

# Automate docker built and push using Jenkinsfile

## 1) Setup a Simple Flask App

### Project Structure

my-flask-app

|— app.py

|— requirements.txt

|— Dockerfile

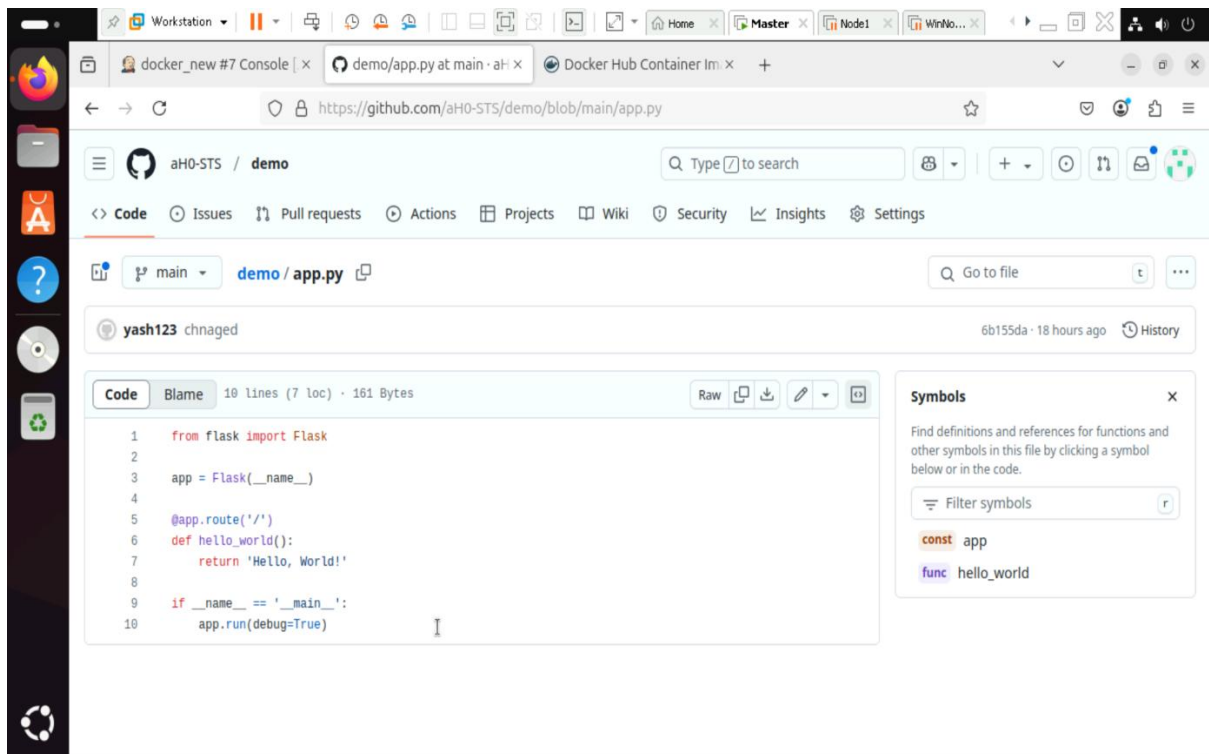
|— Jenkinsfile

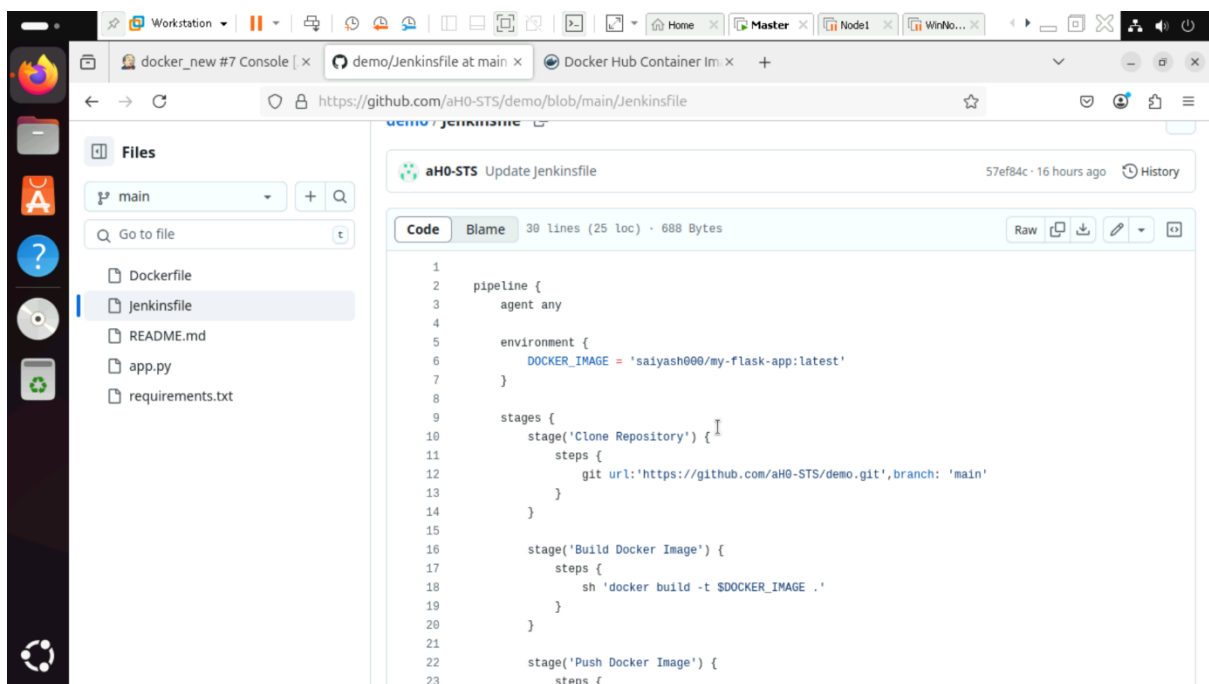
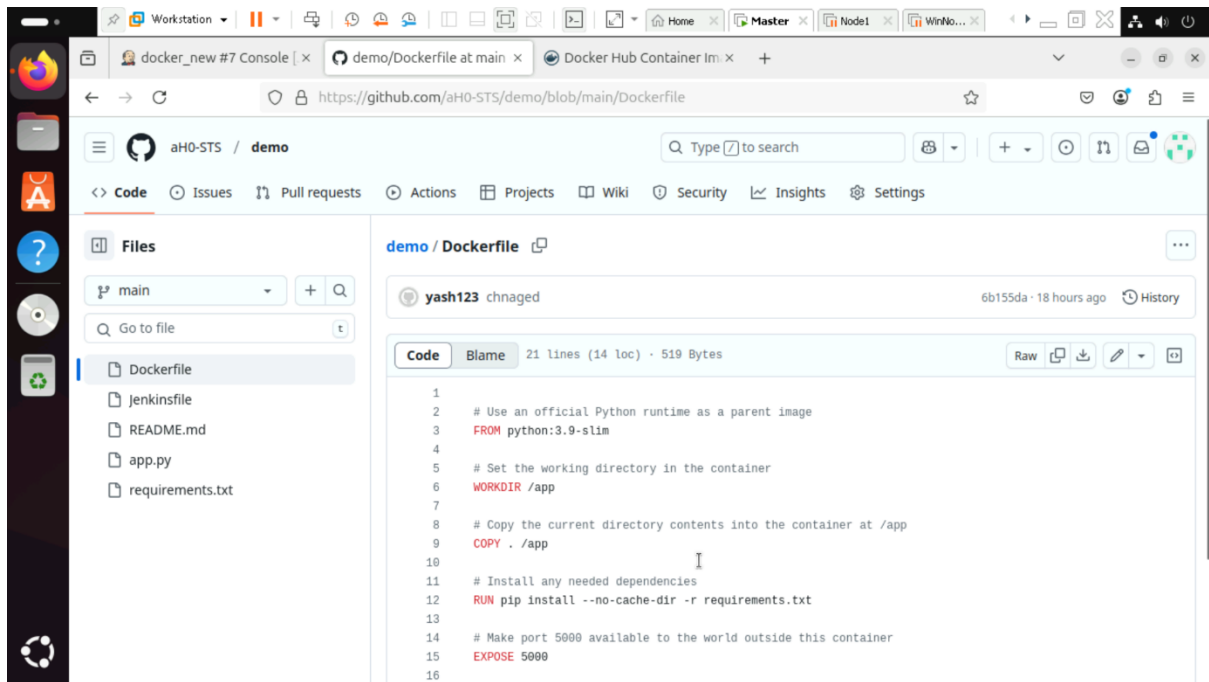
**app.py:** The main Flask application file.

**requirements.txt:** List of dependencies (Flask and others).

**Dockerfile :** Defines the Docker image for the Flask app.

**Jenkinsfile :** Contains the Jenkins pipeline configuration.

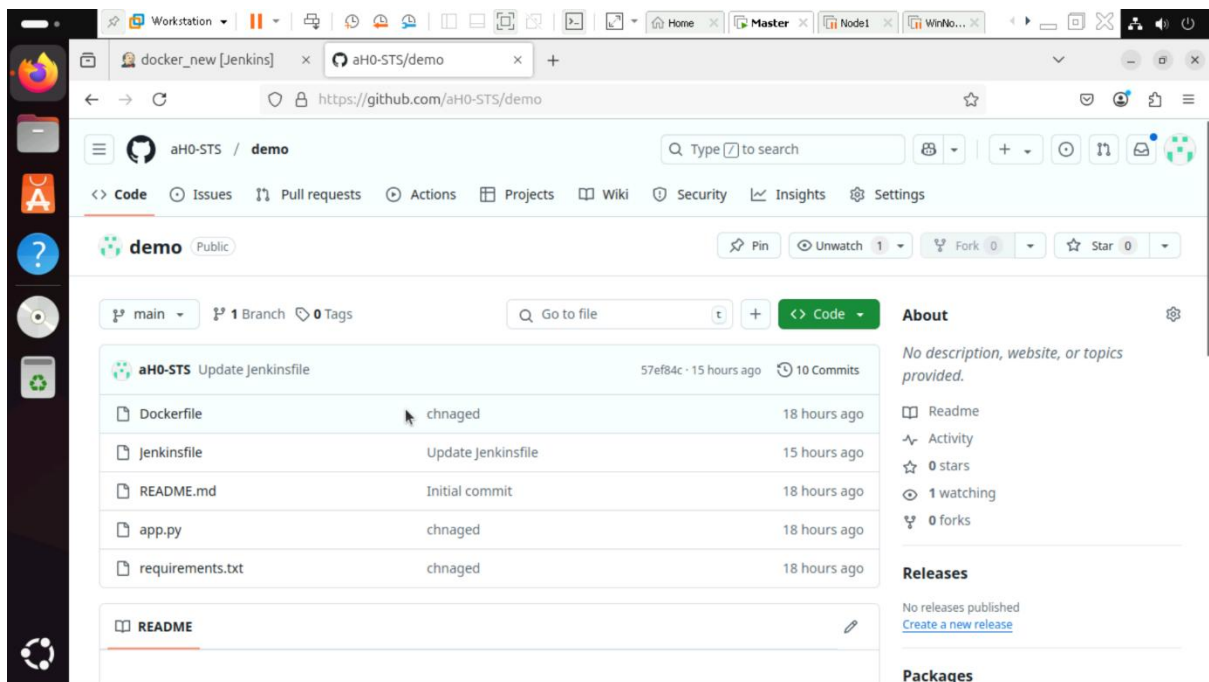




Github link for the code: [aH0-STS/demo](https://github.com/aH0-STS/demo)

## 2. Push the Code to GitHub:

- Make sure you have a GitHub repository created for the project.
- Push all the files (app.py, requirements.txt, Dockerfile, Jenkinsfile) to the GitHub



### 3. Configure Docker Hub Credentials in Jenkins:

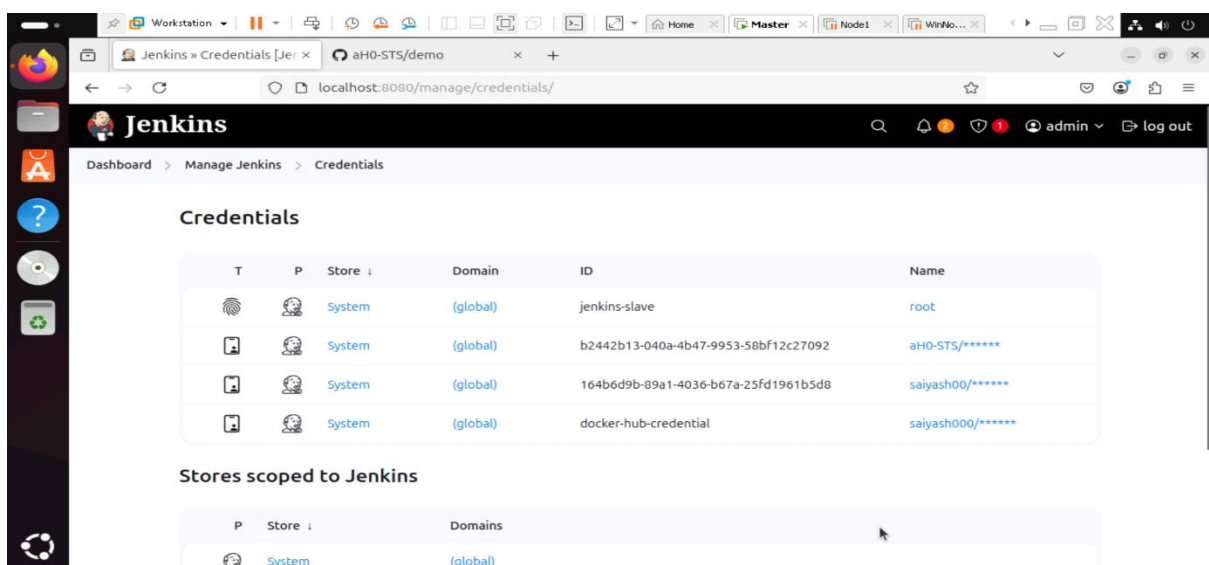
- Go to Jenkins > Manage Jenkins > Manage Credentials.

- Add new credentials:

o Username: Your Docker Hub username.

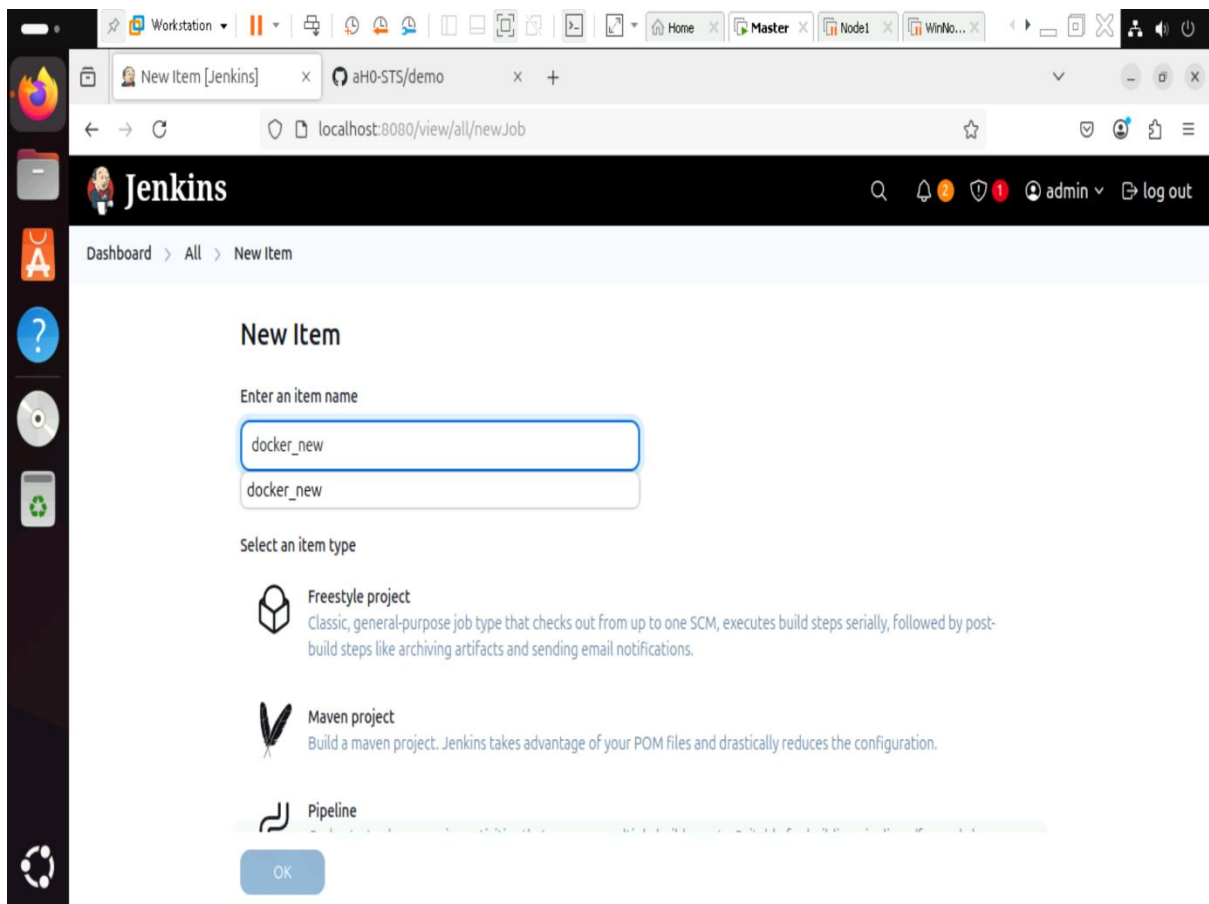
o Password: Your Docker Hub password (or token).

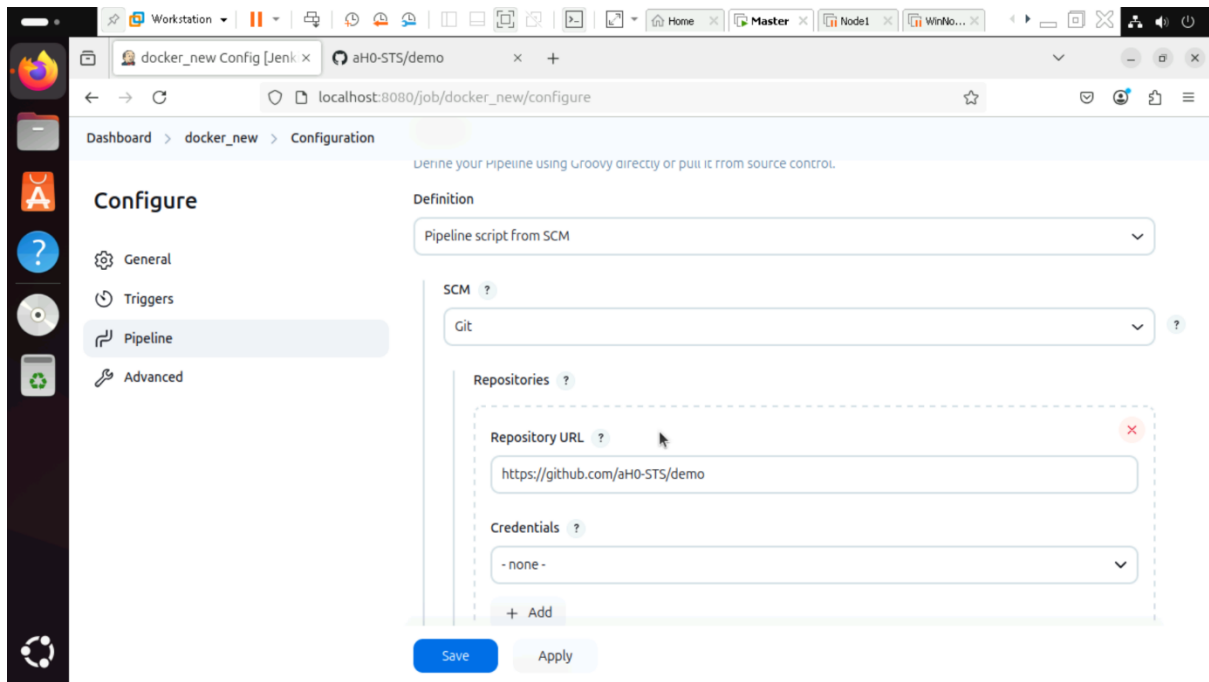
o ID: Name it something like dockerhub-creds (the same name used in the Jenkinsfile).



#### 4. Create a New Pipeline in Jenkins

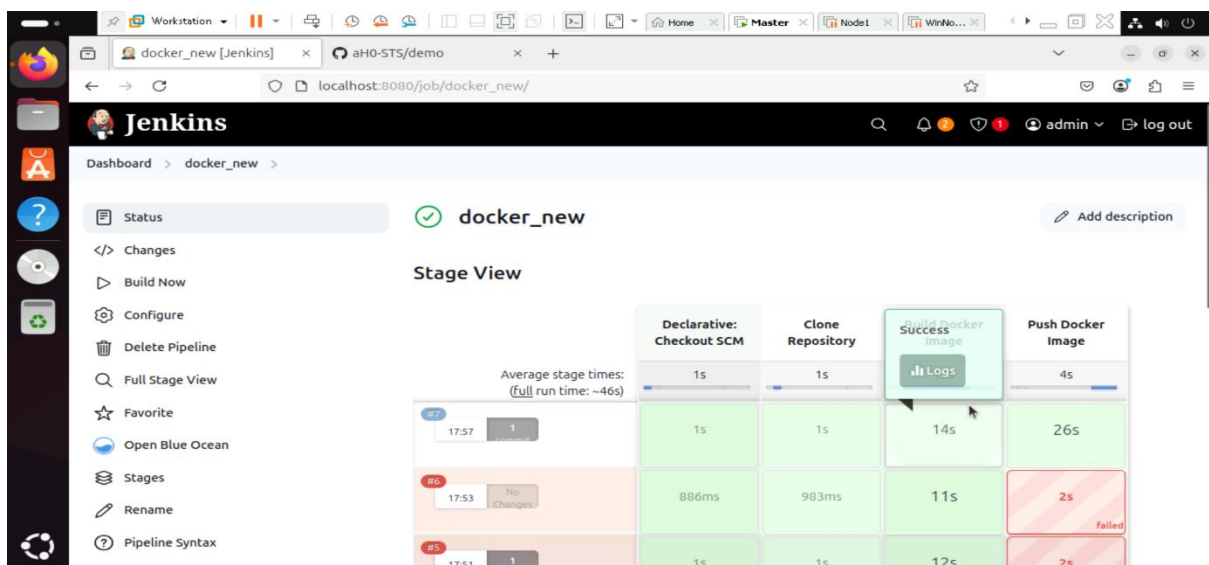
- In Jenkins, click New Item > Pipeline.
- Enter a name for the pipeline.
- Under Pipeline Definition, select Pipeline script from SCM.
  - o Select Git as the SCM.
  - o Enter the GitHub repository URL (<https://github.com/your-username/my-flask-app.git>).
  - o Set the branch (typically master or main).
- Click Save.

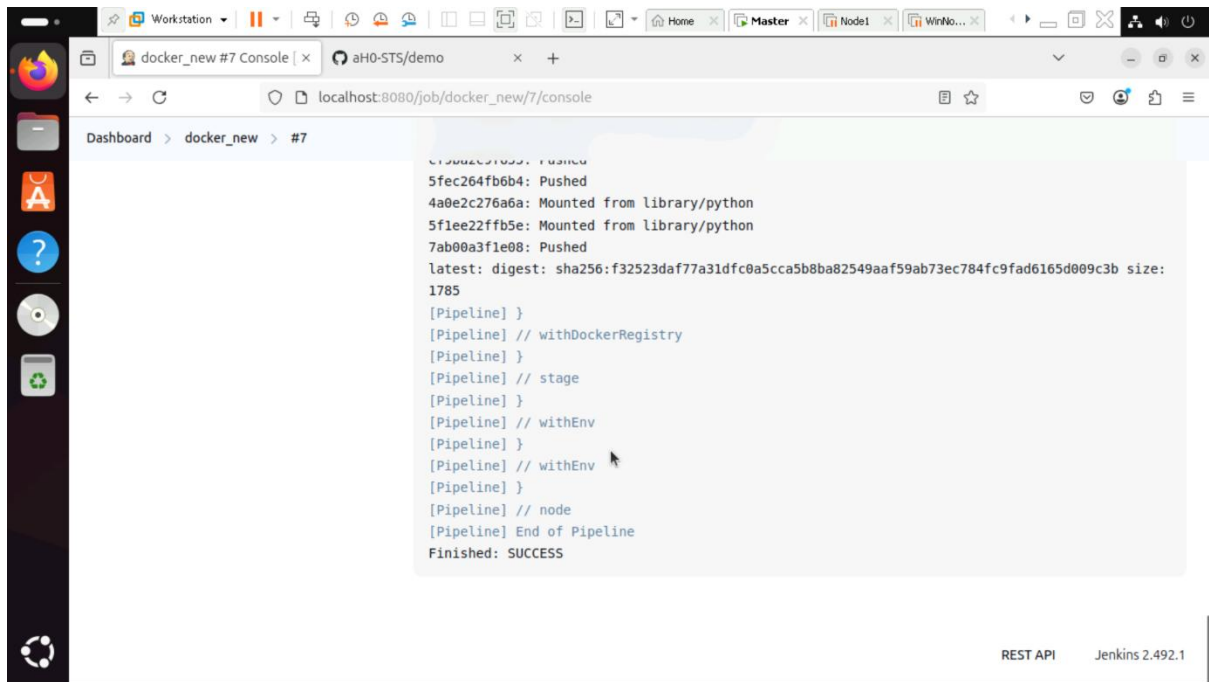




## 5. Click Build Now

- Click Build Now in Jenkins to trigger the build.
- Jenkins will:
  - o Checkout the code from GitHub.
  - o Build the Docker image.
  - o Push the image to Docker Hub.





## 6. Verify Docker Image on Docker Hub

- After the build finishes, log into your Docker Hub account.
- You should see the my-flask-app image under Repositories with the latest tag.

