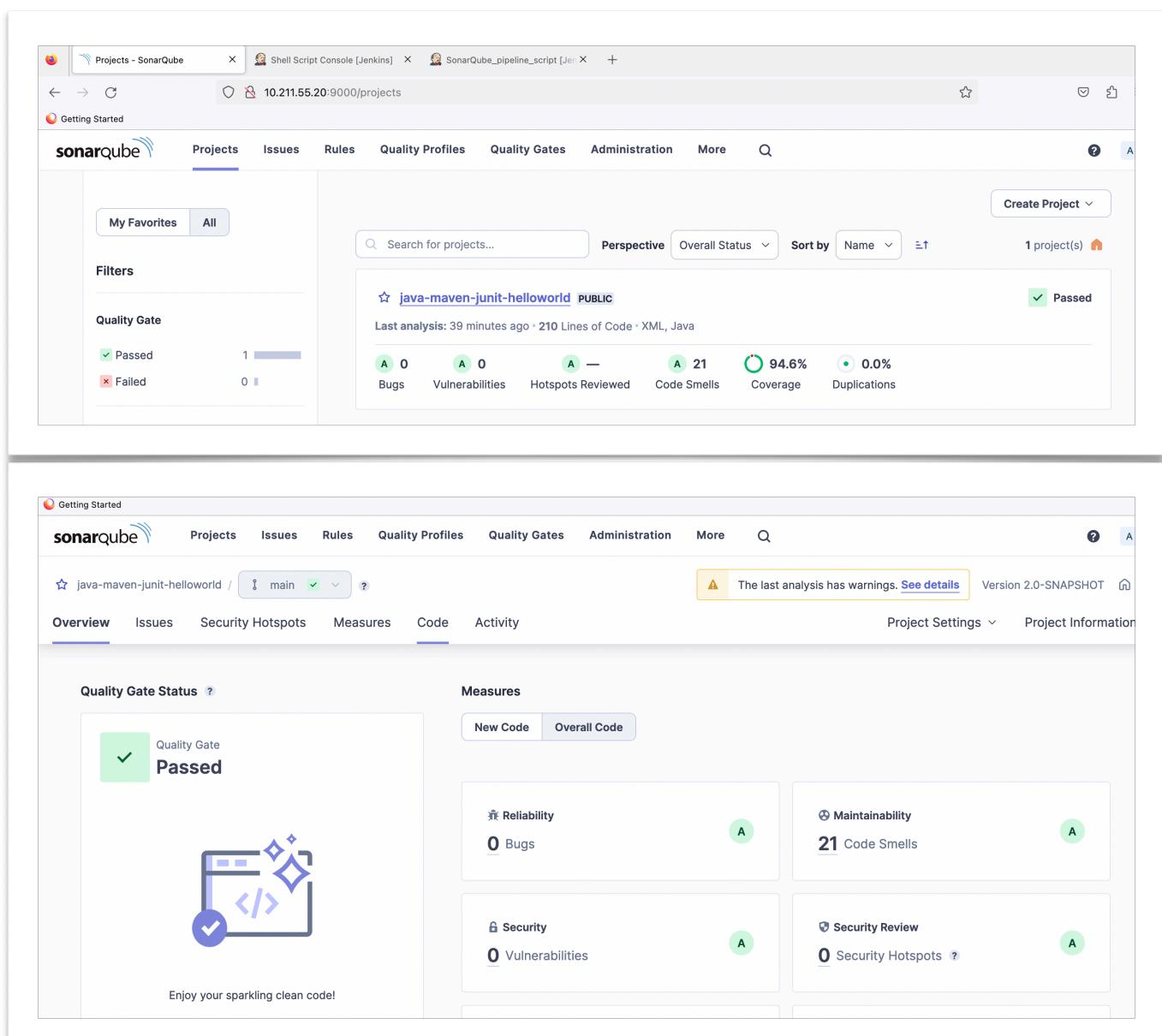
```
[DEBUG] org.apache.maven:maven-model-builder:jar:3.0:compile
[DEBUG] org.apache.maven:maven-aether-provider:jar:3.0:runtime
[DEBUG] org.sonatype.aether:aether-impl:jar:1.7:compile
[DEBUG] org.sonatype.aether:aether-spi:jar:1.7:compile
[DEBUG] org.sonatype.aether:aether-api:jar:1.7:compile
[DEBUG] 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 options like Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, SonarQube, Rename, Maven, Pipeline Syntax, and Build History. The main area is titled "Pipeline SonarQube_pipeline_script". It displays "Last Successful Artifacts" (java-maven-junit-helloworld-2.0-SNAPSHOT.jar and java-maven-junit-helloworld-2.0-SNAPSHOT.pom) and a "Test Result Trend" chart showing a green bar from 5.5 to 11. Below that is a "Stage View" showing "Source" (8s) and "Build" (9s) times with a total average stage time of 13s. A build history entry for Oct 14 at 14:26 is shown 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. The top navigation bar includes Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search bar. The main content area shows a summary for the "main" branch, indicating a "Passed" status with a green checkmark. It displays metrics such as 0 bugs, 0 vulnerabilities, 1 hotspot reviewed, 21 code smells, 94.6% coverage, and 0.0% duplications. Below this, the "Overview" tab is selected, showing the "Quality Gate Status" as "Passed" with a green checkmark. The "Measures" section provides detailed data on Reliability (0 bugs), Maintainability (21 code smells), Security (0 vulnerabilities), and Security Review (0 security hotspots). A message at the bottom encourages users to "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

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 original. The commit history shows a recent update to the Dockerfile by 'kartikhegadi'. The Dockerfile is highlighted with a red underline. Other files listed include dist, src/main/webapp, README.md, and pom.xml. A 'README.md' file is shown in a preview pane at the bottom.

File	Description	Time
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
- pipeline Pipeline Steps

</> Changes

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

⌚ Started by user [lousy](#)

git Revision: [7cd51da532e01f55ff1e96c6c47be7a59c10a8f8](#)
Repository: 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

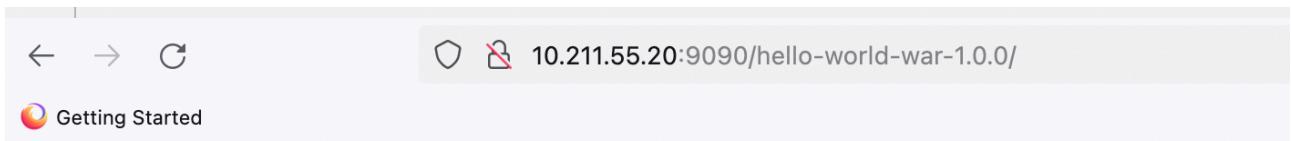
```

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') ?

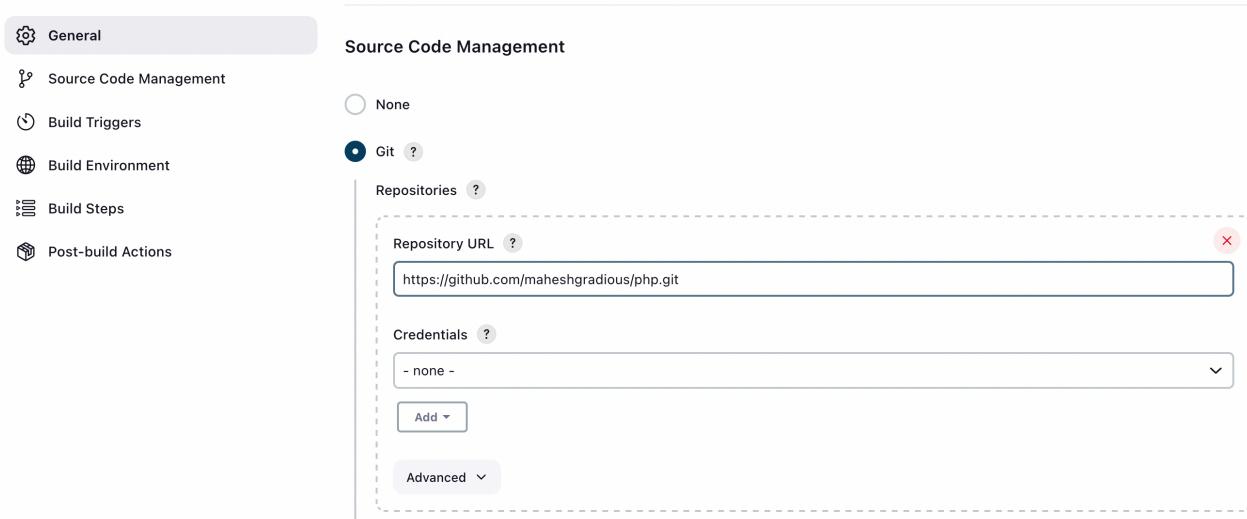
*/main

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'.

Create token here use in link to auto trigger

Build Triggers

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

Authentication Token

lousy@123

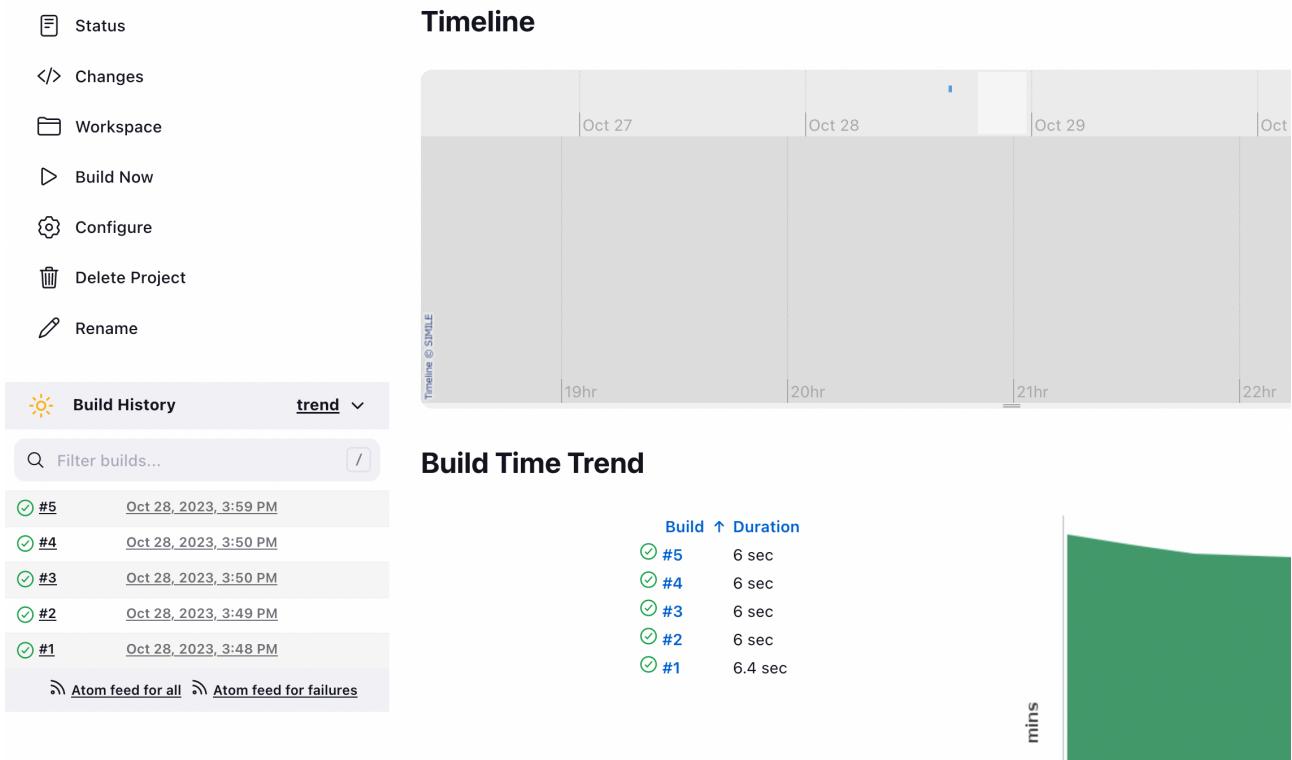
Use the following URL to trigger build remotely: JENKINS_URL/job/php_deployment/build?token=TOKEN_NAME or /buildWithParameters?token=TOKEN_NAME

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". The terminal also shows the output of the curl command, which is the "Hello World" test page.

```

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 "Hello World" test page at the URL "10.211.55.20/index.php". The page content is "Hello World". To the right of the browser is a terminal window showing the Jenkins pipeline logs. The logs show the execution of the "index.php" file, the chmod command, and the curl command used to trigger the build.

```

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

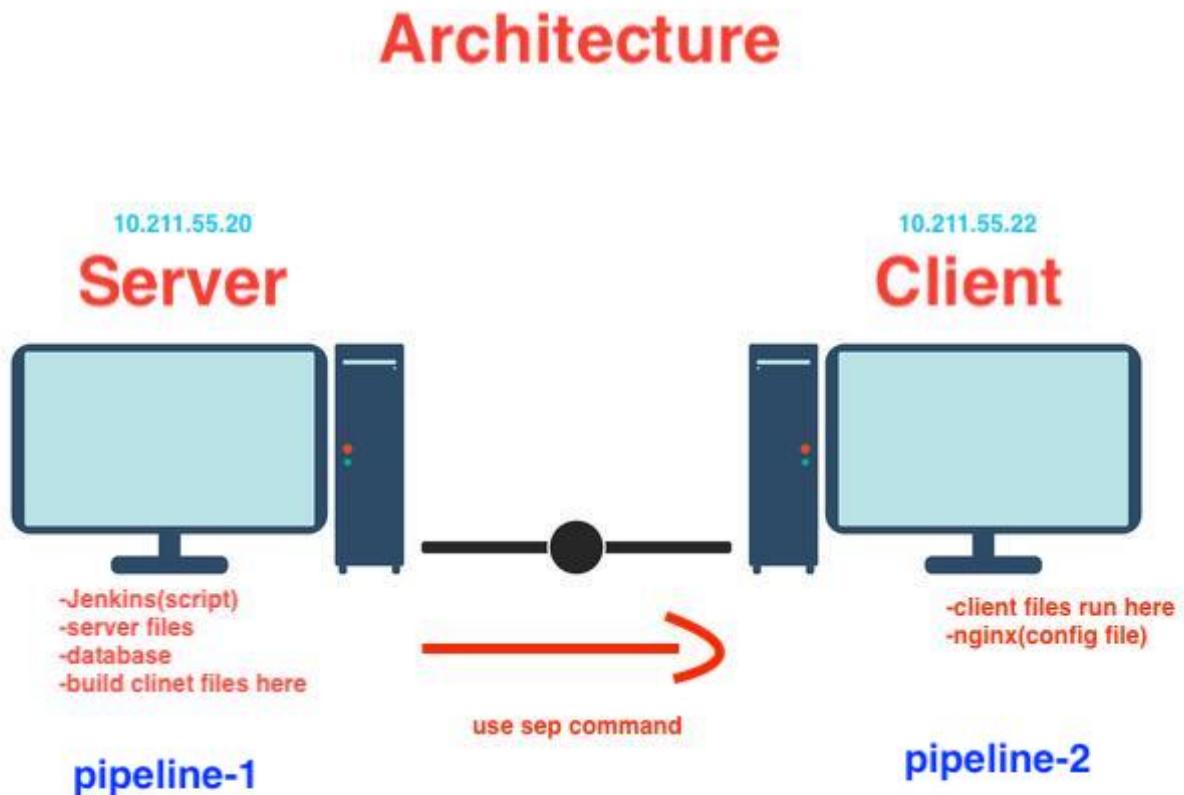
Understood. To follow the provided instructions and create the files/directory using the same name and case as provided in the task steps, please provide me with the specific names and case instructions for the files/directory you want to create.

Jenkins-6

Assignment - 6- 3 Tier architecture app involving NodeJS,

ReactS and MYSOL

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



PIPELINE-1(server-side)

1 creating the project

The screenshot shows the Jenkins configuration interface for the 'nodejs_server' project. The 'General' tab is selected. In the 'Description' field, the text 'nodejs project server side automation' is entered. The 'Enabled' switch is turned on. On the left sidebar, 'Pipeline' is selected. Below the main area, there are several checkboxes for build options: 'Discard old builds', 'Do not allow concurrent builds', 'Do not allow the pipeline to resume if the controller restarts', 'GitHub project', 'Pipeline speed/durability override', and 'Preserve stashes from completed builds'. A 'Plain text' link is available for previewing the configuration.

2 script

The screenshot shows the Jenkins configuration interface for the 'nodejs_server' project. The 'Pipeline' tab is selected. In the 'Definition' dropdown, 'Pipeline script' is chosen. The 'Script' editor contains the following Groovy code:

```
1+ node {
2+   try {
3+     // Stage 1: Install Node.js
4+     stage('Install Node.js') {
5+       sh 'sudo yum install -y nodejs'
6+     }
7+
8+     // Stage 2: Install MySQL
9+     stage('Install MySQL') {
10+
11+
12+   }
13+
14+   // Stage 3: Clone Git Repository
15+   stage('Clone Git Repository') {
16+     sh 'git clone https://github.com/username/repo.git'
17+   }
18+ }
```

Below the script, the 'Use Groovy Sandbox' checkbox is checked. At the bottom, there are 'Save' and 'Apply' buttons. A preview window on the right shows a Jenkins job summary.

3 steps in script

```
nodejs deployment
deployment at server-side

steps
1. install nodejs
2. install mysqld
3. git clone
   1. link :https://github.com/kartikhegadi/nodejs\_server.git
   2. as main branch
4. auto login mysqld with credentials
   1. as root user
   2. password is "password"
   3. create a database doctor_appointment;
   4. exit
5. import data base
   1. doctor_appointment to mysql
   2. as root user
   3. password is "password"
6. open app.js
   1. go to const pool function
   2. in that go to password = "password" change to password
   3. save it and come back
7. in this directory /home/lusy/nodejs_server
   1. run npm start
```

4 whole script

```
===== (pipeline)
node {
try {
// Stage 1: Install Node.js
stage('Install Node.js') {
sh 'sudo yum install -y nodejs'
}

// Stage 2: Install MySQL
stage('Install MySQL') {
sh 'already installed'
}

// Stage 3: Clone Git Repository
stage('Clone Git Repository') {
checkout([$class: 'GitSCM', branches: [[name: 'main']], userRemoteConfigs: [[url: 'https://github.com/kartikhegadi/nodejs_server.git']]])
}

// Stage 4: Auto-login to MySQL and create a database
stage('Create MySQL Database') {
sh 'mysql -u root --password="password" -e "CREATE DATABASE IF NOT EXISTS doctor_appointment;"' creating data-base if it is not exist
}

// Stage 5: Import the database
stage('Import Database') {
sh 'mysql -u root --password="password" doctor_appointment < doctor_appointment.sql'
}
```

```

// Stage 6: Modify app.js
stage('Modify app.js') {
    sh 'sed -i \'s/password = "root"/password = "password"/' /var/lib/jenkins/workspace/nodejs_server/app.js'
}

// Stage 7: Run the Node.js Application
stage('Run Node.js Application') {
    dir('/var/lib/jenkins/workspace/nodejs_server/') {
        sh 'npm install'      start server
        sh 'npm start'
    }
}

} catch (Exception e) {
    currentBuild.result = 'FAILURE'
    throw e
} finally {

}

```

5 build it

The screenshot shows the Jenkins Pipeline nodejs_server stage view. It displays a timeline of stages: Install Node.js, Install MySQL, Clone Git Repository, Create MySQL Database, Import Database, Modify app.js, Run Node.js Application, and Cleanup. Stage #14 is shown with a duration of 1min 11s. Stage #13 is shown with a duration of 3min 8s. The 'Build Now' button is highlighted.

	Install Node.js	Install MySQL	Clone Git Repository	Create MySQL Database	Import Database	Modify app.js	Run Node.js Application	Cleanup	
Average stage times:	1s	16ms	6s	291ms	280ms	287ms	2min 37s	12ms	
#14	Nov 01 06:20	No Changes	1s	14ms	6s	279ms	280ms	283ms	1min 11s
#13	Nov 01 06:18	No Changes	1s	9ms	6s	283ms	279ms	279ms	3min 8s

6 out put

The screenshot shows the Jenkins Pipeline nodejs_server #14 build log. The output includes npm audit results and a red arrow pointing to the 'server started and listening' message at the bottom.

```

+ sed -i 's/password = "root"/password = "password"/' /var/lib/jenkins/workspace/nodejs_server/app.js
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Run Node.js Application)
[Pipeline] dir('/var/lib/jenkins/workspace/nodejs_server')
Running in /var/lib/jenkins/workspace/nodejs_server
[Pipeline] {
[Pipeline] sh
+ npm install

up to date, audited 116 packages in 6s

11 packages are looking for funding
  run `npm fund` for details

3 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
[Pipeline] sh
+ npm start

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

listening on port 3001

```

server started and listening

Note*-> before going to server we should conf some things in Jenkins and we need to set the credentials to use gmail and ssh (to move the files form one machine to another)**

a) sonar cube

SonarQube servers

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

Environment variables

SonarQube installations

List of SonarQube installations

Name	sonar
Server URL	Default is http://localhost:9000



b) Java Development Toolkit

JDK installations

JDK installations ^ Edited

Add JDK

JDK	Name
	jdk11
<input checked="" type="checkbox"/> Install automatically	?
≡ Install Oracle Java SE Development Kit from the website ?	
Version	Java SE Development Kit 9.0.4

c) Node js extension



d) Credentials

Credentials

T	P	Store ↓	Domain	ID	Name
💻	👤	System	(global)	sonarcube_key	sonarcube_key
💻	👤	System	(global)	ea2bd294-296b-4a4d-8392-5b88eec4bb74	2ke20cs086_@kleit.ac.in /***** (Hey kartihik .. this is message from jenkins.)
⌚	👤	System	(global)	MySSHKey	Iusy
💻	👤	System	(global)	fdbed7f5-c072-4b6b-ac6a-a18ce9b17eeb	2ke20cs086_t@kleit.ac.in/***** (Hey kartihik .. this is message from jenkins.)
💻	👤	System	(global)	gmail.conf	2ke20cs086_t@kleit.ac.in/***** (gmail.conf)

Note***-> Email configurations

a) downloading plugins

The screenshot shows the Jenkins plugin manager interface. The title bar indicates the URL is 10.211.55.20:8080/manage/pluginManager/installed. The left sidebar has tabs for 'Available plugins', 'Installed plugins' (which is selected), 'Advanced settings', and 'Download progress'. The main area is titled 'Plugins' with a count of 21. A search bar at the top right contains the text 'email'. Below the search bar, a table lists a single plugin:

Name	Enabled
Email Ext Recipients Column Plugin 27.vb_9404db_b_018d This plugin is a sample to explain how to write a Jenkins plugin. Report an issue with this plugin.	<input checked="" type="checkbox"/> <input type="button" value="Uninstall"/>

b) Setting the app password after verifying the 2 step verification

c) E-mail notification settings

d) Extended Email notifications

The screenshot shows the Jenkins system configuration page under 'Email Notifications'. The title bar indicates the URL is 10.211.55.20:8080/manage/system. The left sidebar has tabs for 'Dashboard', 'Manage Jenkins', 'System' (which is selected), and 'Jenkins'. The main area has sections for 'Content Token Reference', 'SMTP Port' (set to 465), 'Reply-To Address' (empty), 'Charset' (set to UTF-8), and a 'Test configuration' section. In the 'Test configuration' section, the 'Test configuration by sending test e-mail' checkbox is checked, and the recipient is set to 'karthikhegdi143@gmail.com'. A success message 'Email was successfully sent' and the word 'testing' are displayed.

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.

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 ?

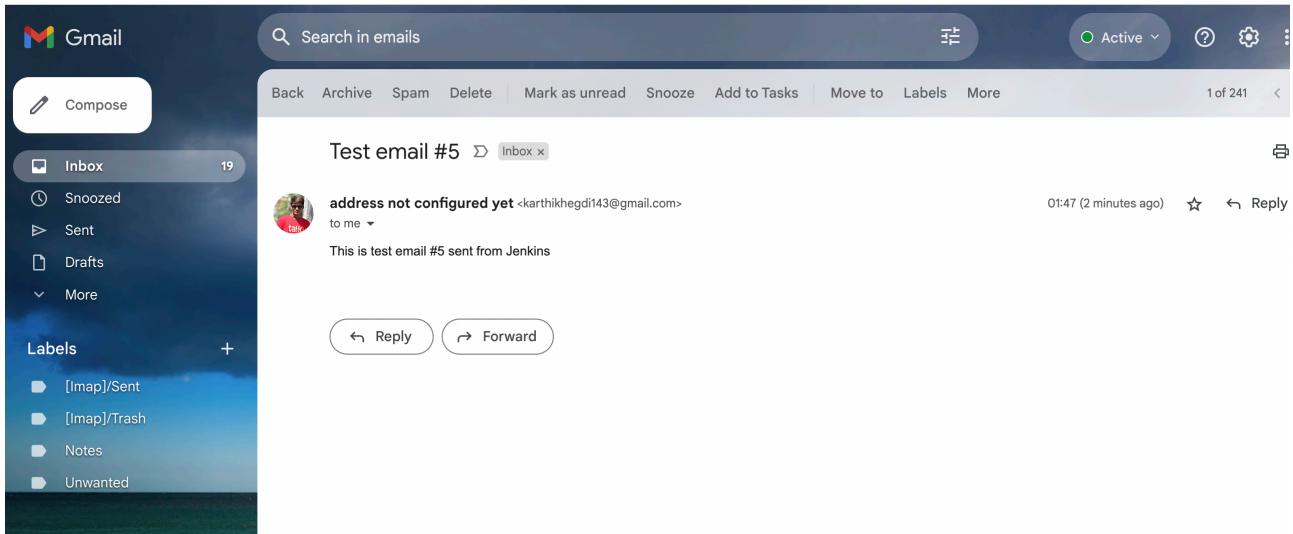
\$DEFAULT SUBJECT

Default Content ?

\$DEFAULT_CONTENT

e) Test mail

(Same steps I have set for for other mail and build the script)



PIPELINE-2(clinet-side)

a) name the project select it as pipeline

A screenshot of a web browser displaying the Jenkins dashboard. The address bar shows the URL 10.211.55.20:8080/view/all/newJob. The page title is "Jenkins". The main content area has a search bar with "Search (⌘+K)". Below the search bar, there are notification icons for 1 message, 2 builds, and 1 user labeled "lousy". The main form is titled "Enter an item name" and contains a text input field with "nodejs_deployment" entered. A note below the input says "» Required field". There are four project types listed: "Freestyle project" (represented by a blue box icon), "Maven project" (represented by a red owl icon), "Pipeline" (represented by a green circular icon), and "Multi-configuration project" (represented by a blue gear icon). The "Pipeline" option is currently selected.

b) About the project

The screenshot shows the Jenkins configuration interface for a job named 'nodejs_deployment'. The 'General' tab is selected. The 'Description' field contains the text: 'Assignment 6 : Build Jenkins Pipeline for 3 tier architecture involving Nodejs'. Below the description are several checkboxes: 'Discard old builds', 'Do not allow concurrent builds', 'Do not allow the pipeline to resume if the controller restarts', and 'GitHub project'. A 'Plain text Preview' link is also present. On the left sidebar, there are links for 'General', 'Advanced Project Options', and 'Pipeline'.

c) Manually conf the nginx config and set default 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
```

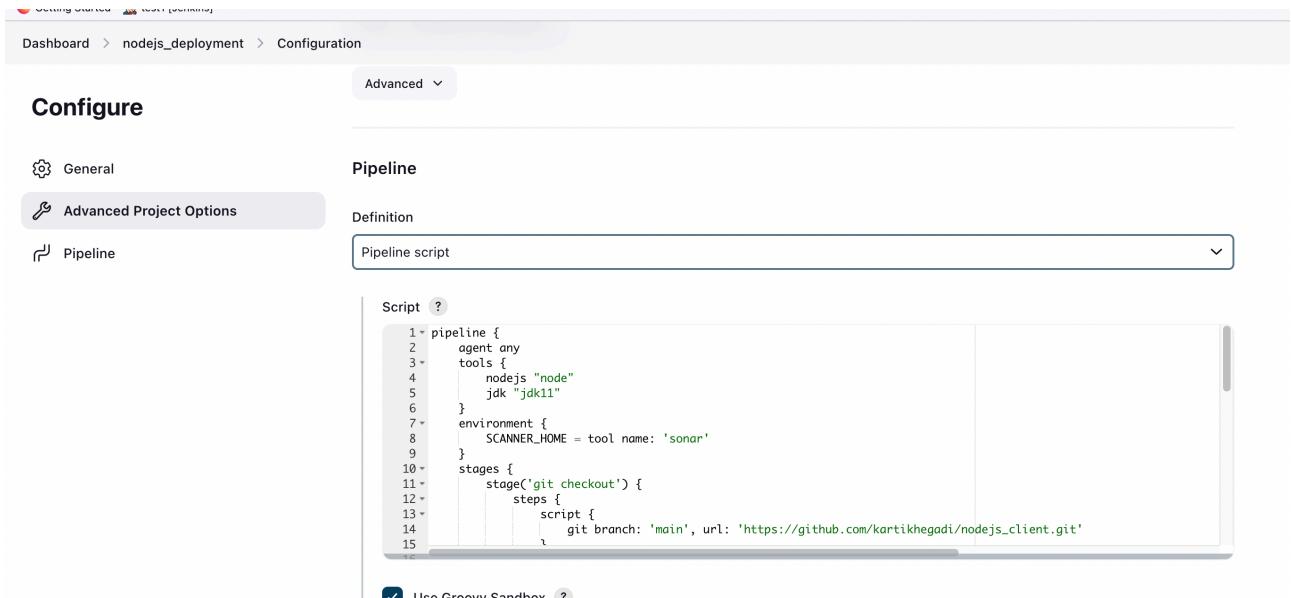
```
lusy@localhost.localdomain /etc/nginx/conf.d (0.022s)
cat reactjs.conf
server {
    listen 80;
    listen [::]:80;
    root /usr/share/nginx/html/build;

    location /api/users {
        proxy_pass http://10.211.55.20:3001/api/users;           reverse proxy to trigger
        proxy_set_header Host $host;                                server in which DB is
        proxy_set_header X-Real-IP $remote_addr;                    setup
    }

    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;
    }
}
```

d) Pipeline script



Full script

2 November 2023 at 12:57 AM

```
pipeline {
  agent any
  tools {
    nodejs "node"
    jdk "jdk11"
  }
  environment {
    SCANNER_HOME = tool name: 'sonar'
  }
  stages {
    stage('git checkout') {
      steps {
        script {
          git branch: 'main', url: 'https://github.com/kartikhegadi/nodejs_client.git'
        }
      }
    }
    stage('npm install') {
      steps {
        sh 'npm install'
      }
    }
    stage('npm with SonarQube') {
      steps {
        withSonarQubeEnv('sonar') {
          sh """
            ${SCANNER_HOME}/bin/sonar-scanner \
            -Dsonar.projectName=npmlclient \
            -Dsonar.projectKey=npmclient \
            -Dsonar.sources=.
            -Dsonar.language=js
          """
        }
      }
    }
  }
}
```

```

        }
    }
}

stage('copy build folder and run build (client)') {
    steps {
        sh 'sudo npm run build'
        sh 'sudo scp -r /home/jenkins/workspace/nodejs_client/build/ lusy@10.211.55.22:/usr/share/nginx/html/'
        echo "Files copied"
    }
}
post {
    always {
        emailext body: 'Hey! This email is sent from Jenkins about your build status! 😊', subject: 'Build Status of Jenkins', to: '2ke20cs086_t@kleit.ac.in'
    }
}
=====
===== (script running )

```

e) Build status

Pipeline nodejs_deployment

Average stage times:
(Average full run time: ~2min 38s)

Declarative: Tool Install	git checkout	npm install	npm with SonarQube	copy build folder and run build (client)	Declarative: Post Actions
215ms	6s	2min 23s	756ms	678ms	5s

#52 Nov 01 05:00 No Changes

f) Out-put

The screenshot shows a web browser window with multiple tabs open. The active tab is titled 'Appointment Booking'. The page content is a form titled 'Welcome to Gradious Doctor Appointment Booking' with fields for Patient Name, Phone Number, Doctor Name, Gender, Date, Age, Time, and a 'Book Appointment' button. Below this, there is a table listing patient appointments:

Patient	Status	Appointment	Phone	Doctor	Actions
User Img John Doe 28 yrs, Male	Consult	06:00 PM 2 Feb 2021	+91 987654321 Contact	Dr. Ananth	⋮
User Img Mukul Rao yrs, Male	Revisit	2 Feb 2021	+91 987654321 Contact	Dr. Ananth	⋮

g) Final mail

The screenshot shows a Gmail inbox. The 'Inbox' folder is selected, showing 278 messages. One message is highlighted, titled 'Build Status of Jenkins' from '2ke20cs086_t@kleit.ac.in' (to me). The email body says: 'Hey! This email is sent from Jenkins about your build status! 😊'. Below the message are 'Reply' and 'Forward' buttons.