

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).

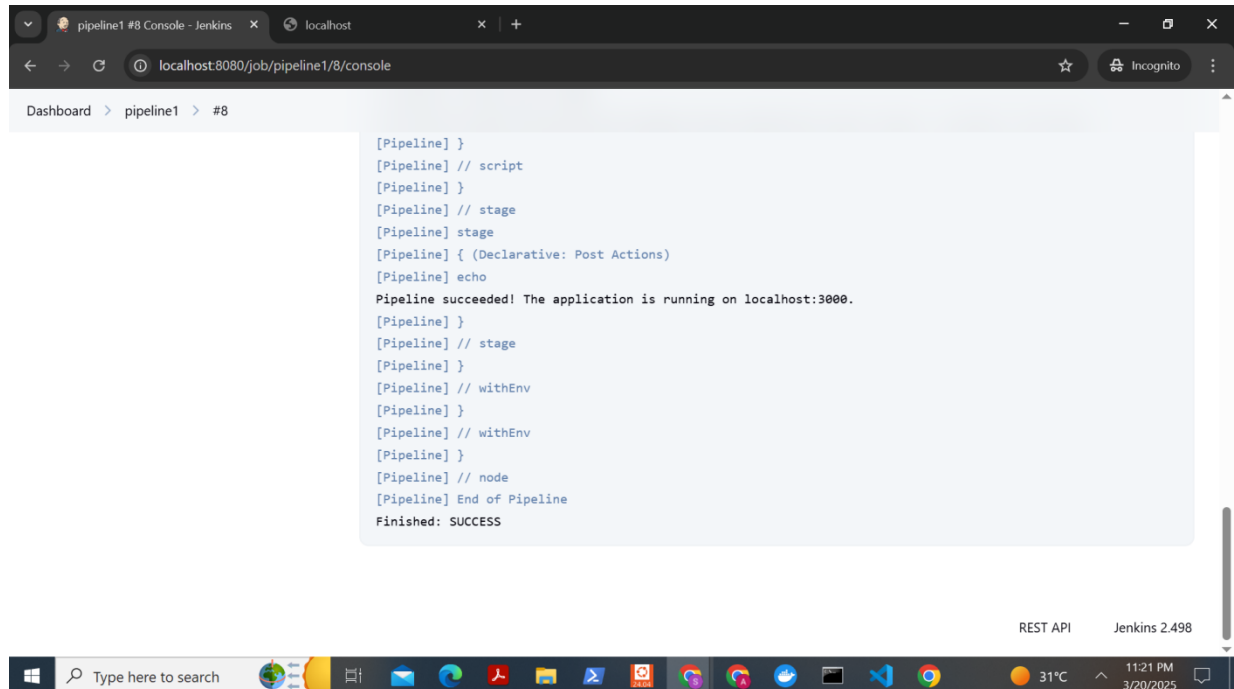
5. **Add Docker Credentials (if needed):**

- 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.

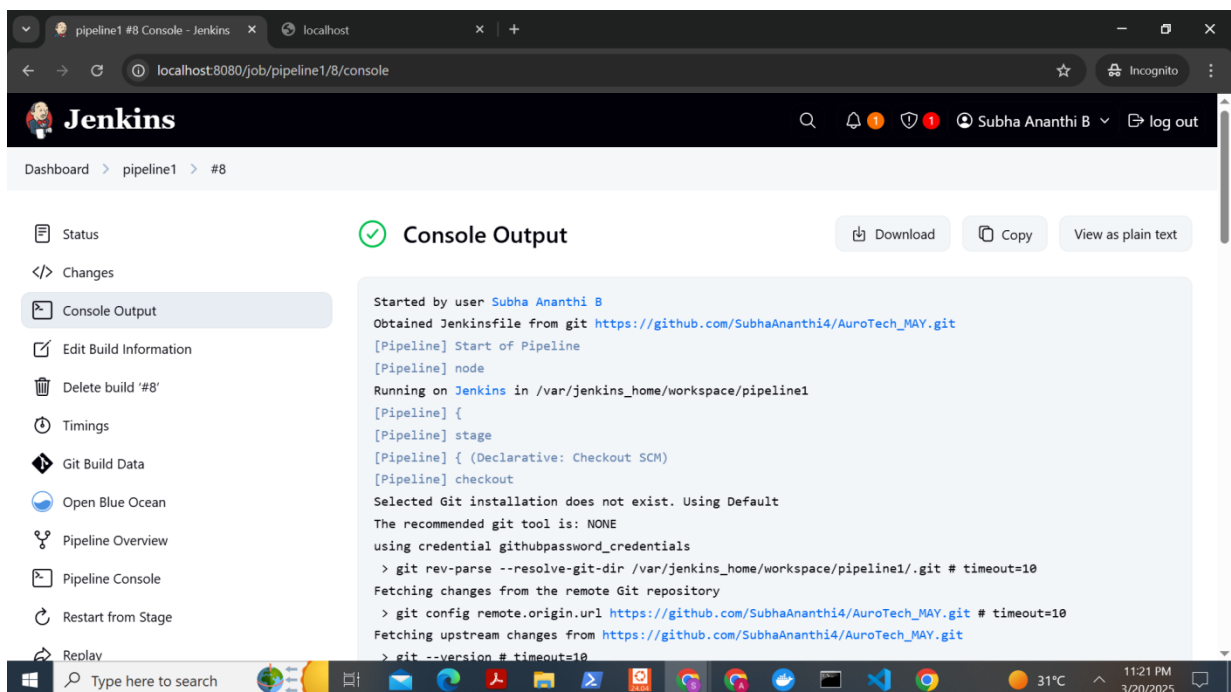
OUTPUT:



A screenshot of a web browser showing the Jenkins console output for a pipeline named 'pipeline1' at build #8. The browser's address bar shows 'localhost:8080/job/pipeline1/8/console'. The console output is as follows:

```
[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
```

The bottom of the screenshot shows the Windows taskbar with various application icons and the system clock indicating 11:21 PM on 3/20/2025.



A screenshot of the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search icon, a notification bell, and a user profile for 'Subha Ananthi B' with a 'log out' button. The breadcrumb trail shows 'Dashboard > pipeline1 > #8'. On the left sidebar, the 'Console Output' tab is selected. The main content area, titled 'Console Output' with a green checkmark icon, displays the following log:

```
Started by user Subha Ananthi B
Obtained Jenkinsfile from git https://github.com/SubhaAnanthi4/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
> git config remote.origin.url https://github.com/SubhaAnanthi4/AuroTech_MAY.git # timeout=10
Fetching upstream changes from https://github.com/SubhaAnanthi4/AuroTech_MAY.git
> git --version # timeout=10
```

The bottom of the screenshot shows the Windows taskbar with various application icons and the system clock indicating 11:21 PM on 3/20/2025.