

ASSIGNMENT 2

CI/CD Pipeline Setup Using Jenkins

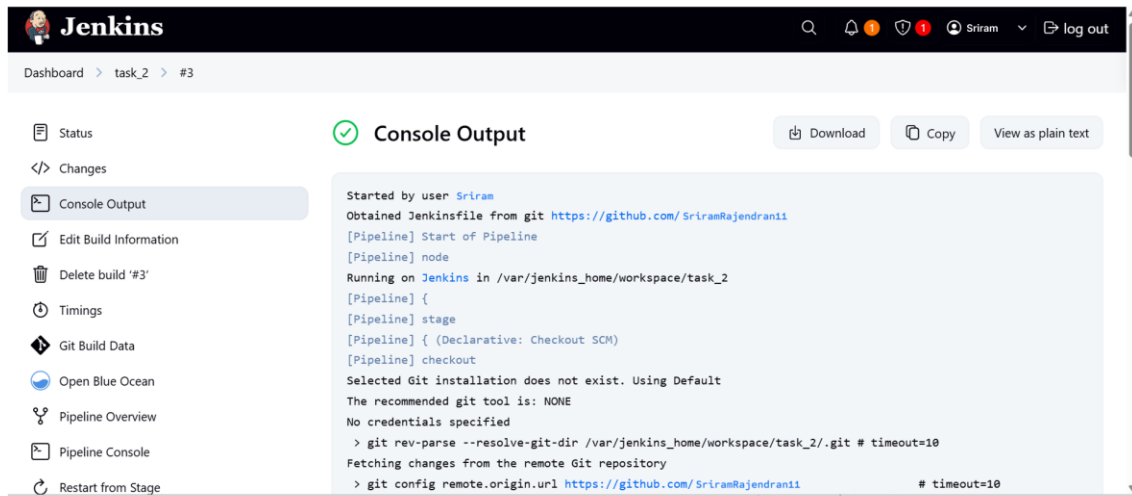
Task Description:

This task involves setting up a Continuous Integration and Continuous Deployment (CI/CD) pipeline using Jenkins to automate the build, test, and push process of a containerized web application. The pipeline integrates GitHub for source code management, Docker for containerization, and a container registry (such as Docker Hub) for storing the container image.

Steps to Set Up the CI/CD Pipeline:

1. **Install Jenkins:**
 - Ensure Jenkins is running and install necessary plugins (Git, Docker).
2. **Create a New Pipeline Job:**
 - Create a new Pipeline job in Jenkins to automate the build, test, and deployment process.
3. **Configure the GitHub Repository:**
 - Link your GitHub repository to Jenkins to automatically pull the latest code.
4. **Write the Pipeline Script (Jenkinsfile):**
 - Define stages in the Jenkinsfile:
 - **Clone Repository:** Pull the latest code from GitHub.
 - **Build Docker Image:** Build the image using Docker.
 - **Push Docker Image:** Push the image to a container registry (e.g., Docker Hub).
5. **Add Docker Credentials (if needed):**
 - Add Docker credentials to Jenkins if using a private registry for authentication.
6. **Trigger the Pipeline:**
 - Trigger the pipeline manually or set up GitHub webhooks for automatic triggers on code changes.

OUTPUT:



The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search icon, a notification bell, a shield icon, a user profile for 'Sriram', and a 'log out' button. The breadcrumb trail indicates the current location: 'Dashboard > task_2 > #3'. On the left sidebar, several menu items are listed: 'Status', 'Changes', 'Console Output' (which is highlighted), 'Edit Build Information', 'Delete build '#3'', 'Timings', 'Git Build Data', 'Open Blue Ocean', 'Pipeline Overview', 'Pipeline Console', and 'Restart from Stage'. The main content area is titled 'Console Output' with a green checkmark icon. To the right of the title are three buttons: 'Download', 'Copy', and 'View as plain text'. The console output itself is a text log showing the pipeline's execution. It starts with 'Started by user Sriram', followed by 'Obtained Jenkinsfile from git https://github.com/SriramRajendran11'. The log then shows '[Pipeline] Start of Pipeline', '[Pipeline] node', and 'Running on Jenkins in /var/jenkins_home/workspace/task_2'. A YAML-style block '[Pipeline] { [Pipeline] stage [Pipeline] { (Declarative: Checkout SCM) [Pipeline] checkout' is shown. Below this, it says 'Selected Git installation does not exist. Using Default' and 'The recommended git tool is: NONE'. It then notes 'No credentials specified' and shows a terminal command: '> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/task_2/.git # timeout=10'. This is followed by 'Fetching changes from the remote Git repository' and another terminal command: '> git config remote.origin.url https://github.com/SriramRajendran11 # timeout=10'. The log ends with a downward arrow icon.

Jenkins

Dashboard > task_2 > #3

Status

Changes

Console Output

Edit Build Information

Delete build '#3'

Timings

Git Build Data

Open Blue Ocean

Pipeline Overview

Pipeline Console

Restart from Stage

Console Output

Download Copy View as plain text

```
Started by user Sriram
Obtained Jenkinsfile from git https://github.com/SriramRajendran11
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/task_2
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/task_2/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/SriramRajendran11 # timeout=10
```