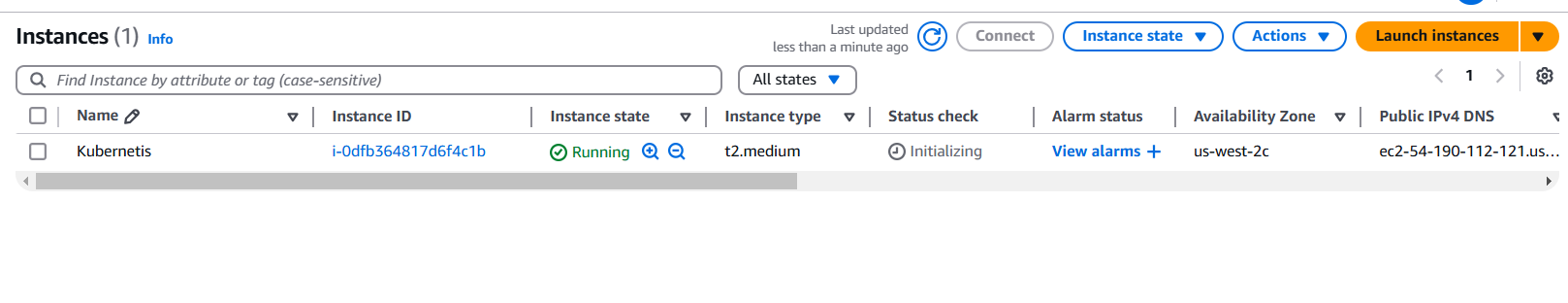
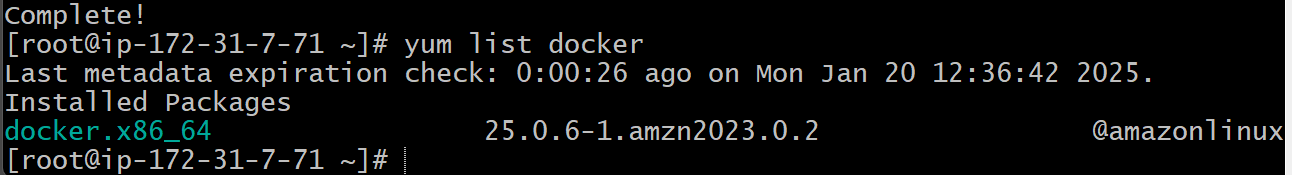
Assignment : Kubernetes

Step 1: Create server with t2.medium



Step 2: Install Docker in it and start the docker service

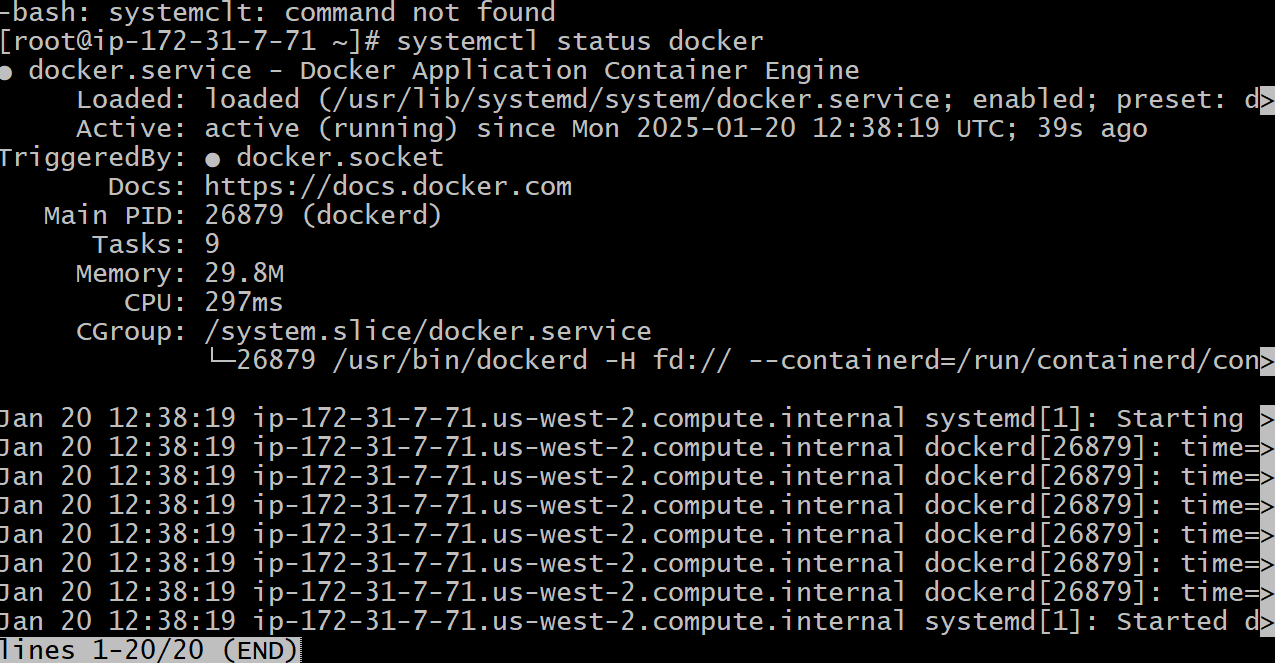
Command : yum install docker -y



Step 3: Start the docker service

Command: systemctl start docker

Systemctl enable docker



Step 4 : Create Docker for Jenkins which contains java-17, docker ,maven

Dockerfile:  
FROM jenkins/jenkins:lts

USER root

# Install Docker

RUN apt-get update && apt-get install -y docker.io \

&& rm -rf /var/lib/apt/lists/\*

# Install JDK 17 and Maven

RUN apt-get update && apt-get install -y openjdk-17-jdk maven \

&& rm -rf /var/lib/apt/lists/\*

# Set environment variables

ENV JAVA\_HOME=/usr/lib/jvm/java-17-openjdk-amd64

ENV PATH="$JAVA\_HOME/bin:$PATH"

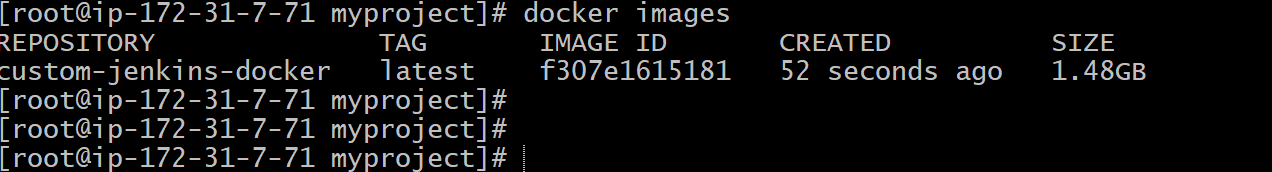
USER Jenkins

After that buid the docker file

Command: docker build -t custom-jenkins-docker .

Step 5: Check the custom image built or not

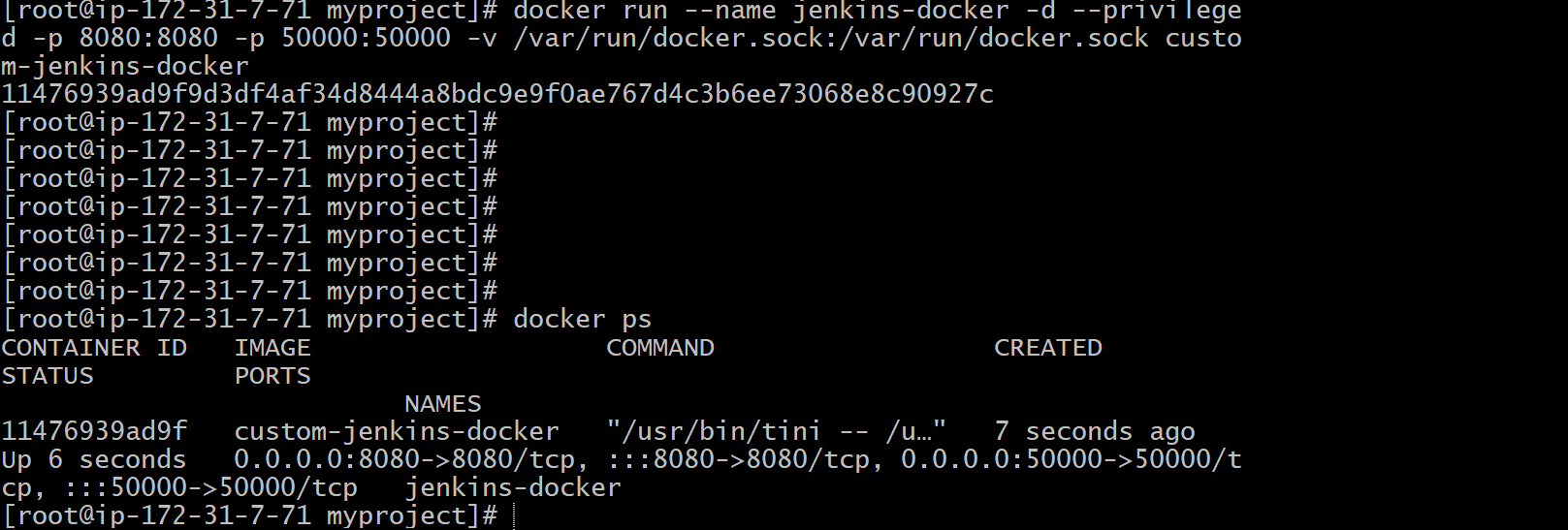
Command: docker images



Step 6:Run the images and create container for the same

Command: docker run --name jenkins-docker -d --privileged -p 8080:8080 -p 50000:50000 -v /var/run/docker.sock:/var/run/docker.sock custom-jenkins-docker  
  
and check process is running or not

Command: docker ps



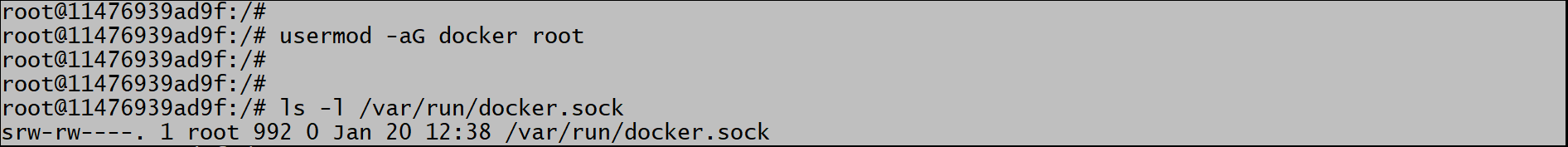
Step 6: Now enete in container

Command: docker exec -it --user root 11476939ad9f bash

Now create rrot user in container

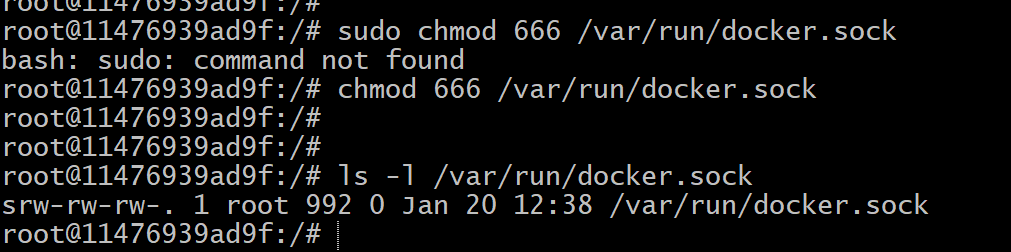
Command: usermod -aG docker root

Now provide the credentials for container



Now change the permission for docker

chmod 666 /var/run/docker.sock



Step 7: Now provide the access to Jenkins user using visudo

Command: visudo

Prerqusite: install sudo

Command: apt update && apt install nano -y

Install yum :

Command: apt update && apt install vim -y

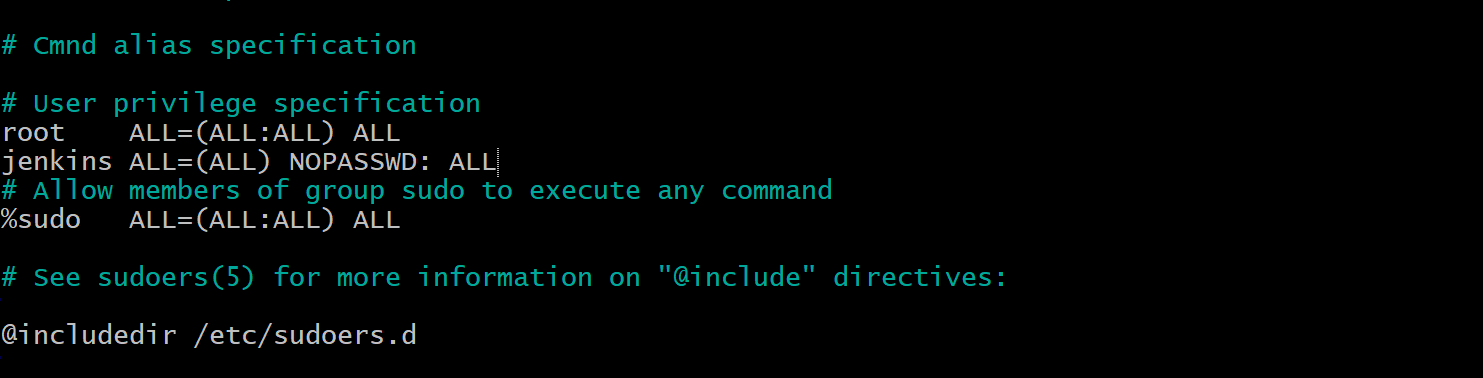
apt-get update && apt-get install -y sudo

Now open visudo

Command: jenkins ALL=(ALL) NOPASSWD: ALL

Save ctrl+s

Exit:ctrl+x



Step 7:Check docker install in container along with maven



Step 8:Now Add the jenknis script

pipeline {

agent any

environment {

DOCKER\_HUB\_USERNAME = 'manish527'

DOCKER\_HUB\_PASSWORD = 'Manish@1058'

KUBECONFIG = '/var/lib/jenkins/.kube/config' // Optional if using Kubeconfig directly

}

stages {

stage('Git Checkout') {

steps {

git 'https://github.com/Manish172-hub/myweb.git'

}

}

stage('Build') {

steps {

sh "mvn clean package"

}

}

stage('Docker Image Build') {

steps {

script {

// Ensure Docker is installed and running on the agent

sh "docker build -t b840image ."

}

}

}

stage('Docker Image Push to Docker Hub') {

steps {

script {

// Log in to Docker Hub securely

withCredentials([usernamePassword(credentialsId: 'docker-hub-creds', passwordVariable: 'DOCKER\_HUB\_PASSWORD', usernameVariable: 'DOCKER\_HUB\_USERNAME')]) {

sh 'echo $DOCKER\_HUB\_PASSWORD | docker login -u $DOCKER\_HUB\_USERNAME --password-stdin'

}

// Tag the image for Docker Hub

sh 'docker tag b840image manish527/b840image'

// Push the Docker image to Docker Hub

sh 'docker push manish527/b840image'

}

}

}

stage('Kubernetes Deployment') {

steps {

script {

// Ensure kubectl has access to your Kubernetes cluster

// Using Kubeconfig for authentication in the pipeline

withCredentials([file(credentialsId: 'kubeconfig-id', variable: 'KUBECONFIG')]) {

// Apply the Kubernetes deployment

sh "kubectl apply -f deployments.yml"

// Expose the deployment as a NodePort

sh "kubectl expose deployment mywebdeployment --port=8080 --type=NodePort"

}

}

}

}

}

post {

failure {

echo "Pipeline failed, please check the logs."

}

success {

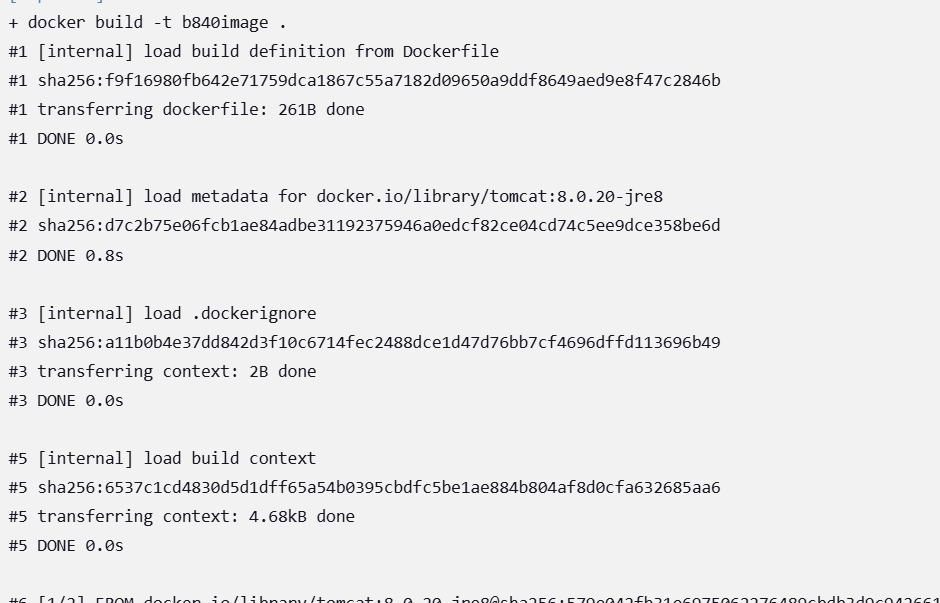
echo "Pipeline completed successfully!"

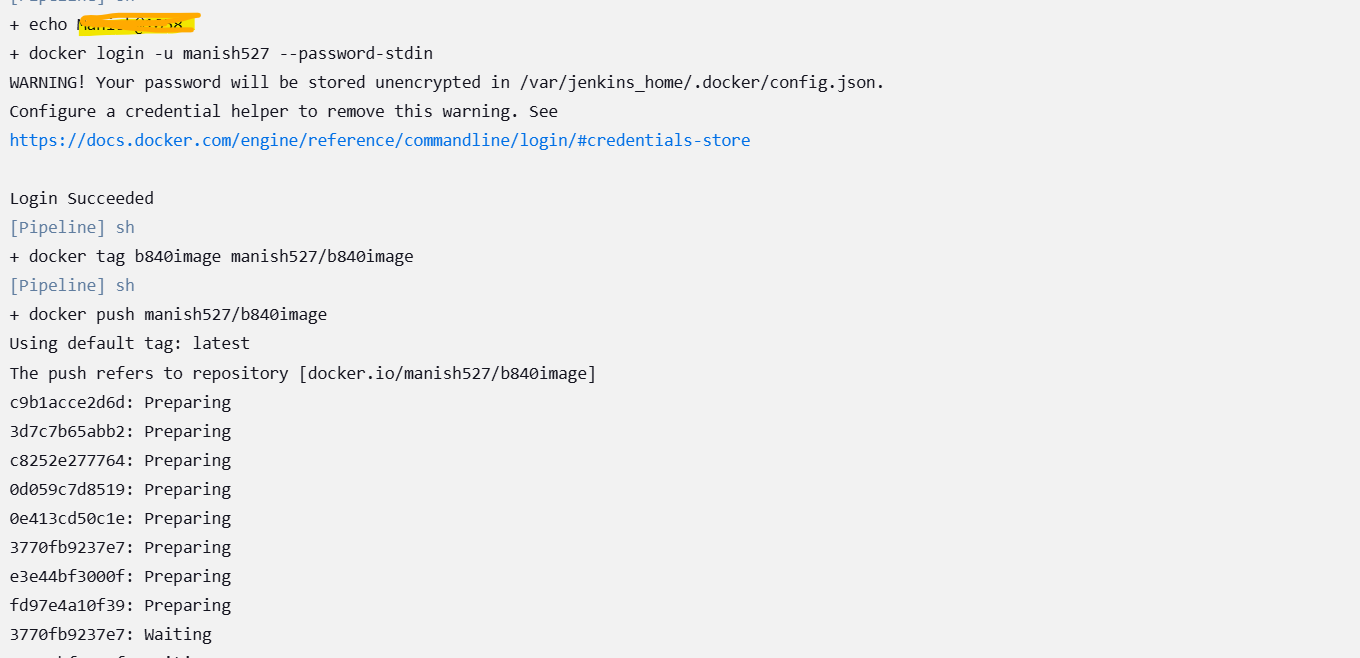
}

}

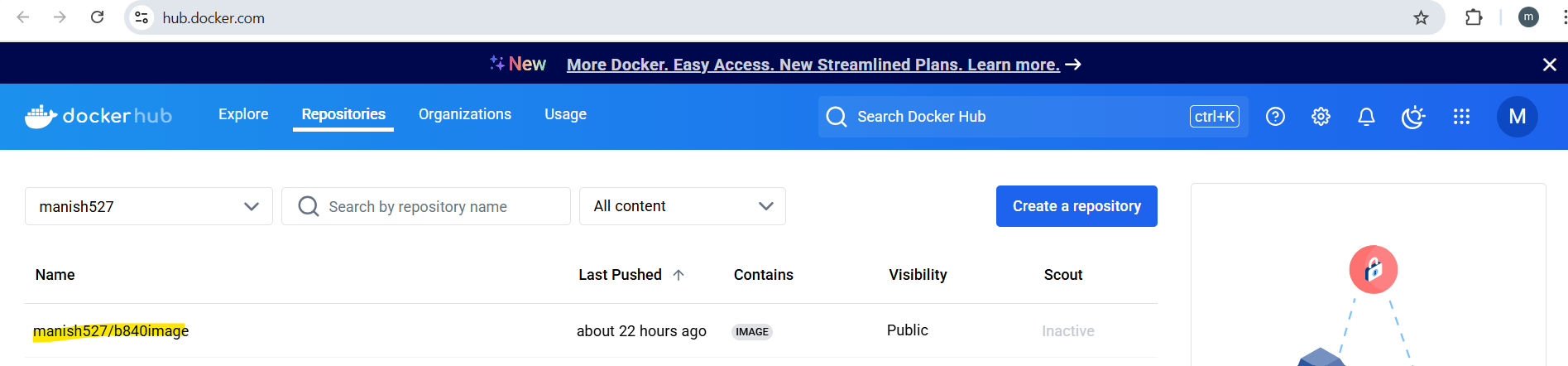
}

Step 9:Check the console output





Step 9: Verify the image in docker hub



Step 10:Now install kubectl in container or create docker file

In progress

3.Install AWS CLI

# Update the package index

sudo apt-get update

# Install required dependencies

sudo apt-get install unzip curl -y

# Download the AWS CLI v2 package

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

# Unzip the downloaded package

unzip awscliv2.zip

# Run the install script

sudo ./aws/install

# Verify the installation

aws --version

4. Provide aws configure

5aws eks --region us-west-2 update-kubeconfig --name cluster-name

6.sudo apt-get update

sudo apt-get install less