

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).

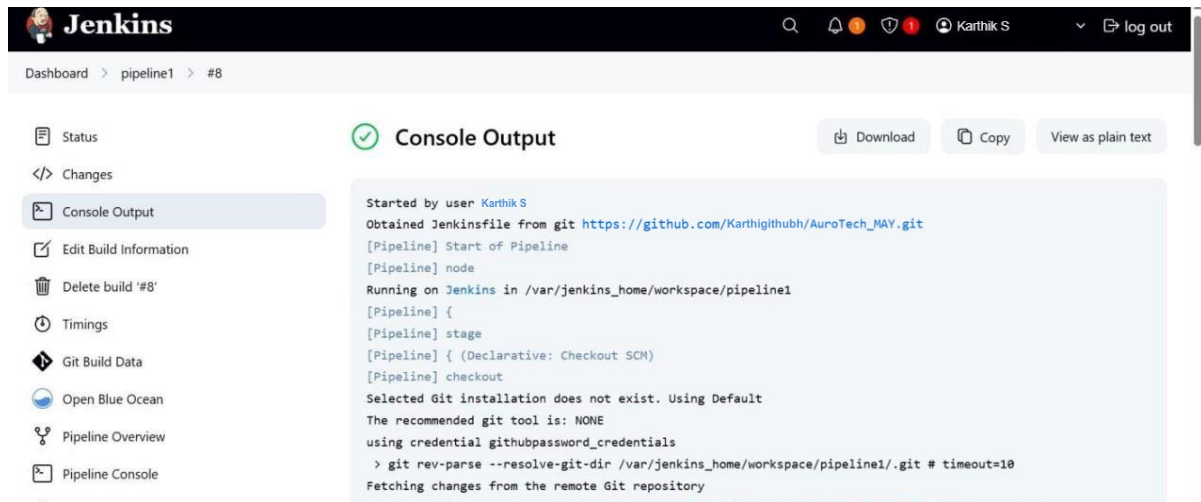
5. **Add Docker Credentials (if needed):**

- 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, notification icons, a user profile for 'Karthik S', and a 'log out' button. The breadcrumb trail is 'Dashboard > pipeline1 > #8'. On the left sidebar, the 'Console Output' tab is selected. The main content area displays the console output for build #8 of pipeline1, which is in a 'Completed' state (indicated by a green checkmark). The output text shows the build was started by user 'Karthik S', the Jenkinsfile was obtained from a GitHub repository, and the pipeline executed successfully. The output is displayed in a light blue box with a monospaced font. Action buttons for 'Download', 'Copy', and 'View as plain text' are visible at the top right of the output area.

Dashboard > pipeline1 > #8

Console Output

Started by user [Karthik S](#)
Obtained Jenkinsfile from git https://github.com/Karthigithub/AuroTech_MAY.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/pipeline1
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential githubpassword_credentials
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/pipeline1/.git # timeout=10
Fetching changes from the remote Git repository



This screenshot shows the continuation of the Jenkins console output for build #8 of pipeline1. The output text continues from the previous block, showing the end of the pipeline stage and the final success message. The output is displayed in a light blue box with a monospaced font. The breadcrumb trail remains 'Dashboard > pipeline1 > #8'. At the bottom right of the interface, the text 'REST API' and 'Jenkins 2.498' is visible.

Dashboard > pipeline1 > #8

[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Pipeline succeeded! The application is running on localhost:3000.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

REST API Jenkins 2.498