

JENKINS ASSIGNMENT

1. Build and Deploy a Simple App using Jenkins

Step 1: Install Required Plugins

1. Open Jenkins Dashboard → Manage Jenkins → Manage Plugins.

2. Install the following plugins:

- Pipeline
- Git Plugin
- Docker Pipeline Plugin (if using Docker)

Step 2: Create a New Jenkins Pipeline Job

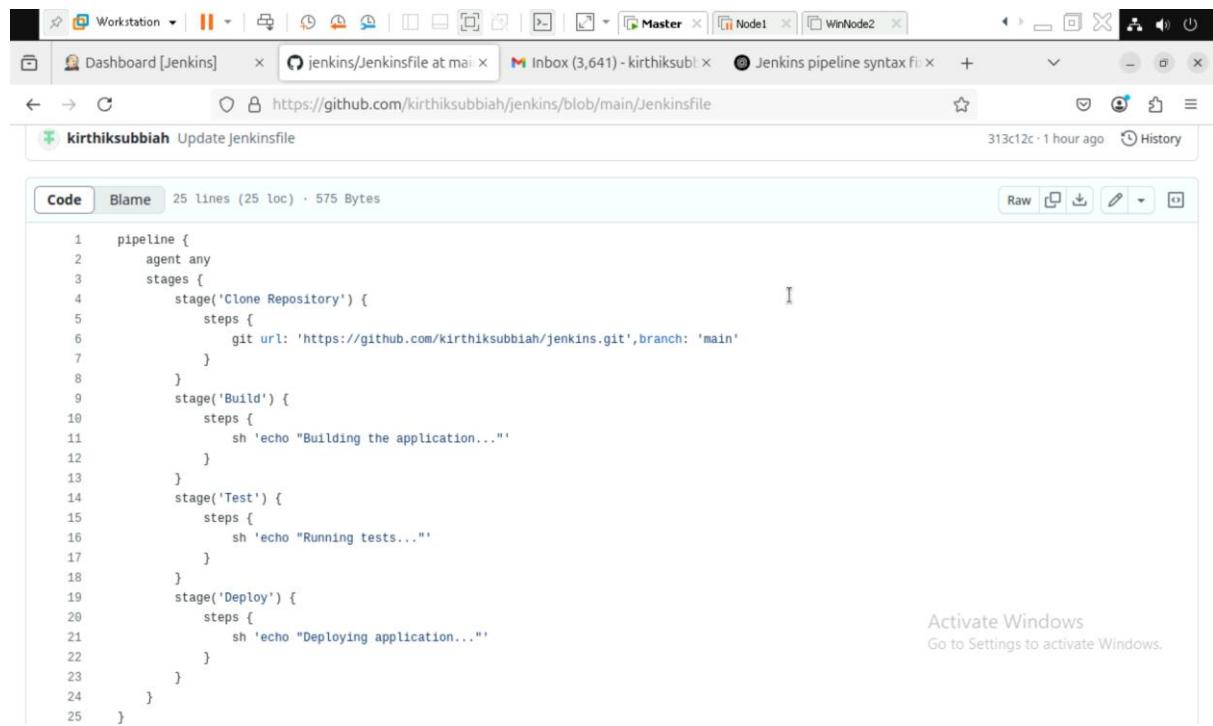
1. In Jenkins, click **New Item** → **Pipeline** → Give it a name → Click **OK**.

2. Scroll down to the **Pipeline** section and select **Pipeline script from SCM**.

3. In the **SCM** field, select **Git**, then enter your **GitHub repository URL**.

4. In the **Branch Specifier**, enter `*/main` or `*/master`.

5. In the **Script Path**, enter `Jenkinsfile` (which we will create in the next step).



The screenshot shows a GitHub code editor window with the URL <https://github.com/kirthiksubbiah/jenkins/blob/main/Jenkinsfile>. The Jenkinsfile contains the following Groovy script:

```
1 pipeline {
2     agent any
3     stages {
4         stage('Clone Repository') {
5             steps {
6                 git url: 'https://github.com/kirthiksubbiah/jenkins.git', branch: 'main'
7             }
8         }
9         stage('Build') {
10            steps {
11                sh 'echo "Building the application..."'
12            }
13        }
14        stage('Test') {
15            steps {
16                sh 'echo "Running tests..."'
17            }
18        }
19        stage('Deploy') {
20            steps {
21                sh 'echo "Deploying application..."'
22            }
23        }
24    }
25 }
```

At the bottom right of the code editor, there is a watermark that says "Activate Windows Go to Settings to activate Windows."

Build History:

The screenshot shows the Jenkins Build History page. On the left, there's a sidebar with links like 'New Item', 'Build History' (which is selected and highlighted in grey), 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', 'My Views', 'Open Blue Ocean', and 'Lockable Resources'. Below the sidebar, there's a 'Build Queue' section stating 'No builds in the queue.' and a 'Build Executor Status' section. The main area is titled 'Build History of Jenkins' and lists four builds for the 'demo' project:

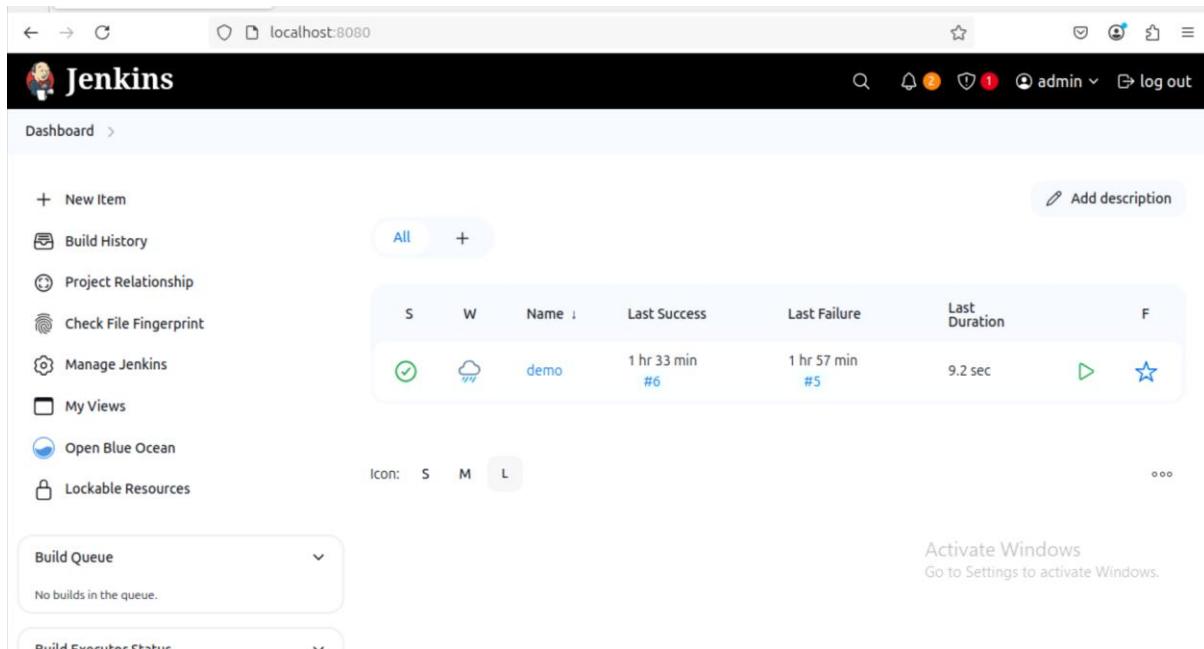
S	Build	Time Since	Status
1	demo #6	1 hr 42 min	back to normal
2	demo #5	2 hr 6 min	broken for a long time
3	demo #4	2 hr 11 min	broken for a long time
4	demo #3	2 hr 13 min	broken since this build

On the right side of the main area, there's a message: 'Activate Windows Go to Settings to activate Windows.' with a small link icon.

Status of Each stages of the pipeline:

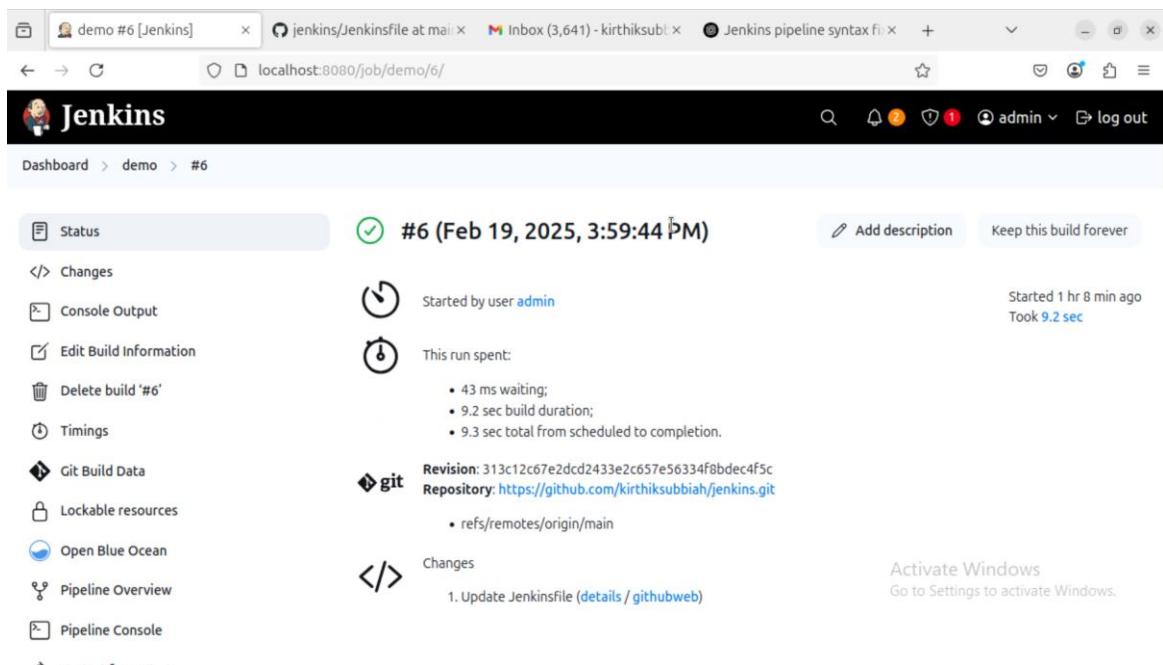
The screenshot shows the Jenkins Pipeline Stage View for the 'demo' job. On the left, there's a sidebar with links like 'Status' (selected and highlighted in grey), 'Changes', 'Build Now', 'Configure', 'Delete Pipeline', 'Full Stage View', 'Favorite', 'Open Blue Ocean', 'Stages', 'Rename', and 'Pipeline Syntax'. Below the sidebar is a 'Builds' section listing builds from today: #7 5:44 PM, #6 3:59 PM, #5 3:35 PM, #4 3:31 PM, and #3 3:28 PM. The main area is titled 'Stage View' and displays a grid of stages for each build. The stages are: Declarative: Checkout SCM, Clone Repository, Build, Test, and Deploy. The 'Build' stage is currently selected. The grid shows run times for each stage across the five builds. A message on the right says: 'Activate Windows Go to Settings to activate Windows.'

Workspace:



The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links like 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', 'My Views', 'Open Blue Ocean', and 'Lockable Resources'. The main area displays a table for jobs. One job, 'demo', is listed with a green checkmark icon. Its details are: Name: demo, Last Success: 1 hr 33 min #6, Last Failure: 1 hr 57 min #5, and Last Duration: 9.2 sec. There are icons for more details and favoriting. Below the table, there are sections for 'Build Queue' (empty) and 'Build Executor Status'.

Status of the Pipeline:



The screenshot shows the Jenkins job details for '#6'. The top navigation bar includes tabs for 'Status', 'Changes', 'Console Output', 'Edit Build Information', 'Delete build #6', 'Timings', 'Git Build Data', 'Lockable resources', 'Open Blue Ocean', 'Pipeline Overview', and 'Pipeline Console'. The main content area shows the build status as successful ('#6 (Feb 19, 2025, 3:59:44 PM)'). It details the start by user 'admin', run duration (1 hr 8 min ago, took 9.2 sec), and spent time (43 ms waiting, 9.2 sec build duration, 9.3 sec total). It also shows the git revision (313c12c67e2dc2433e2c657e56334f8bdec4f5c) and repository (https://github.com/kirthiksubbiah/jenkins.git). A 'Changes' section lists '1. Update Jenkinsfile (details / githubweb)'. A note says 'Activate Windows'.

Step 3: Create a Jenkinsfile (CI/CD Pipeline)

In your GitHub repository, create a new file named **Jenkinsfile** and add the following code:

```

groovy
CopyEdit
pipeline {
    agent any
    stages {
        stage('Clone Repository') {
            steps {
                git 'https://github.com/your-username/your-repo.git'
            }
        }
        stage('Build') {
            steps {
                sh 'echo "Building the application..."'
            }
        }
        stage('Test') {
            steps {
                sh 'echo "Running tests..."'
            }
        }
        stage('Deploy') {
            steps {
                sh 'echo "Deploying application..."'
            }
        }
    }
}

```

Explanation:

- **Stage 1: Clone Repository** → Pulls the latest code from GitHub.
- **Stage 2: Build** → Builds the application (placeholder command).
- **Stage 3: Test** → Runs tests (placeholder command).
- **Stage 4: Deploy** → Deploys the app (placeholder command).

Step 4: Build and Run the Pipeline

1. Go back to **Jenkins Dashboard** → Click on your job.
2. Click **Build Now**.
3. Click **Console Output** to see the progress.

If everything is successful, you will see all **stages completed** in the output.

Step 5: Deploy with Docker (Optional)

If you want to deploy your application as a **Docker container**, modify your **Jenkinsfile** like this:

```
groovy
CopyEdit
pipeline {
    agent any
    environment {
        IMAGE_NAME = 'your-dockerhub-username/my-app'
    }
    stages {
        stage('Clone Repository') {
            steps {
                git 'https://github.com/your-username/your-repo.git'
            }
        }
        stage('Build Docker Image') {
            steps {
                sh 'docker build -t $IMAGE_NAME .'
            }
        }
        stage('Push Docker Image') {
            steps {
                withDockerRegistry([credentialsId: 'docker-hub-credentials']) {
                    sh 'docker push $IMAGE_NAME'
                }
            }
        }
        stage('Deploy') {
            steps {
                sh 'docker run -d -p 8080:80 $IMAGE_NAME'
            }
        }
    }
}
```

Explanation:

- **Builds a Docker image** of your application.
 - **Pushes it to Docker Hub** (requires credentials setup in Jenkins).
 - **Deploys the container** on a local machine or server.
-
- You have successfully set up **Jenkins CI/CD** to **build, test, and deploy** a simple app!
 - If using **Docker**, your app is now running on **http://localhost:8080**.