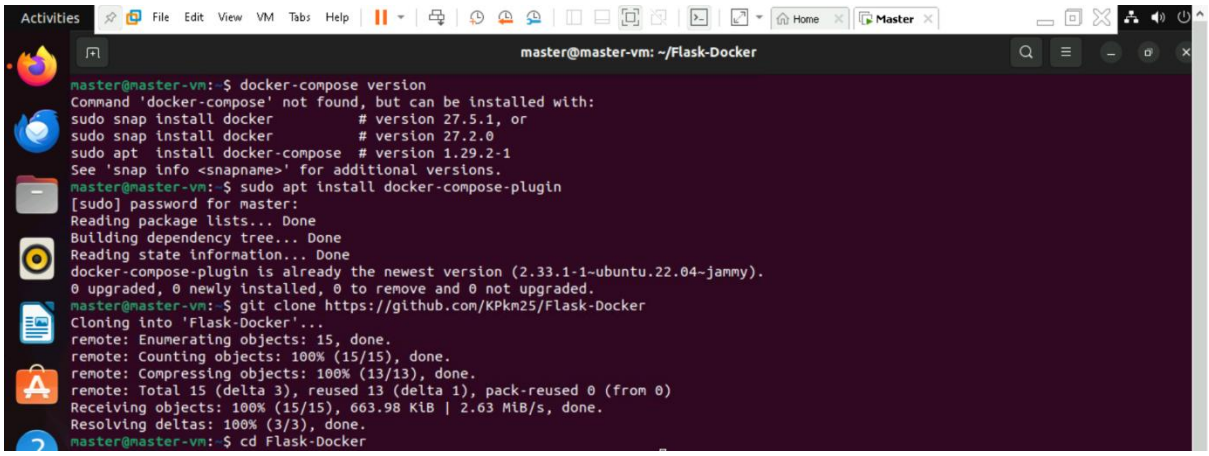
1. **Multi-Container Flask Application with PostgreSQL Using Docker Compose**

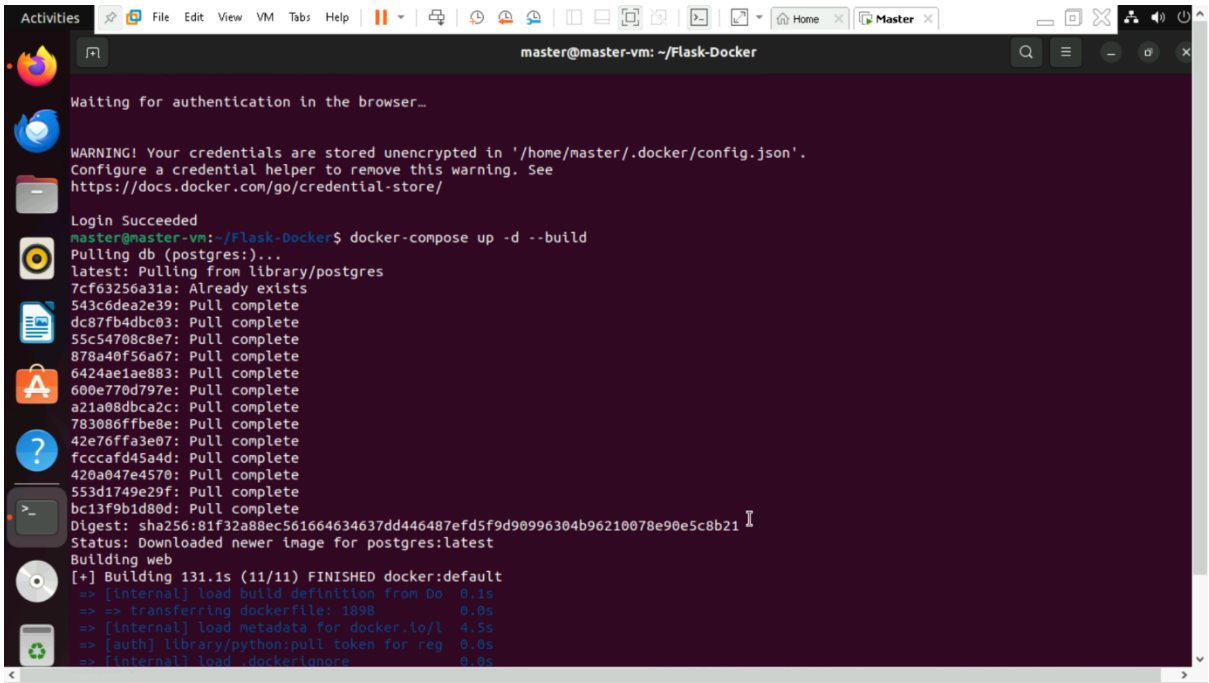
Step 1: Download Docker .

Sudo apt install docker-compose-plugin

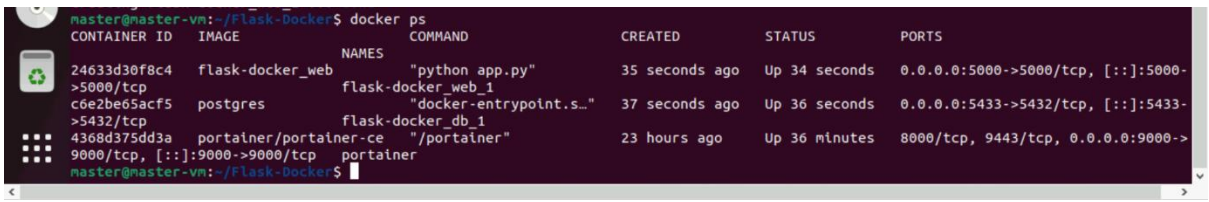
Step 2: Clone the git repository and move to directory cd Flask-Docker



Step 3: Before build the containers login to docker hub using command *"docker login"*. Build and start the containers using *docker-compose up -d –build.*

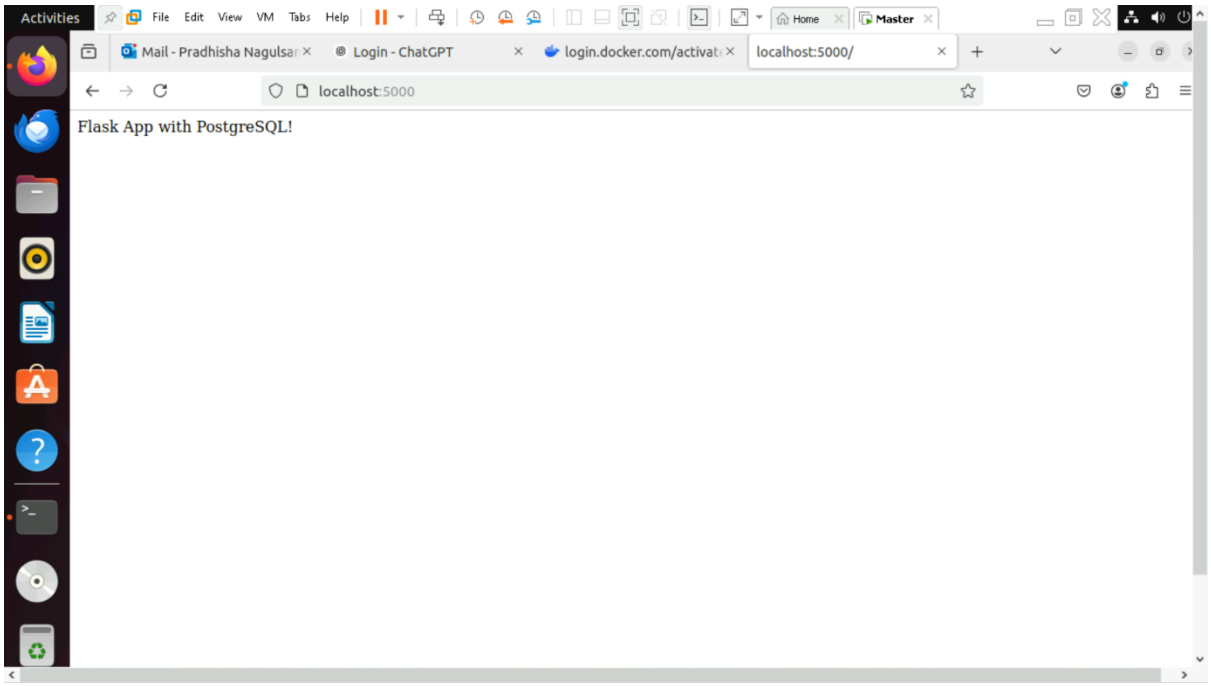


Step 4: Verify the running containers using *"docker ps"*. It used to verify whether web and db is there or not.

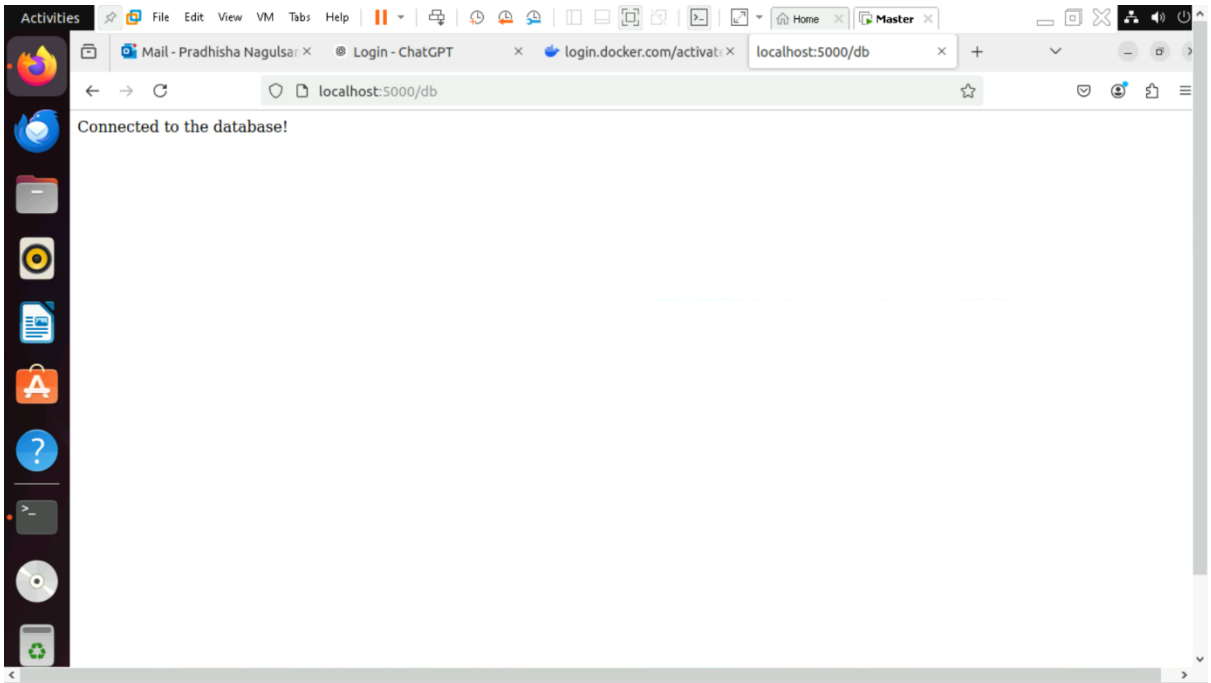


Step 5: Test the application.

http://localhost:5000/ → Should return "Flask App with PostgreSQL!"



http://localhost:5000/db → Should confirm database connection.



2. **Jenkins + Docker Pipeline Project Documentation**

Step 1: Install Docker on Jenkins Server

1. **Update system packages and install Docker:**

   sudo apt update

   sudo apt install docker.io -y

2. **Start and enable Docker:**

   sudo systemctl start docker

   sudo systemctl enable docker

3. **Add Jenkins user to Docker group (to allow Jenkins to run Docker commands):**

   sudo usermod -aG docker jenkins

4. **Restart Jenkins to apply changes:**

   sudo systemctl restart jenkins

5. **Verify Docker installation:**

   docker –version

Step 2: Enable Password Authentication (If Needed)

If SSH key authentication is not set up, enable password login:

1. **Connect to the remote server and edit the SSH configuration file:**

   sudo nano /etc/ssh/sshd_config

2. **Modify these lines:**

   PasswordAuthentication yes

   PermitRootLogin yes

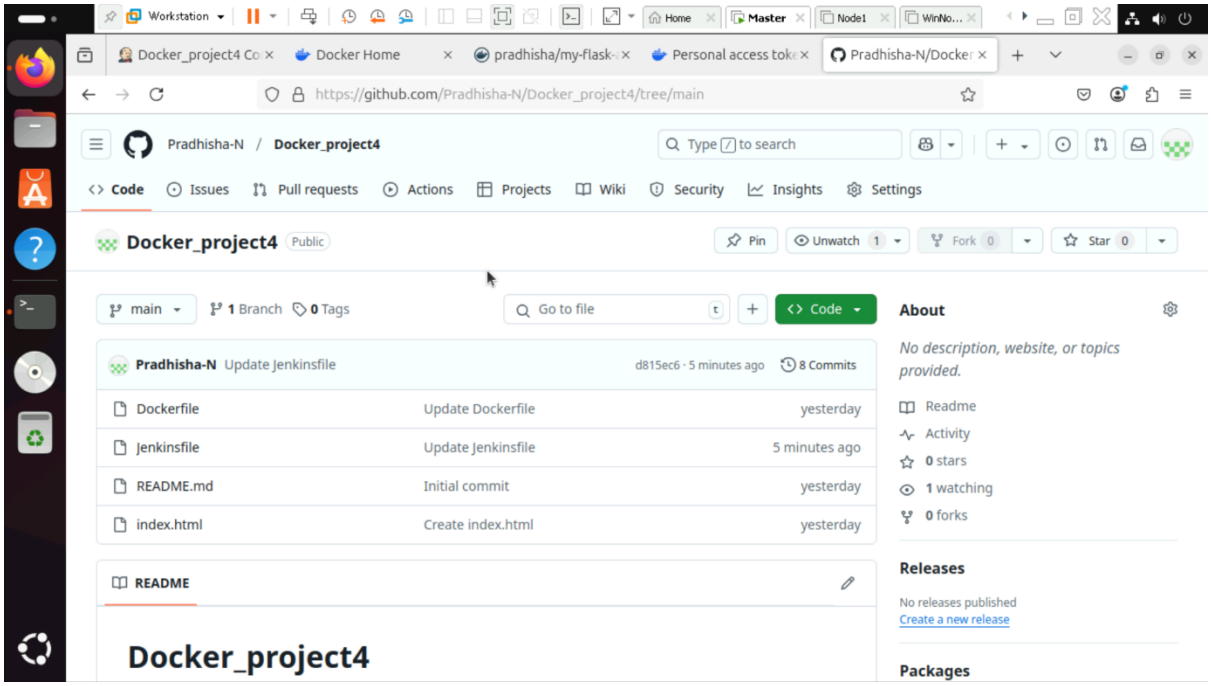
3. **Save the file and restart SSH:**

   sudo systemctl restart ssh

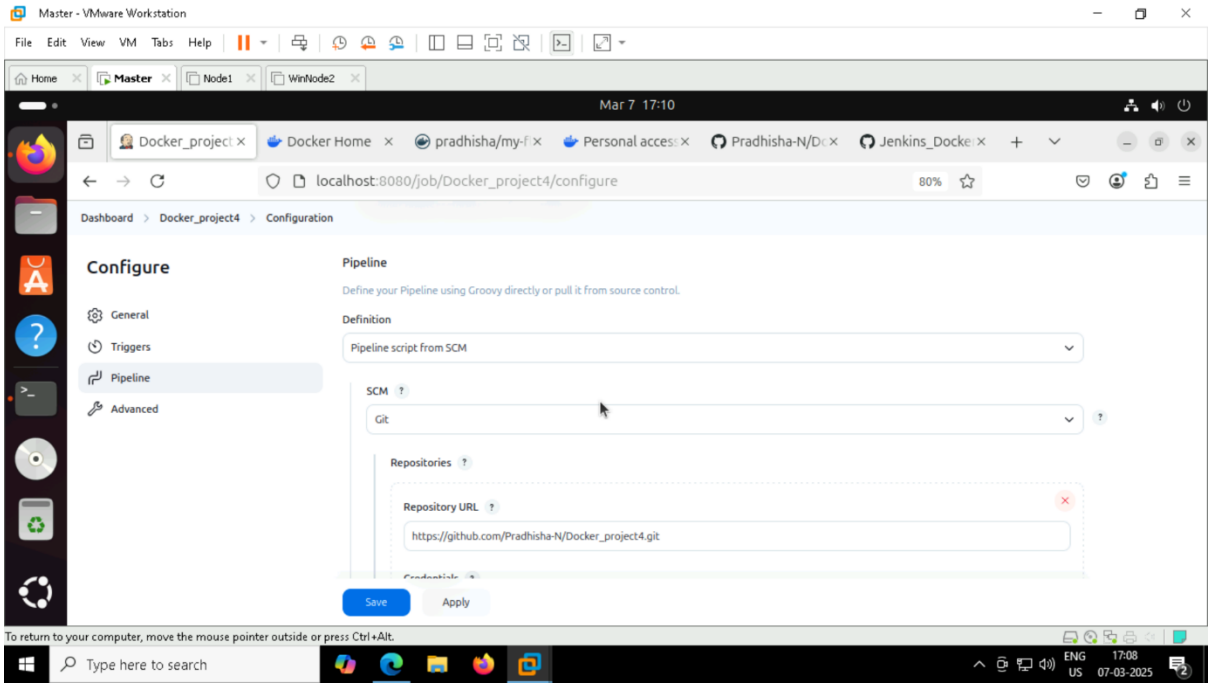
4. **Test SSH login:**

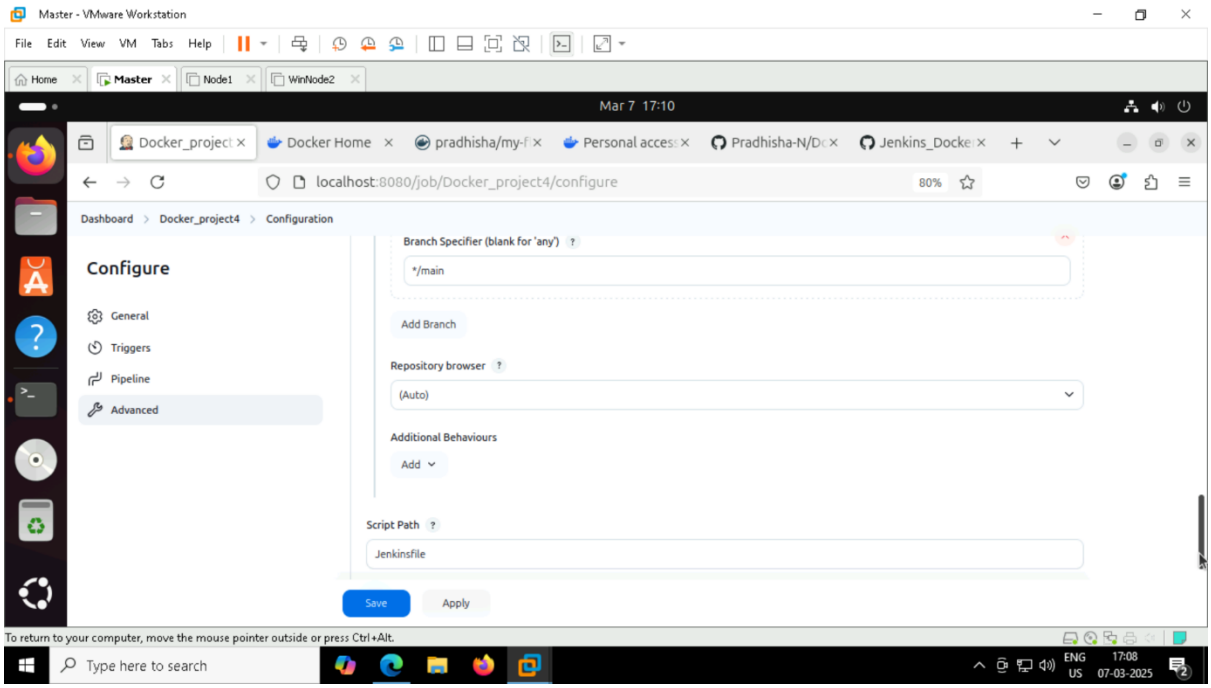
   ssh master@192.168.203.128

Step 3: Create a Simple Web Application

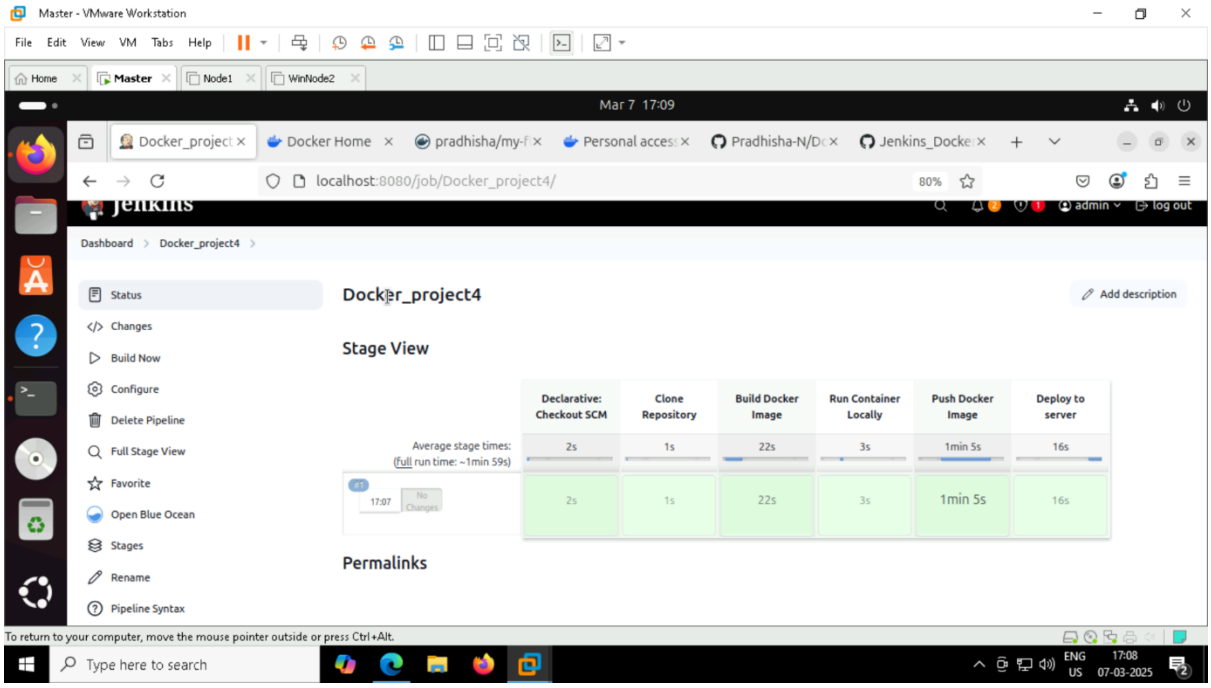


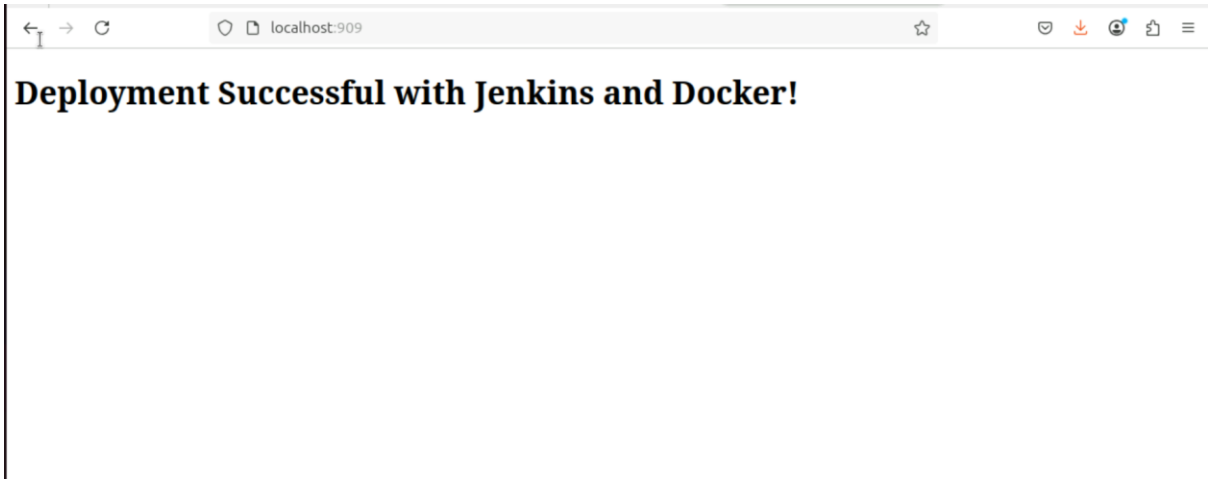
Step 4: Create a Jenkins Pipeline





Step 5: Run the Jenkins Pipeline




localhost:909
Deployment Successful with Jenkins and Docker!