DAY 2

# Creating a docker image using

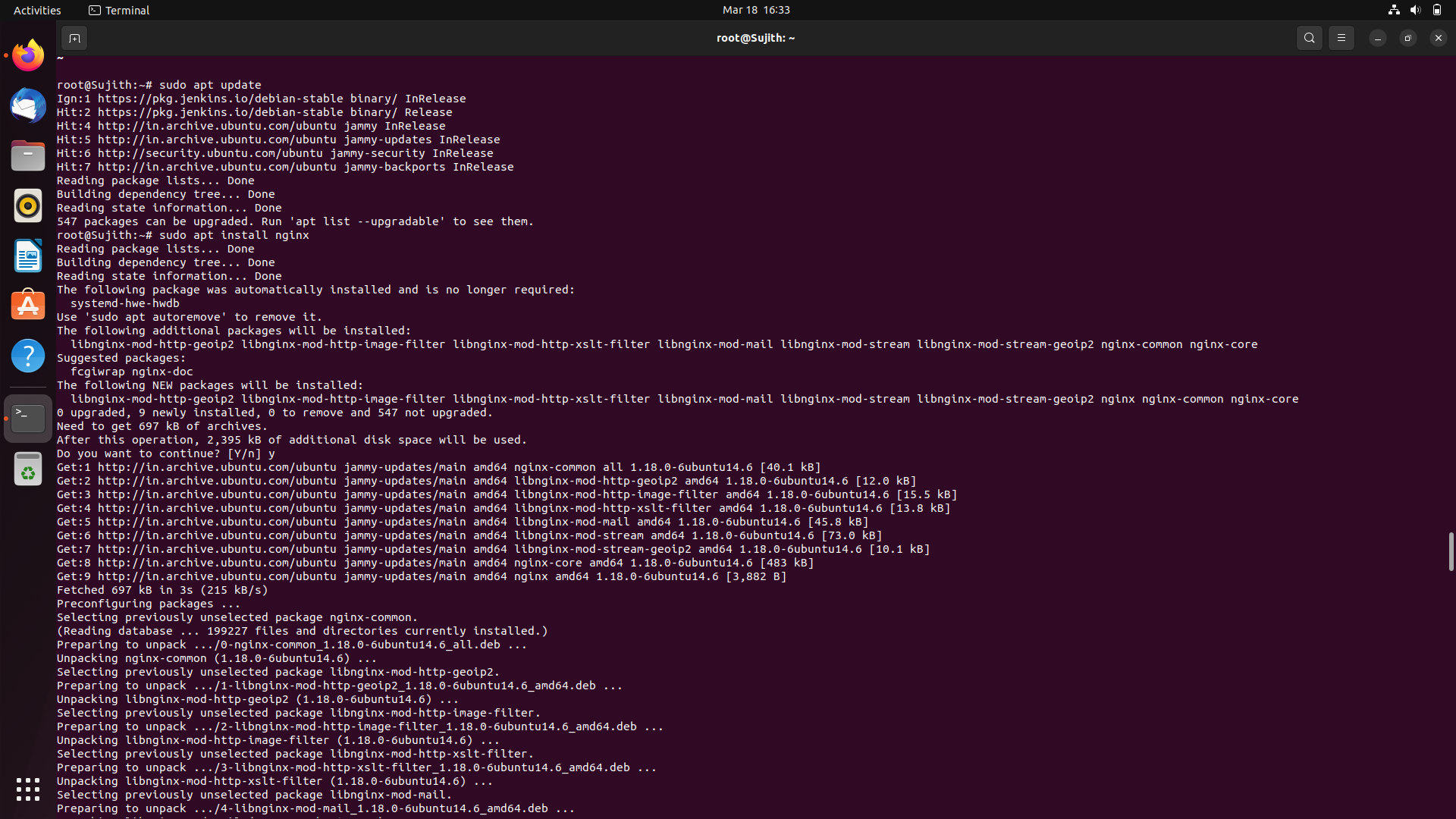
## Installing Docker

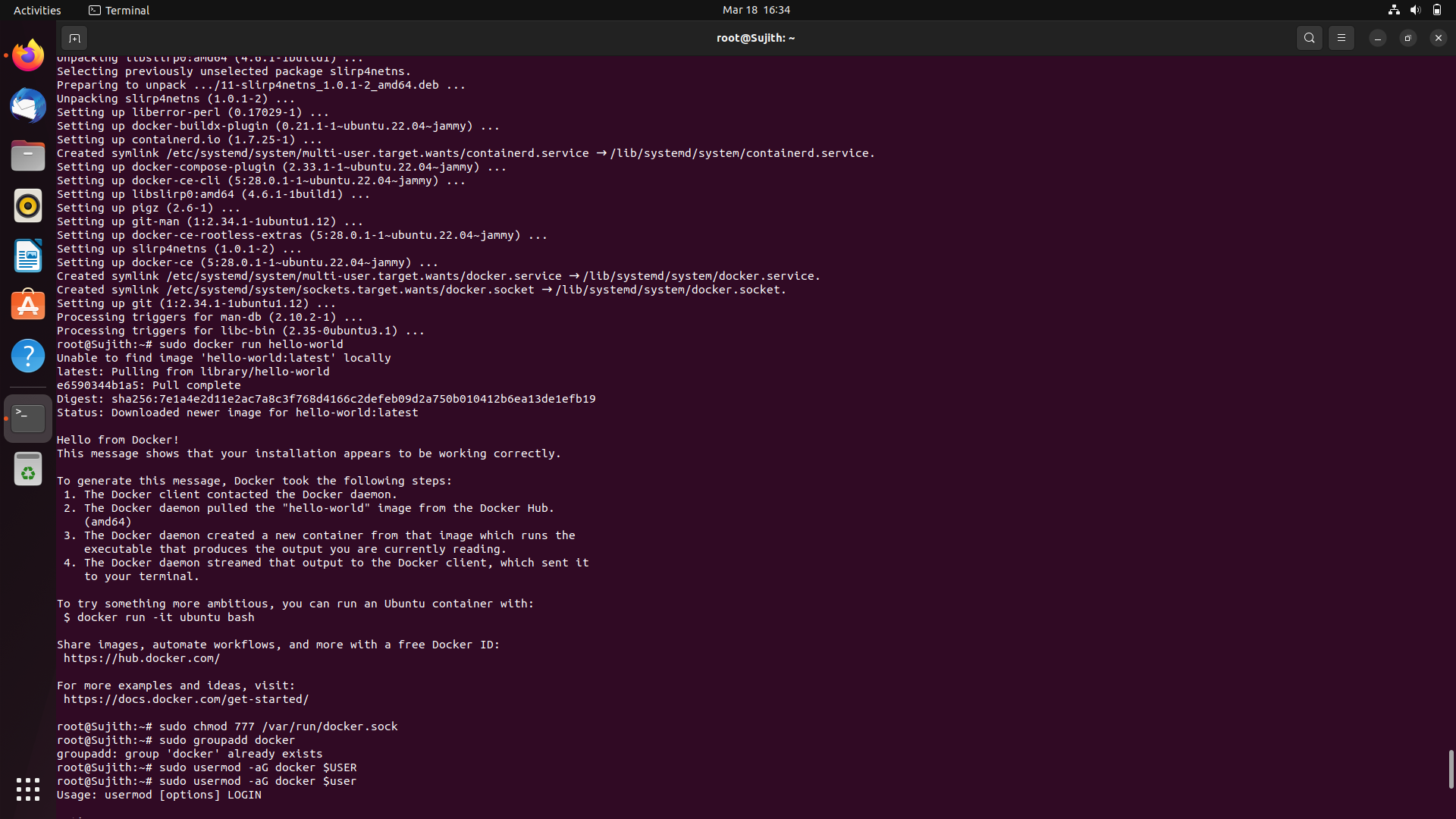
* Enter the following commands to inatall and verify installation of Docker

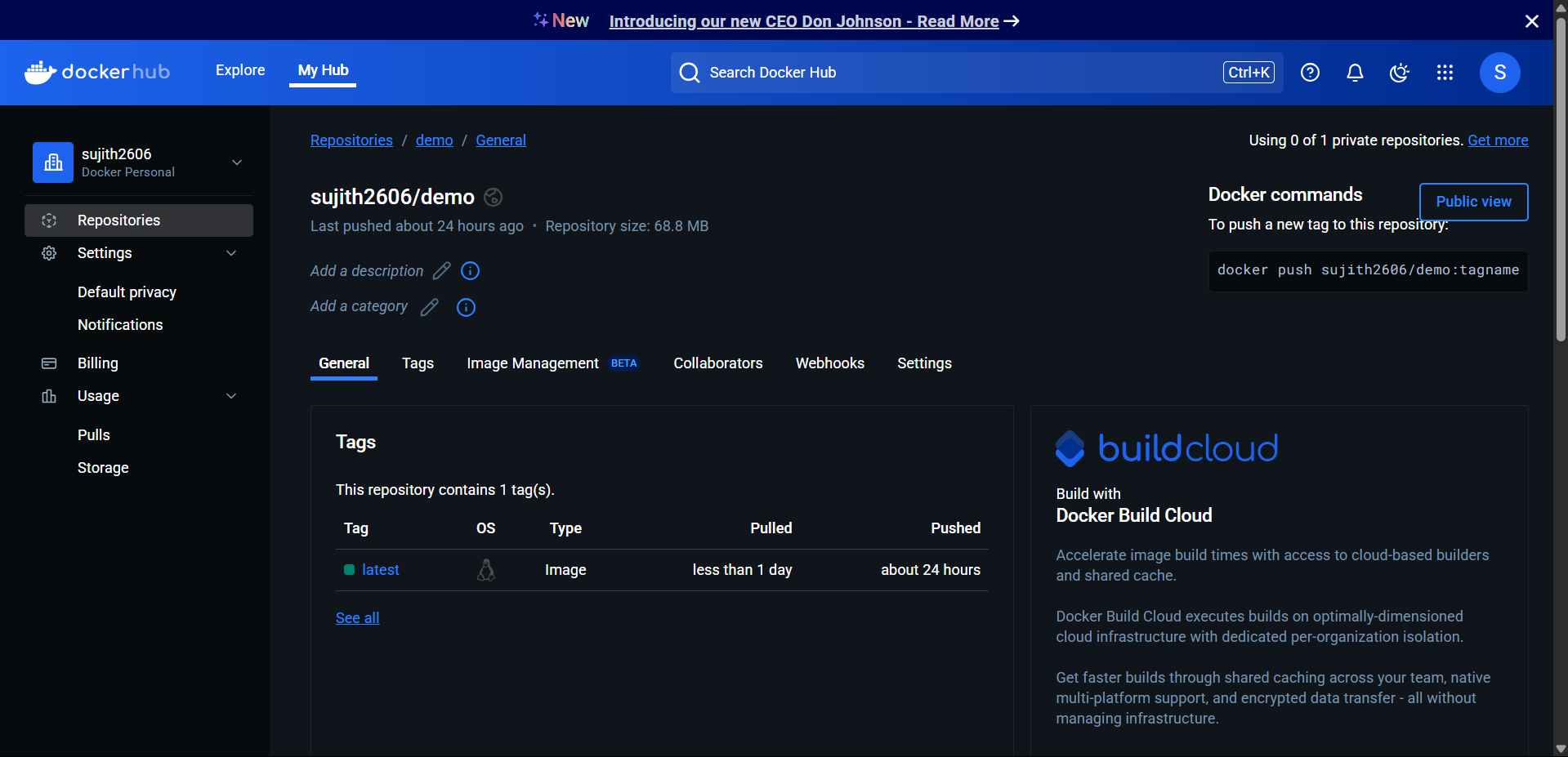
sudo apt update

sudo apt install -y docker.io

sudo systemctl enable docker –now







2.PIPELINE SCRIPT:

pipeline {

agent any

tools {

jdk 'jdk17'

nodejs 'node'

}

stages {

stage('Clean Workspace') {

steps {

script {

echo "Cleaning workspace..."

deleteDir() // Deletes everything in the Jenkins workspace before starting

}

}

}

stage('Git Checkout') {

steps {

script {

// Checkout the GitHub repository

git branch: 'main',

credentialsId: 'github',

url: 'https://github.com/Nithish-ponnusamy/devops.git' // Update with your repository URL

}

}

}

stage('Install Node') {

steps {

script {

// Check if package.json exists, then run npm install

sh '''

if [ -f package.json ]; then

echo "package.json found. Running npm install..."

npm install

else

echo "ERROR: package.json is missing. Skipping npm install."

exit 1

fi

'''

}

}

}

stage('Docker Build & Push') {

steps {

script {

// Build and push Docker image

withDockerRegistry(credentialsId: 'docker', toolName: 'docker') {

def imageName = "nithi1230/docker\_hub" // Your Docker image name

def tag = "latest"

sh "docker build -t ${imageName} ."

sh "docker tag ${imageName} ${imageName}:${tag}"

sh "docker push ${imageName}:${tag}"

}

}

}

}

stage('Deploy Docker Container') {

steps {

script {

// Stop and remove the existing container if it's running

sh "docker stop devops\_container || true"

sh "docker rm devops\_container || true"

// Run the new container with proper port mapping

sh "docker run -d --name devops\_container -p 3001:3000 nithi1230/docker\_hub:latest"

}

}

}

}

}

3. TOOLS CONFIG:

Java

Nodejs

Docker

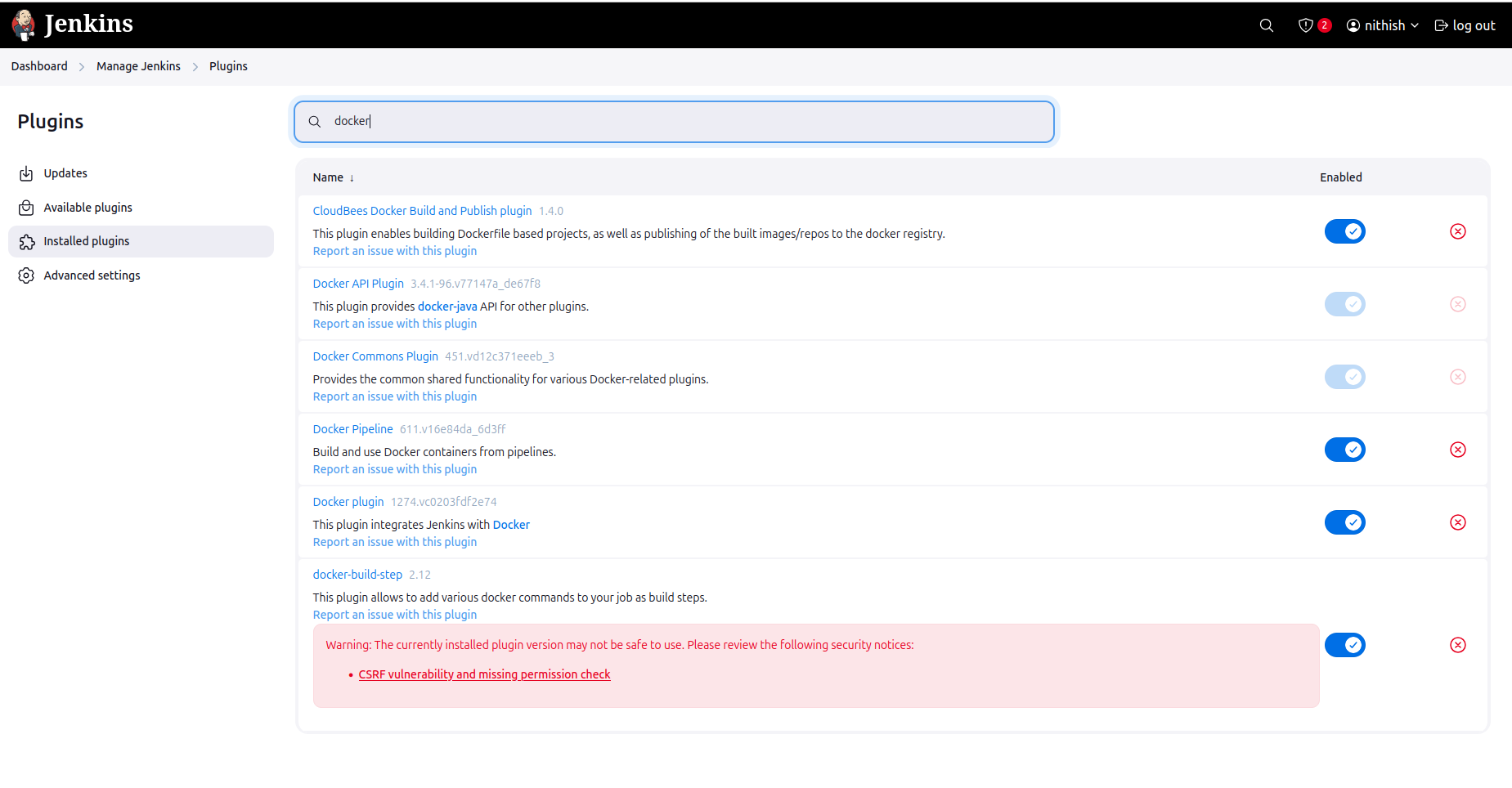
4.Plugins:

Nodejs

Docker related(except build)

Eclipse

Pipeline



5.credentials:

Github&docker

For docker build &deploy

Sudo usemod -aG docker Jenkins

Sudo chmod 666 /var/run/docker.sock

Sudo systemctl restart Jenkins

Sudo systemctl status Jenkins

## Setting up docker credentials

* Go to Jenkins > Manage Jenkins > Credentials > System > Global Credentials (Unrestricted) > Add Credentials
* Fill your Docker hub username , password, and in the id field enter docker-seccred

