**Task 6: CI/CD Pipeline with Jenkins, Docker & AWS**

**Name -Ujjawal Rawat**

**Date-07-10-2025**

**Objective:**  
Automate deployment of an application from GitHub to Docker Hub using Jenkins running on AWS EC2.

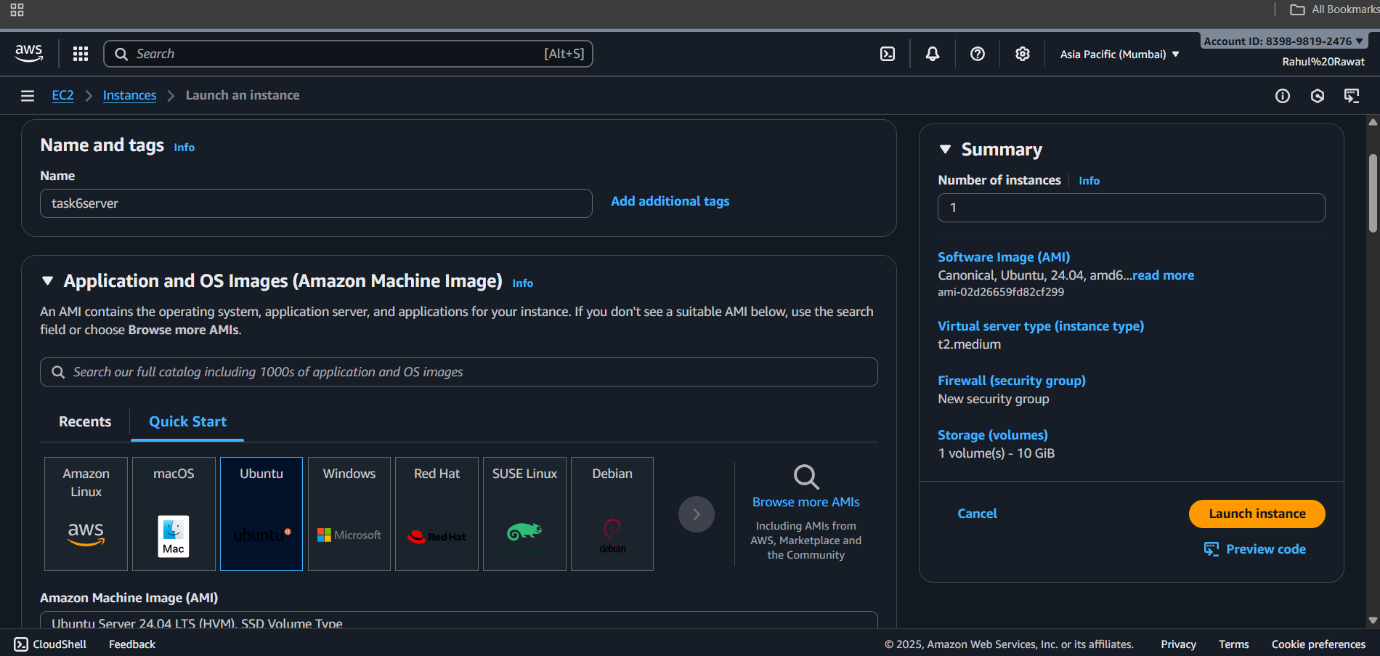
**Step 1: Launch AWS EC2 Instance**

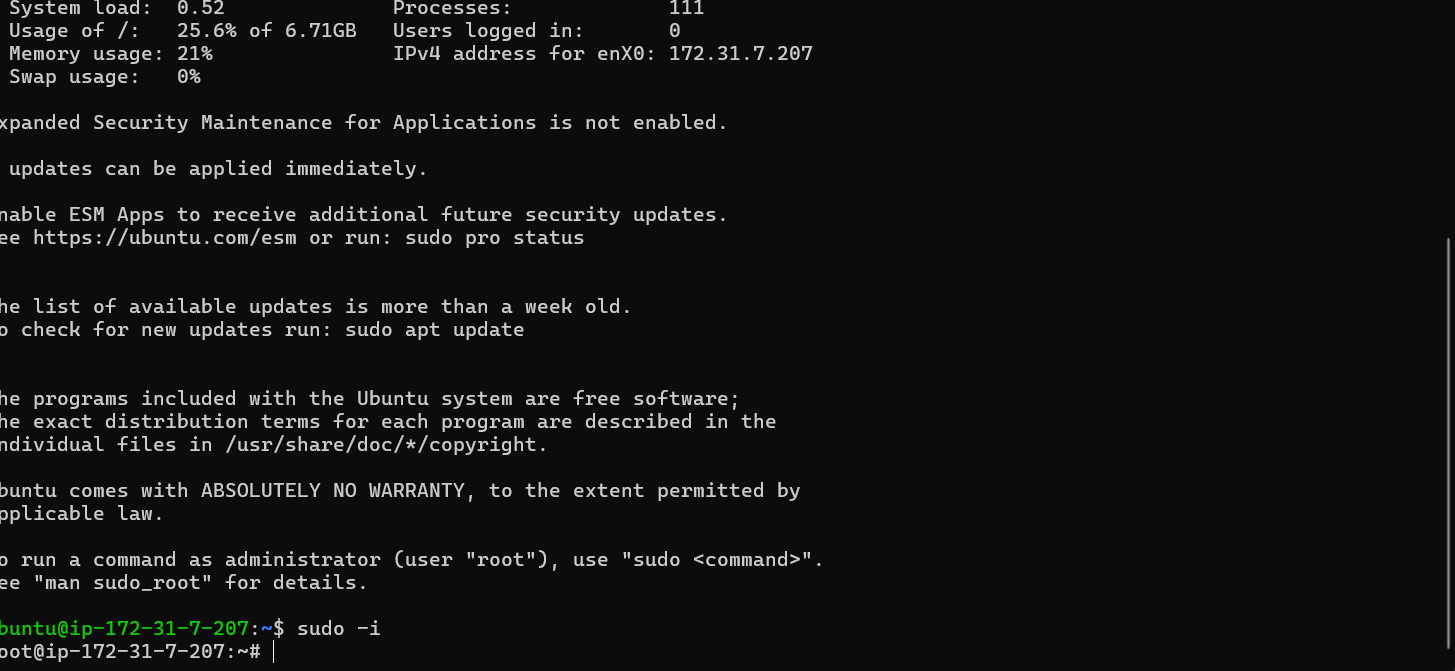
1. **Login to AWS Console → EC2 → Launch Instance**
2. Choose **Ubuntu Server 22.04 LTS**
3. Instance Type: **t2.micro** (free tier)
4. Configure Security Group:
   * **Port 22:** SSH
   * **Port 8080:** Jenkins
   * **Port 80:** Optional, for web app
5. Launch instance and download **key pair (.pem)**

**Commands to SSH into instance:**

chmod 400 your-key.pem

ssh -i "your-key.pem" ubuntu@<EC2\_PUBLIC\_IP>





**Step 2: Install Required Tools on EC2**

1. **Update packages**

sudo apt update -y

sudo apt upgrade -y

1. **Install Java (Jenkins requires Java)**

sudo apt install openjdk-11-jdk -y

java -version

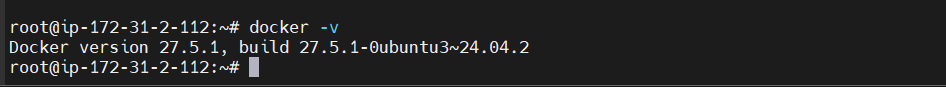
1. **Install Docker**

sudo apt install docker.io -y

sudo systemctl start docker

sudo systemctl enable docker

sudo usermod -aG docker $USER



1. **Install Jenkins**

wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -

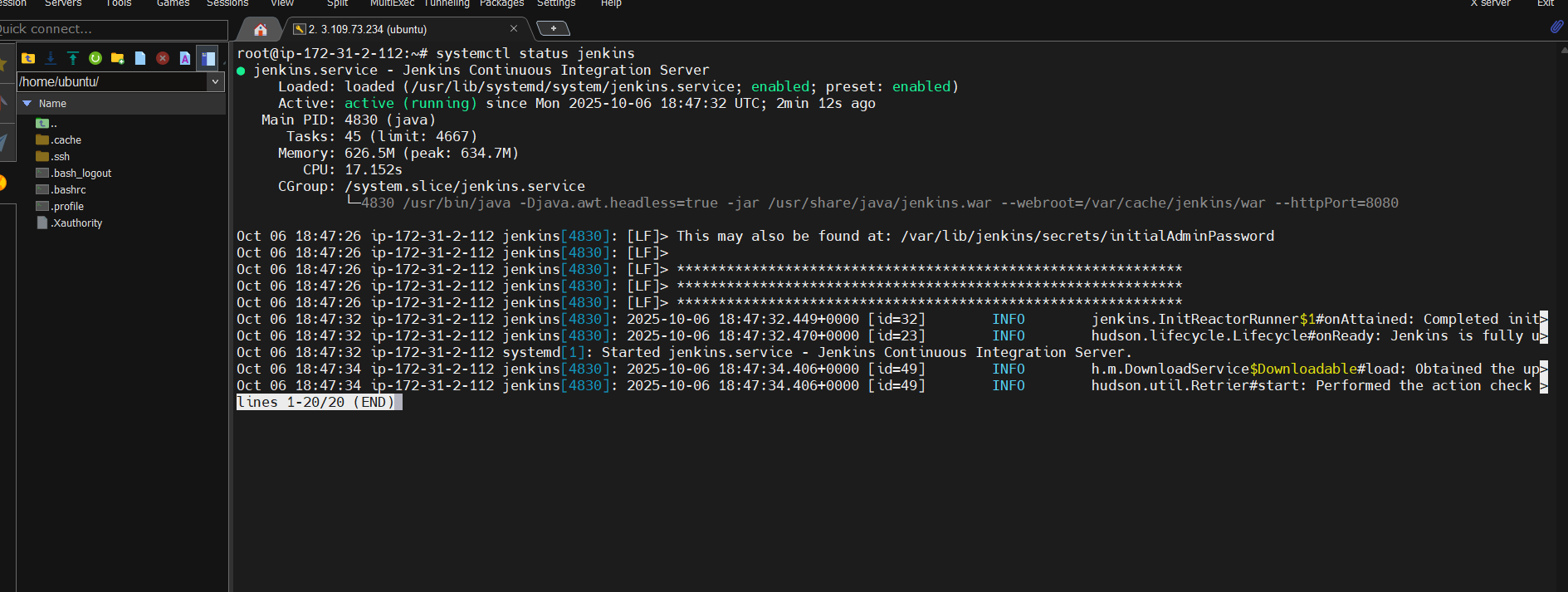
sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

sudo apt update -y

sudo apt install jenkins -y

sudo systemctl start jenkins

sudo systemctl enable Jenkins

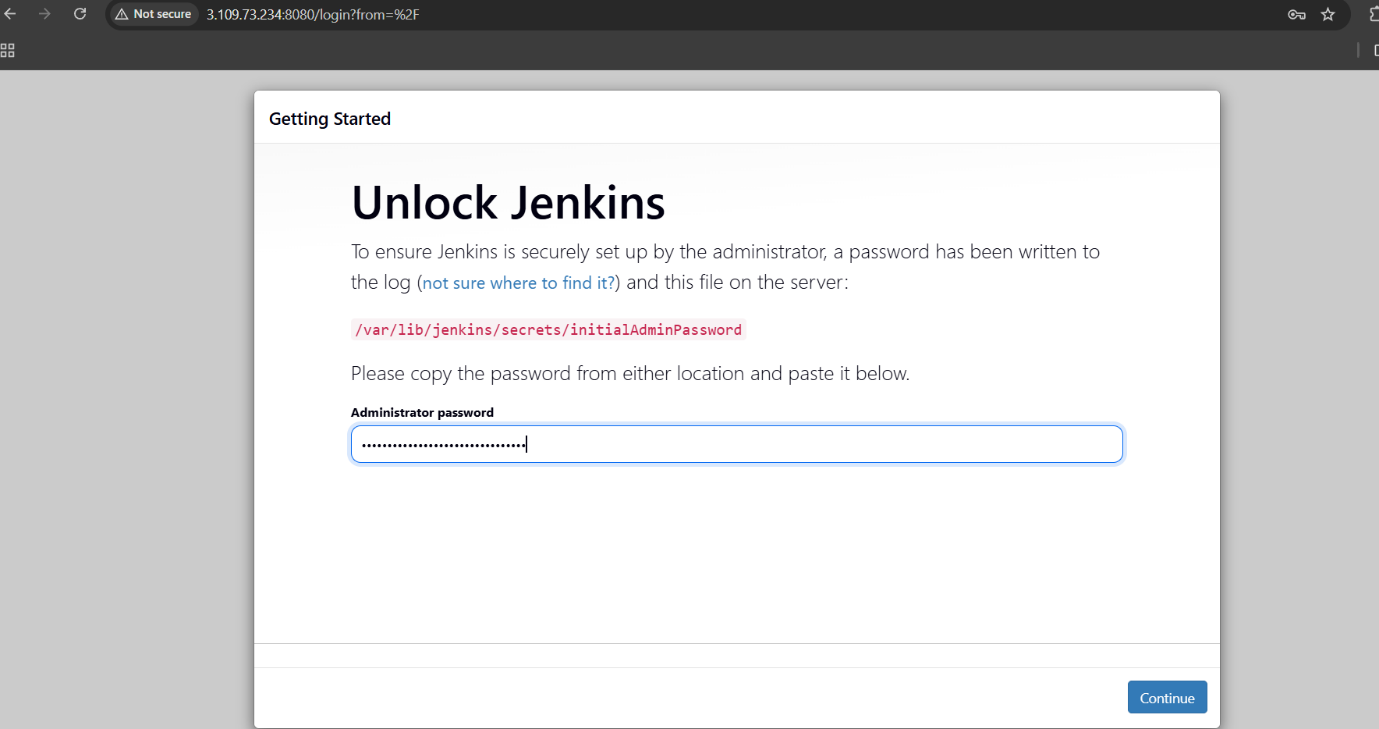


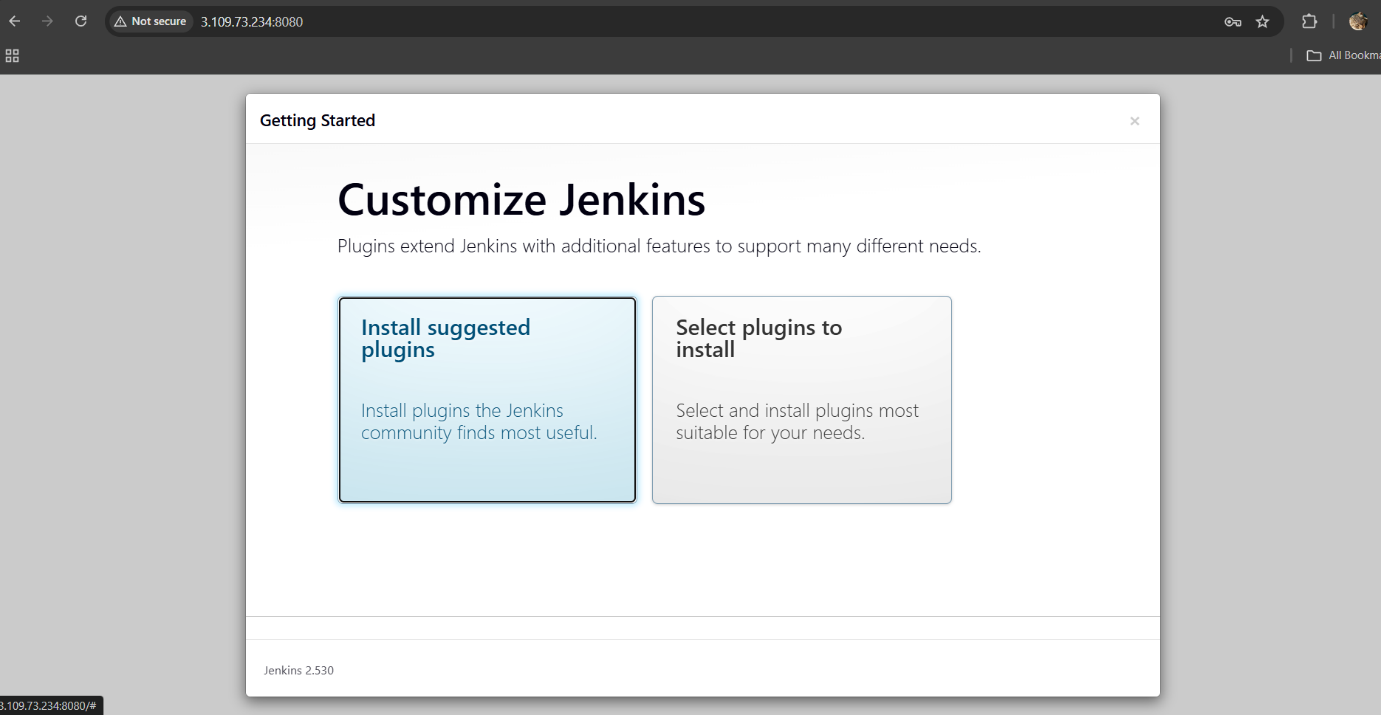
**Step 3: Access Jenkins Web UI**

1. Open browser: http://<EC2\_PUBLIC\_IP>:8080
2. Get initial password:

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

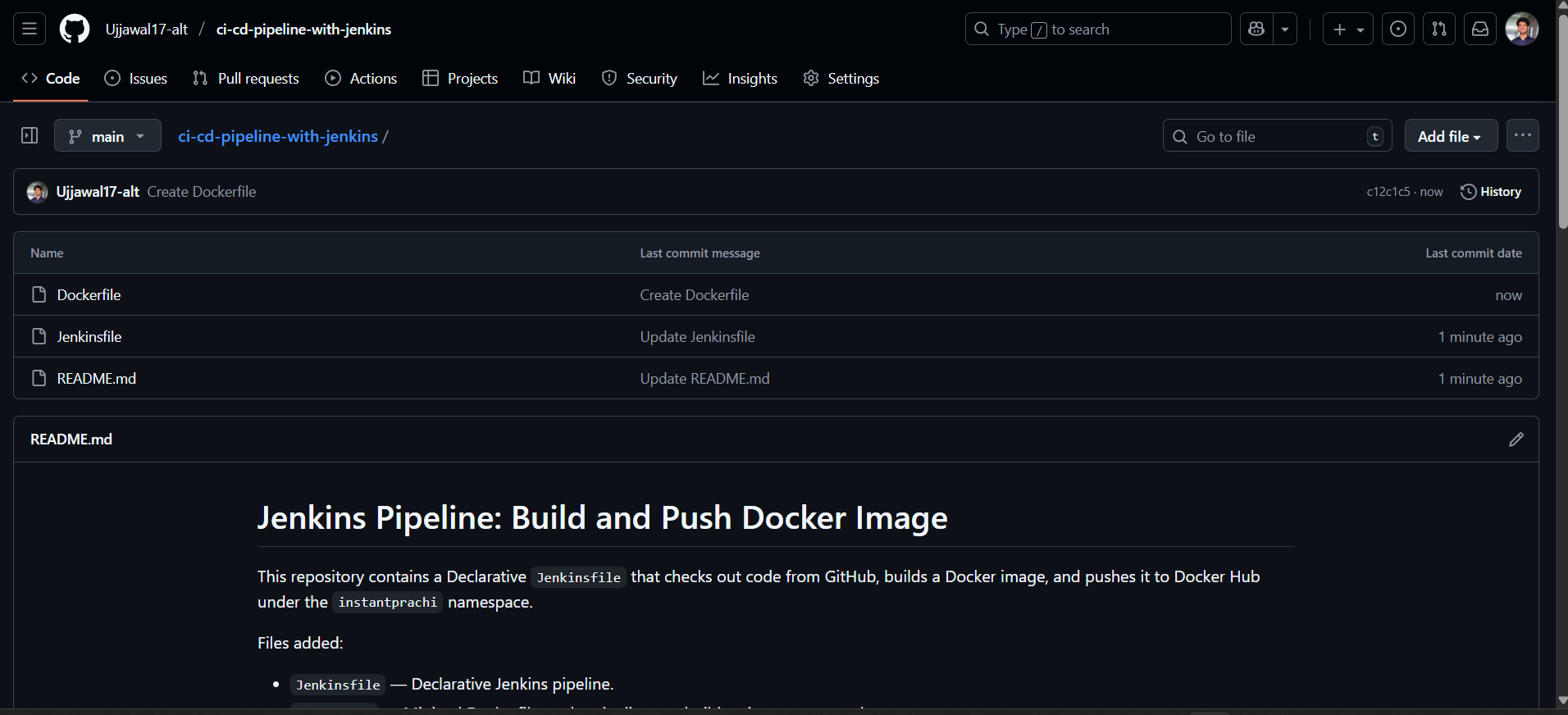
1. Install suggested plugins.
2. Create first admin user.

****

****

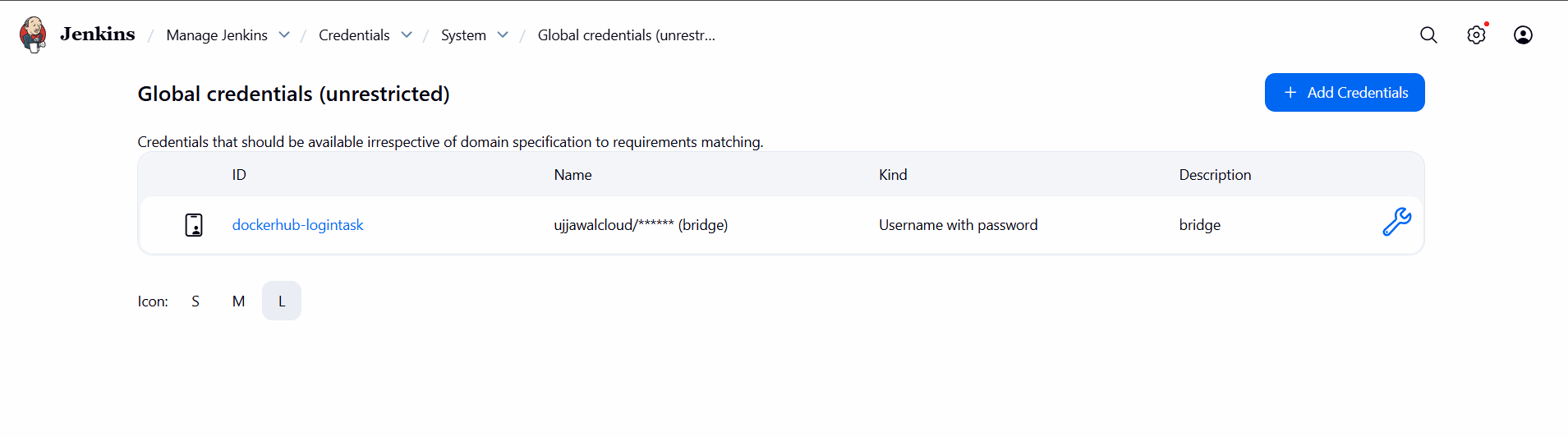
**Step 4: Setup GitHub Repository**

1. Create GitHub repo for your project.
2. Include a **Dockerfile** for your app.
3. Copy HTTPS/SSH URL.

****

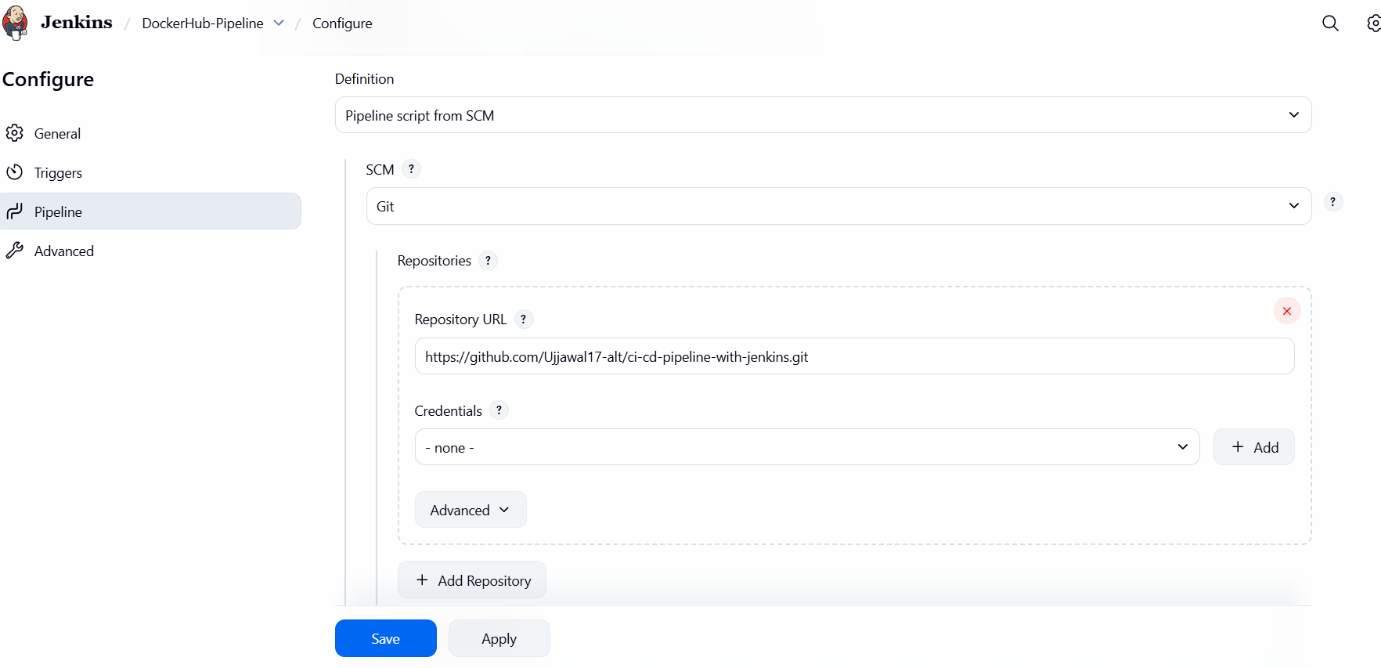
**Step 5: Add Docker Hub Credentials in Jenkins**

1. Jenkins → Manage Jenkins → Credentials → Global → Add Credentials
   * Kind: Username with password
   * Username: instantprachi
   * Password: Docker Hub password or token
   * ID: dockerhub-id



**Step 6: Create Jenkins Pipeline Job**

1. Jenkins → New Item → Pipeline → Name: DockerHub-Pipeline
2. Pipeline definition → **Pipeline script from SCM**
3. SCM: Git  
   Repository URL: <your-GitHub-repo-URL>  
   Branch: main  
   Script Path: Jenkinsfile



**Step 7: Create Jenkinsfile (Declarative Pipeline)**

pipeline {

agent any

environment {

DOCKERHUB\_CREDENTIALS = 'dockerhub-id'

DOCKER\_IMAGE = 'instantprachi/myapp:latest'

GIT\_REPO = 'https://github.com/yourusername/your-repo.git'

}

stages {

stage('Clone Repository') {

steps {

git url: "${GIT\_REPO}", branch: 'main'

}

}

stage('Build Docker Image') {

steps {

script {

docker.build("${DOCKER\_IMAGE}")

}

}

}

stage('Push Docker Image') {

steps {

script {

docker.withRegistry('https://index.docker.io/v1/', "${DOCKERHUB\_CREDENTIALS}") {

docker.image("${DOCKER\_IMAGE}").push()

}

}

}

}

}

post {

success {

echo "Docker Image pushed successfully!"

}

failure {

echo "Pipeline failed!"

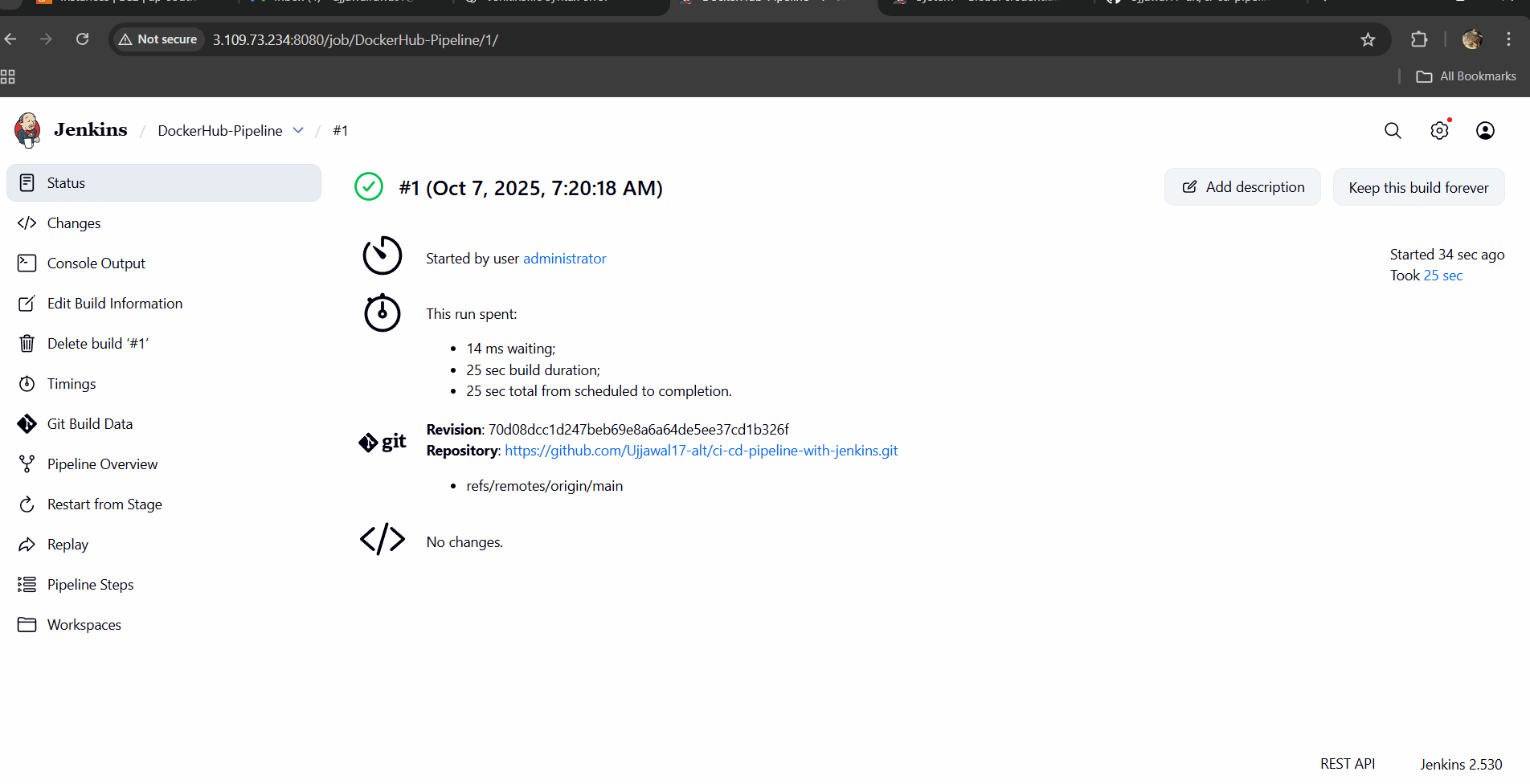
}

}

}

**Step 8: Run Pipeline**

1. Click **Build Now** in Jenkins.
2. Console output shows:
   * Cloning GitHub repo
   * Building Docker image
   * Pushing to Docker Hub



**Step 9: Verify Docker Hub Image**

1. Login to Docker Hub.
2. Check **instantprachi/myapp** repository.
3. Confirm **latest** image pushed.

