

ASSIGNMENT 2

CI/CD Pipeline Setup Using Jenkins

In this assignment, we will set up a **Continuous Integration and Continuous Deployment (CI/CD) pipeline** using **Jenkins**. This pipeline will automate the process of building, testing, and deploying a **containerized web application**.

We will use:

Jenkins – For automation

GitHub – For managing source code

Docker – For containerization

Docker Hub (or another registry) – For storing container images

Steps to Set Up the CI/CD Pipeline

Step 1: Install Jenkins

1. Download and install **Jenkins** from the official website.
 2. Start Jenkins and complete the setup by installing the recommended plugins.
 3. Make sure you install the **Git** and **Docker** plugins to integrate with these tools.
-

Step 2: Create a New Pipeline Job

1. Open **Jenkins Dashboard** and click on "**New Item**".
 2. Select "**Pipeline**" as the job type and give it a name.
 3. Click **OK** and configure the pipeline settings.
-

Step 3: Configure the GitHub Repository

1. In the Jenkins job settings, go to **Source Code Management**.
 2. Select **Git** and enter the **GitHub repository URL**.
 3. If the repository is private, add your **GitHub credentials**.
 4. Set up **webhooks** in GitHub to trigger the pipeline automatically when changes are made.
-

Step 4: Write the Pipeline Script (Jenkinsfile)

The pipeline script defines the **stages** for automation.

Clones the GitHub repository

Builds a Docker image for the application

Pushes the image to Docker Hub

Step 5: Add Docker Credentials (If needed)

1. In Jenkins, go to **Manage Jenkins** → **Credentials**.
2. Add a new **Docker Hub API Key** or login credentials.
3. Use this credential in the **Jenkins file** for authentication.

Step 6: Trigger the Pipeline

- **Manually:** Click "**Build Now**" in Jenkins.
- **Automatically:** Set up **GitHub Webhooks** so Jenkins runs the pipeline when new code is pushed.

Output

When the pipeline runs successfully, you should see:

- ✅ **Repository cloned from GitHub**
- ✅ **Docker image built and tagged**
- ✅ **Docker image pushed to Docker Hub**

The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, user name 'SATHEESH K', and a 'log out' button. The breadcrumb trail is 'Dashboard > task_2 > #3'. On the left sidebar, the 'Console Output' tab is selected. The main area displays the console output for a build that was aborted. The output shows the pipeline starting, cloning the repository, and then failing with an error: 'Error when executing always post condition: Also: org.jenkinsci.plugins.workflow.actions.ErrorAction\$ErrorId: 59547cfd-fc08-4c27-93b6-b690551069f2 org.jenkinsci.plugins.workflow.steps.MissingContextVariableException: Required context class hudson.FilePath is missing'. Below this, a second screenshot shows the same console output but at a later stage where the build is successful. The output shows the pipeline cloning the repository, fetching upstream changes, and then building the Docker image. The final output is 'Building...' followed by '[Pipeline] }'.

```
Started by user SATHEESH K
Obtained Jenkinsfile from git git@github.com:satheesh1022005/DemoRepos.git
[Pipeline] Start of Pipeline
[Pipeline] node
Still waiting to schedule task
There are no nodes with the label 'docker'
Resuming build at Sat Mar 22 03:53:44 UTC 2025 after Jenkins restart
Ready to run at Sat Mar 22 03:53:44 UTC 2025
Aborted by SATHEESH K
[Pipeline] // node
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] cleanWs
Error when executing always post condition:
Also: org.jenkinsci.plugins.workflow.actions.ErrorAction$ErrorId: 59547cfd-fc08-4c27-93b6-b690551069f2
org.jenkinsci.plugins.workflow.steps.MissingContextVariableException: Required context class hudson.FilePath is missing

returning changes from the remote git repository
> git config remote.origin.url https://github.com/ranjanisengottaian/assignment_84/ # timeout=10
Fetching upstream changes from https://github.com/ranjanisengottaian/assignment_84/
> git --version # timeout=10
> git --version # 'git version 2.39.5'
> git fetch --tags --force --progress -- https://github.com/ranjanisengottaian/assignment_84/
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 12ae429b6a11a8d9a446cb7b8c3571f5e530f7e8 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 12ae429b6a11a8d9a446cb7b8c3571f5e530f7e8 # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git branch -D main # timeout=10
> git checkout -b main 12ae429b6a11a8d9a446cb7b8c3571f5e530f7e8 # timeout=10
Commit message: "Merge remote-tracking branch 'origin/main'"
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] echo
Building...
[Pipeline] }
```