DEVOPS TASK 2

Step 1: Open Jenkins and click "New Item."

Step 2 : Enter a project name, select "Pipeline," then click OK.

Step 3 : In the General tab, scroll down to the "Pipeline" section.

Step 4 : Under "Definition," select "Pipeline script from SCM."

Step 5 : In the SCM field, choose "Git," then enter your GitHub repository URL.

Step 6 : Ensure your GitHub repository contains the following files:

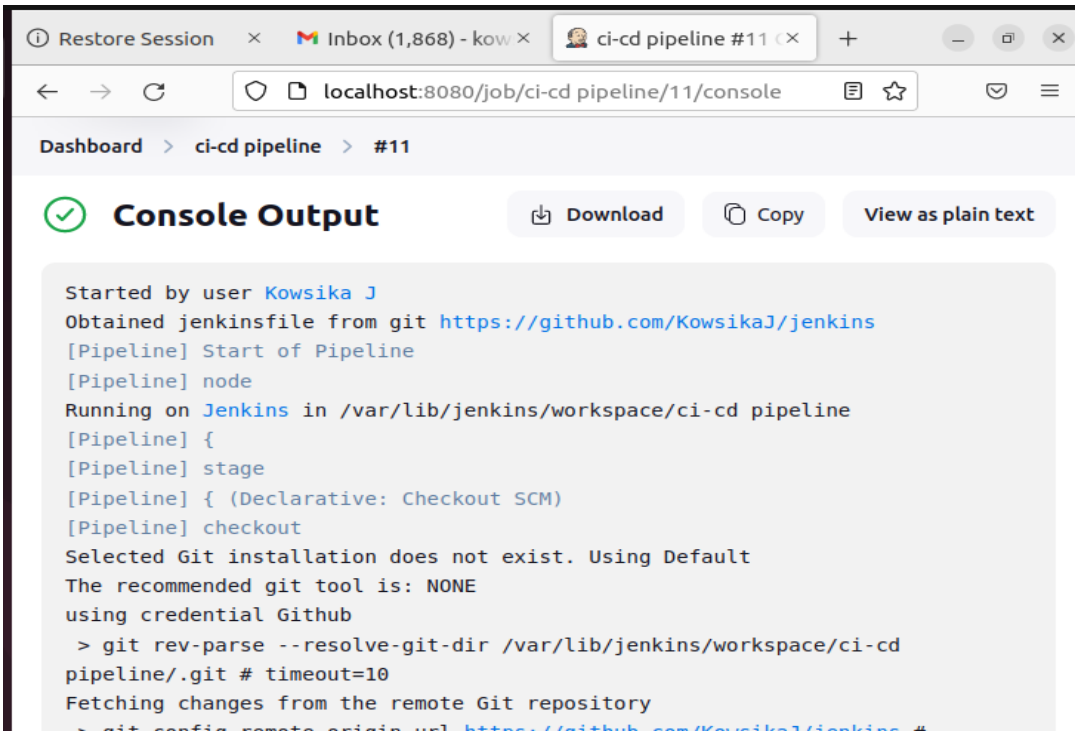
- app.py (Application code)
- Dockerfile (Instructions to build the Docker image)
- docker-compose.yml (For container orchestration)
- Jenkinsfile (Defines the pipeline stages)
- requirements.txt (Dependencies for the application)

Step 7 : Click "Save" to apply the configuration.

Step 8 : On the project page, click "Build Now" to start the job.

Step 9 : The build status will appear in the Build History panel (left-side panel).

Step 10 : Click on the latest build (#1, #2, etc.) and select "Console Output" to view logs and results.

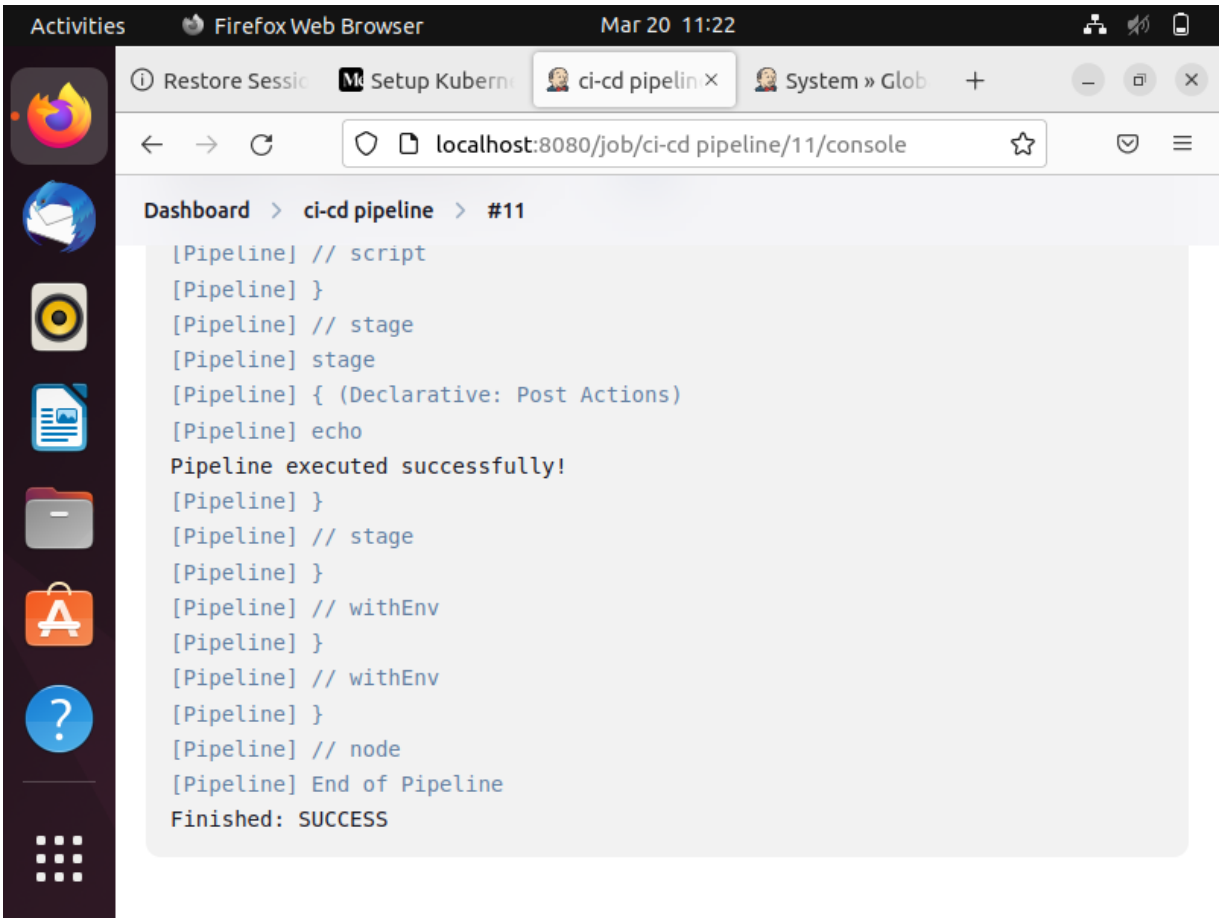


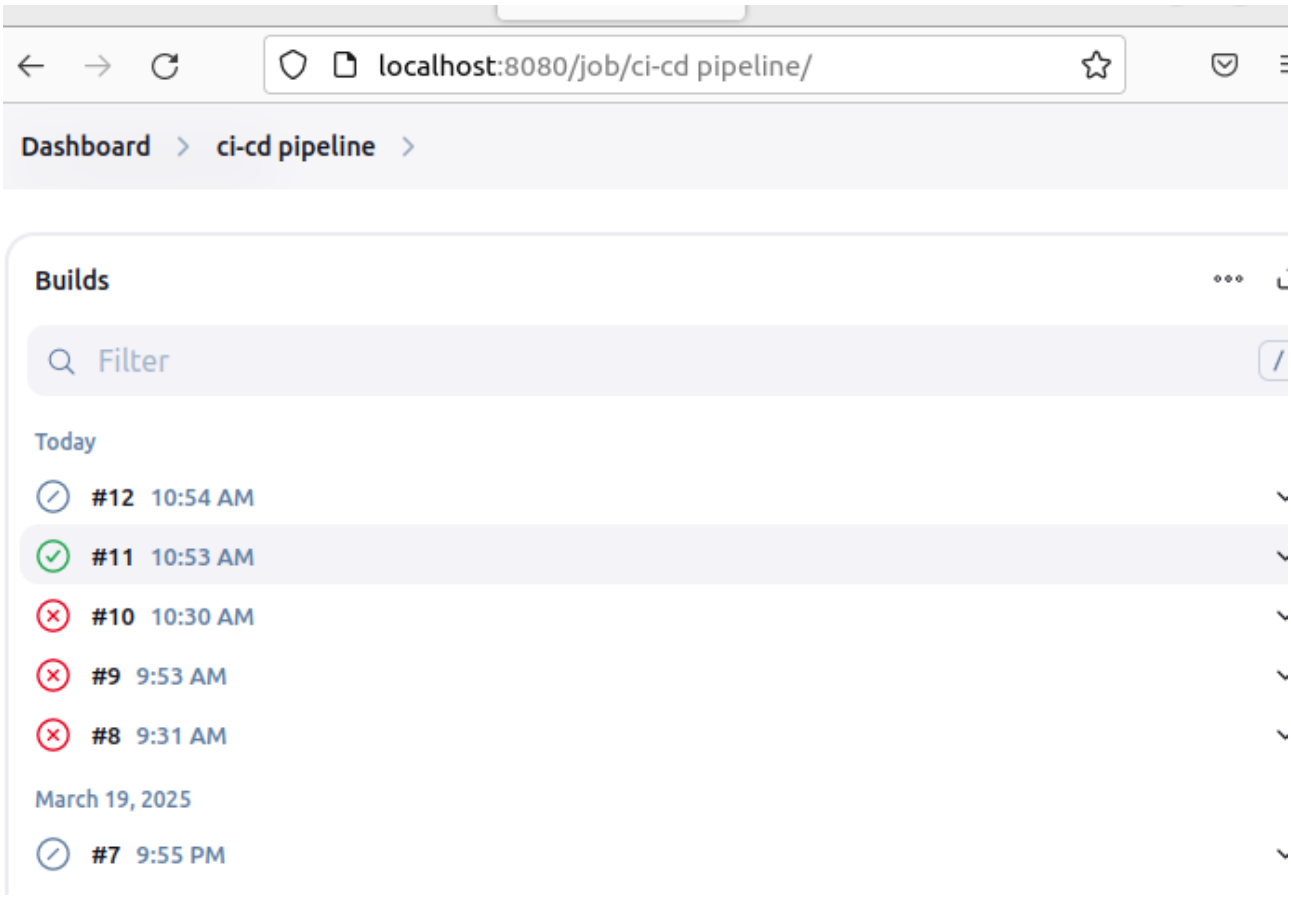
Dashboard > ci-cd pipeline > #11

```
[Pipeline] withCredentials
Masking supported pattern matches of $DOCKER_PASS
[Pipeline] {
[Pipeline] sh
+ docker login -u kowsikaj --password-stdin
+ echo ****

WARNING! Your credentials are stored unencrypted in '/var/lib/jenkins/.docker
/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
```





Commands to overcome error in Task 2:

1. Usermod -aG docker jenkins
2. Apt-get install docker-compose
3. Chmod 777 /var/run/docker.sock