**# Creating a docker image using**

**## Installing Docker**

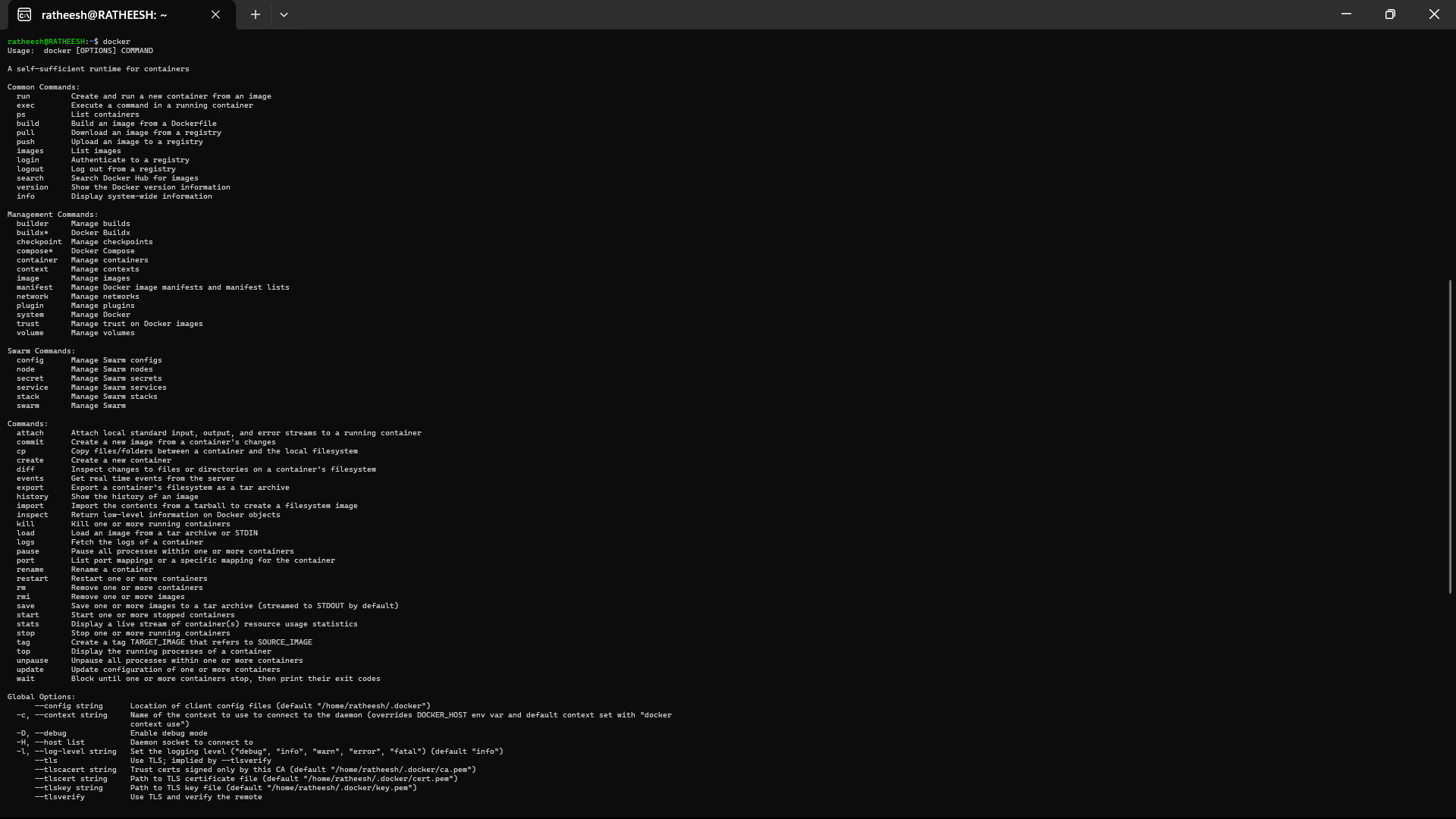
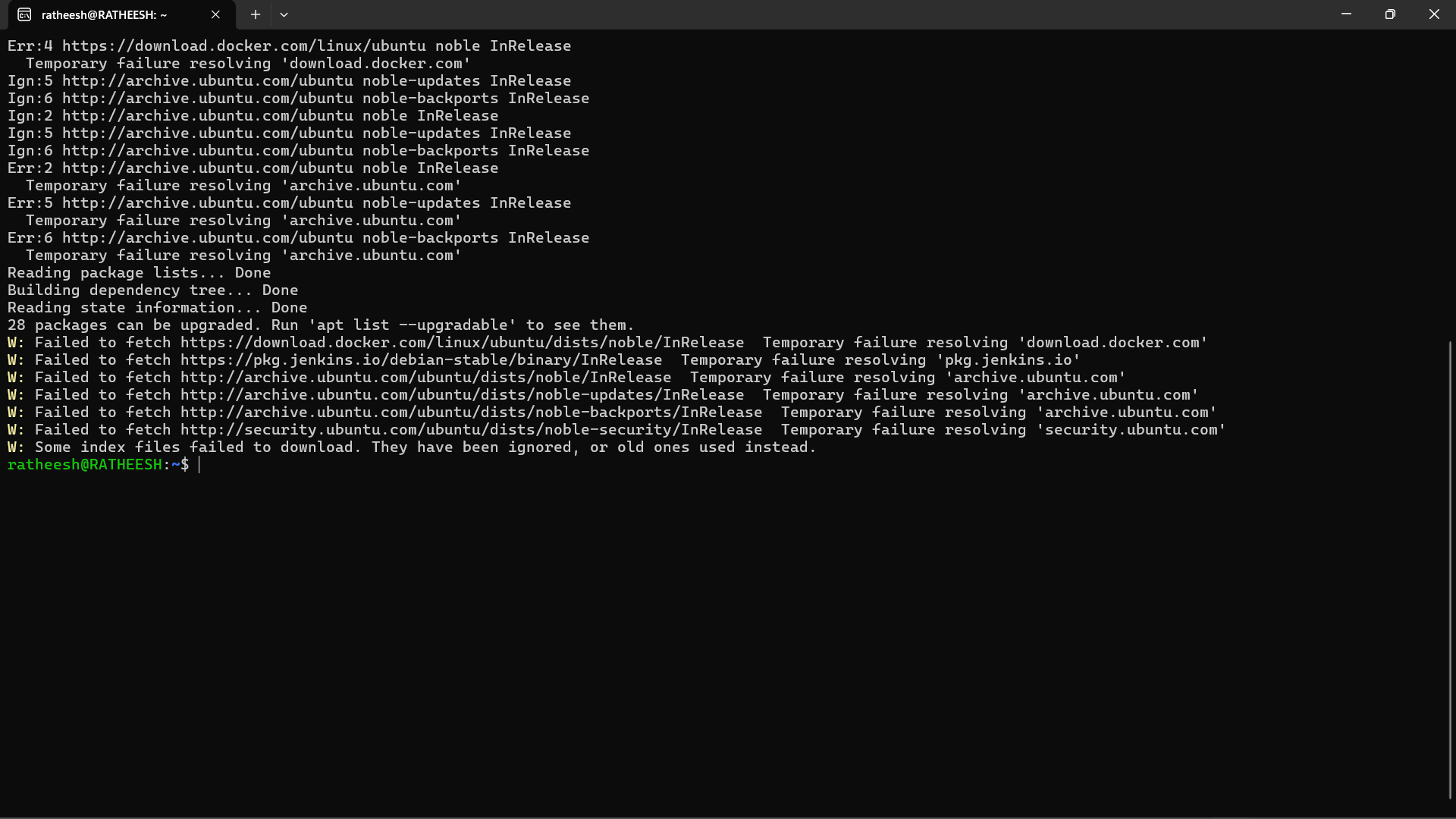
  - -Enter the following commands to inatall and verify installation of Docker

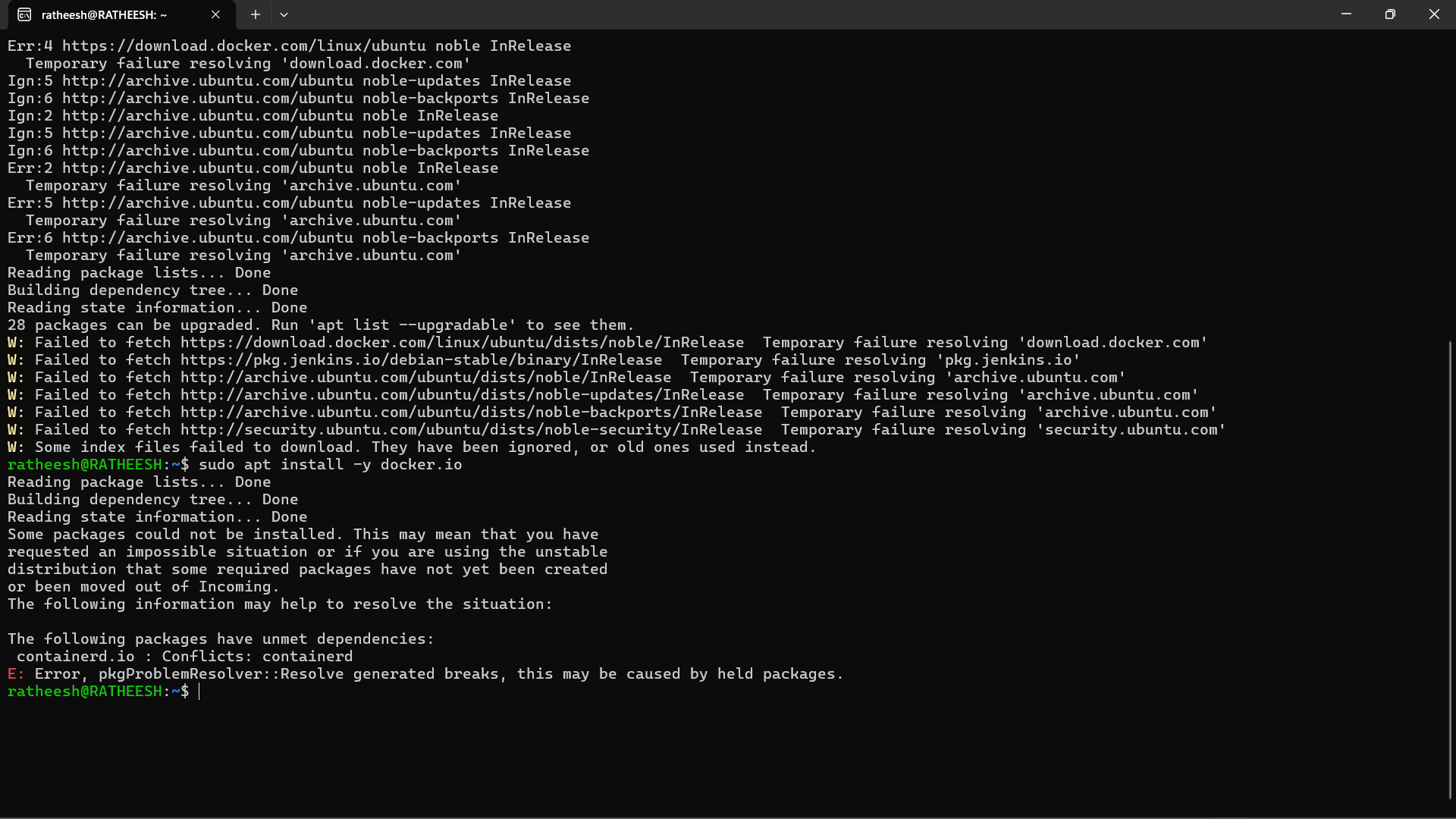
```bash

sudo apt update

sudo apt install -y docker.io

sudo systemctl enable docker --now

docker



**## Download Docker plugins in Jenkins**

 - Go to Jenkins `Dashboard` -> `Manage Jenkins` -> `Available Plugins` -> Search `Docker`

 - Select these plugins and Install

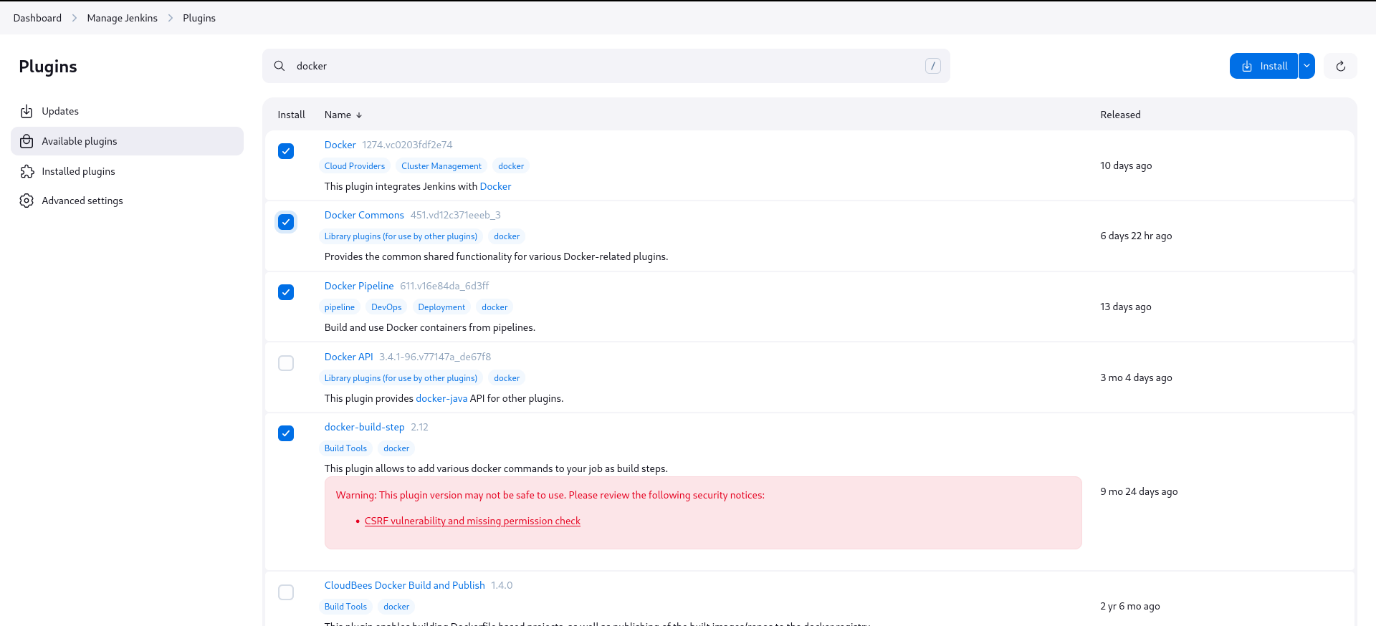
    - Docker

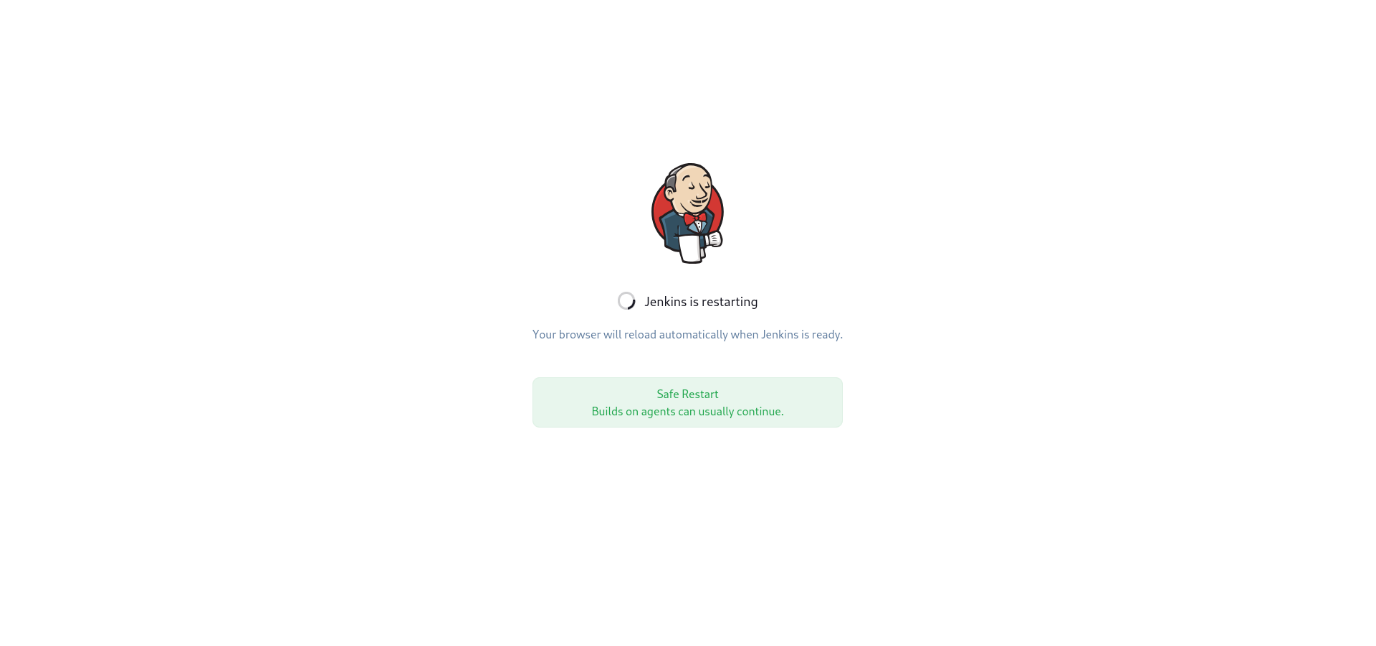
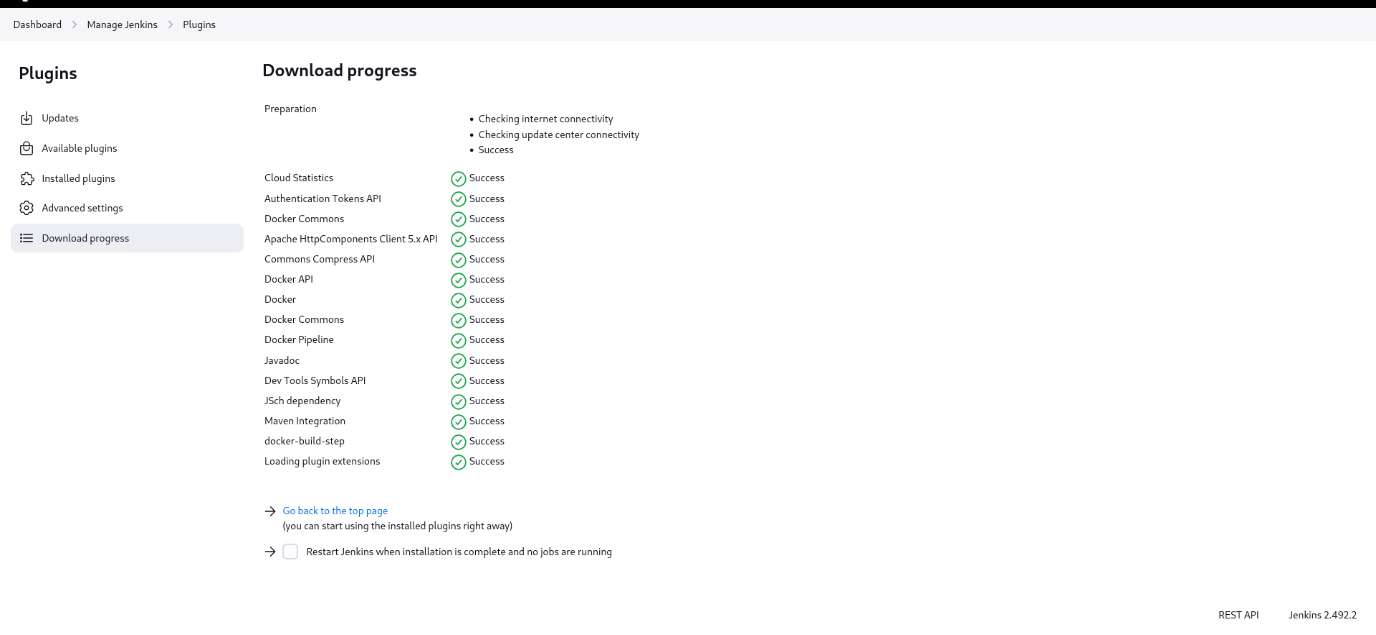
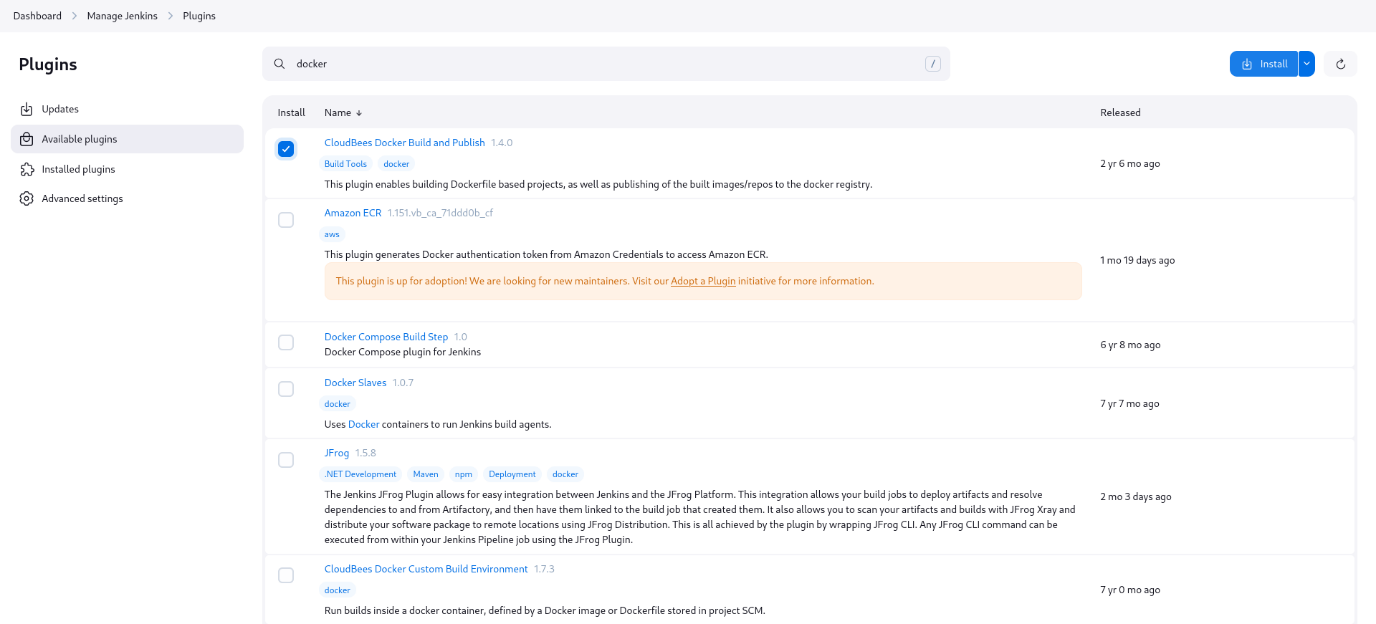
    - Docker Commons

    - Docker Pipeline

    - docker-build-step

    - CoudBees Docker Build and Publish

 - In this page Check the `Restart Jenkins` after installation this will restart Jenkins



**## Add Jenkins to Docker group**

 - Go to terminal and run these commands to add Jenkins to docker group

```bash

sudo usermod -aG docker jenkins

sudo systemctl restart jenkins

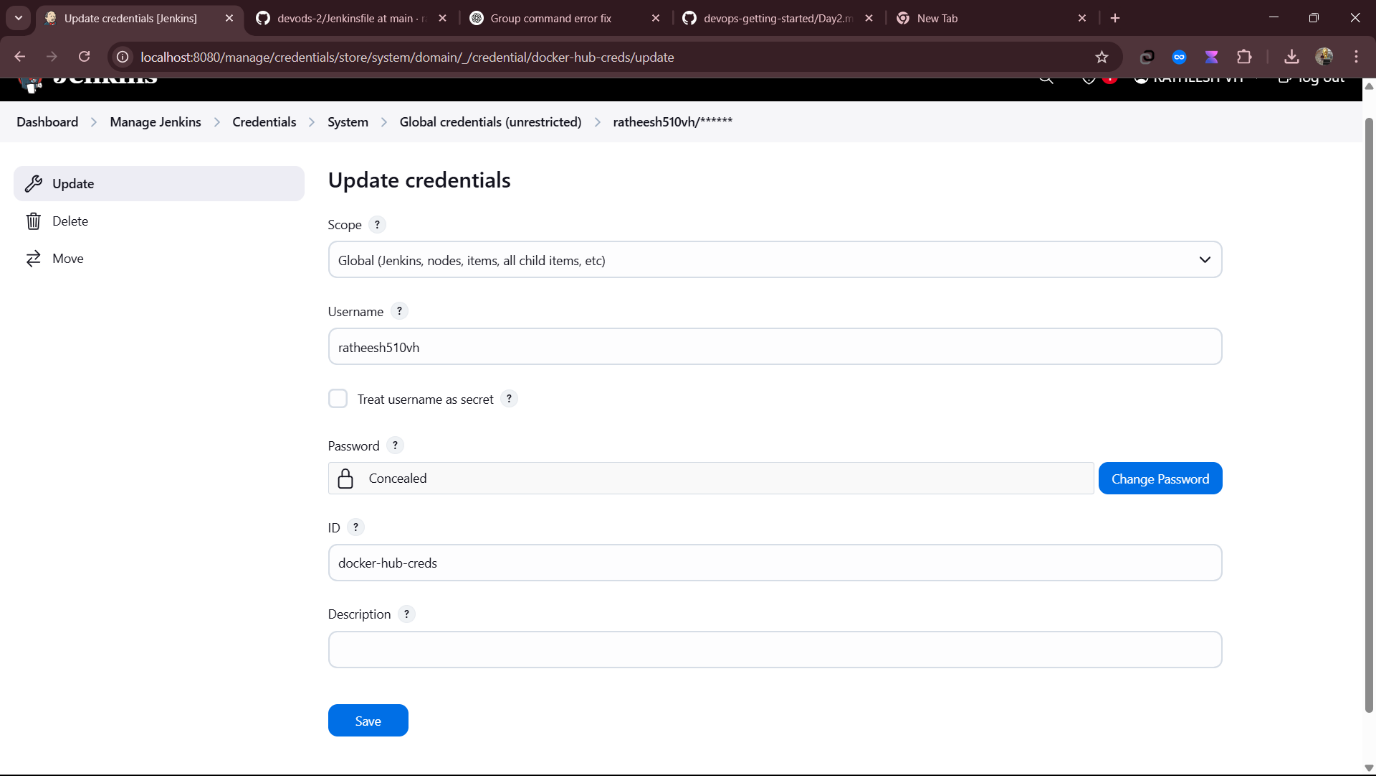
sudo reboot

```

- This will do its thing and reboot the system

**## 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`

**## Creating and building a pipeline**

 - Go to Jenkins `Dashboard` > `Create a Job`

 - Enter a project name

 - Select `pipeline`

 - Click `Ok`

 - Go to `pipeline`

 - Paste this script below and change the credential wherever mentioned:

```groovy

pipeline {

agent any

environment {

IMAGE\_NAME = "sivagurunathan7/test\_repository"

TAG = "latest"

CONTAINER\_NAME = "my-container"

PORT = "3001"

}

stages {

stage('Clone Repository') {

steps {

echo "Cloning GitHub repository..."

git 'https://github.com/Sivagurunathan98/New\_docker\_push.git'

}

}

stage('Build Docker Image') {

steps {

echo "Building Docker image..."

bat 'powershell.exe -Command "Set-ExecutionPolicy RemoteSigned -Force"'

bat 'powershell.exe -File .\\Build.ps1'

}

}

stage('Login to Docker Hub') {

steps {

echo "Logging into Docker Hub..."

withCredentials([usernamePassword(credentialsId: 'docker-hub-creds', usernameVariable: 'DOCKER\_USER', passwordVariable: 'DOCKER\_PASS')]) {

bat 'echo %DOCKER\_PASS% | docker login -u %DOCKER\_USER% --password-stdin'

}

}

}

stage('Push Docker Image') {

steps {

echo "Pushing Docker image to Docker Hub..."

bat "docker tag %IMAGE\_NAME%:%TAG% %IMAGE\_NAME%:%TAG%"

bat "docker push %IMAGE\_NAME%:%TAG%"

}

}

stage('Deploy Docker Container') {

steps {

echo "Deploying Docker container..."

bat 'powershell.exe -File .\\Deploy.ps1'

}

}

}

post {

success {

echo "Deployment Successful!"

}

failure {

echo "Deployment Failed!"

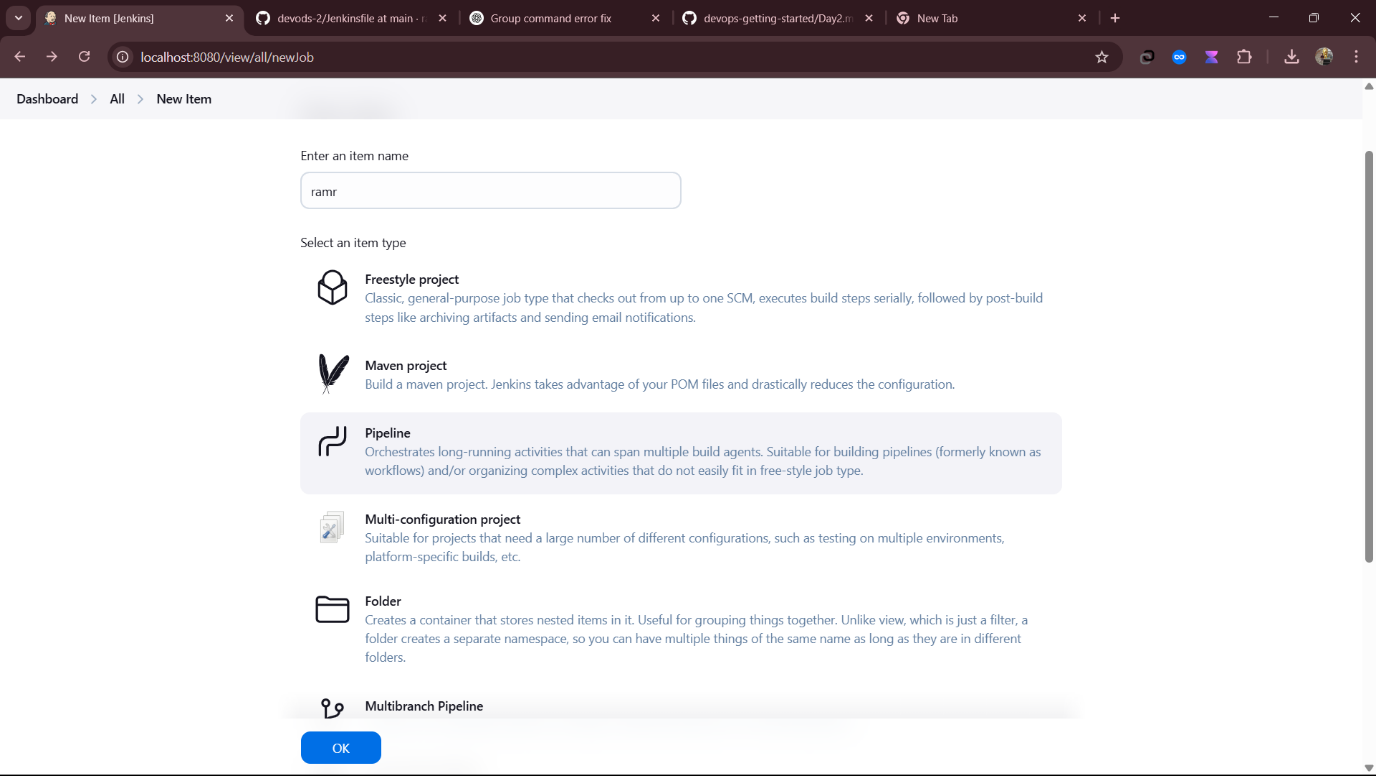
}

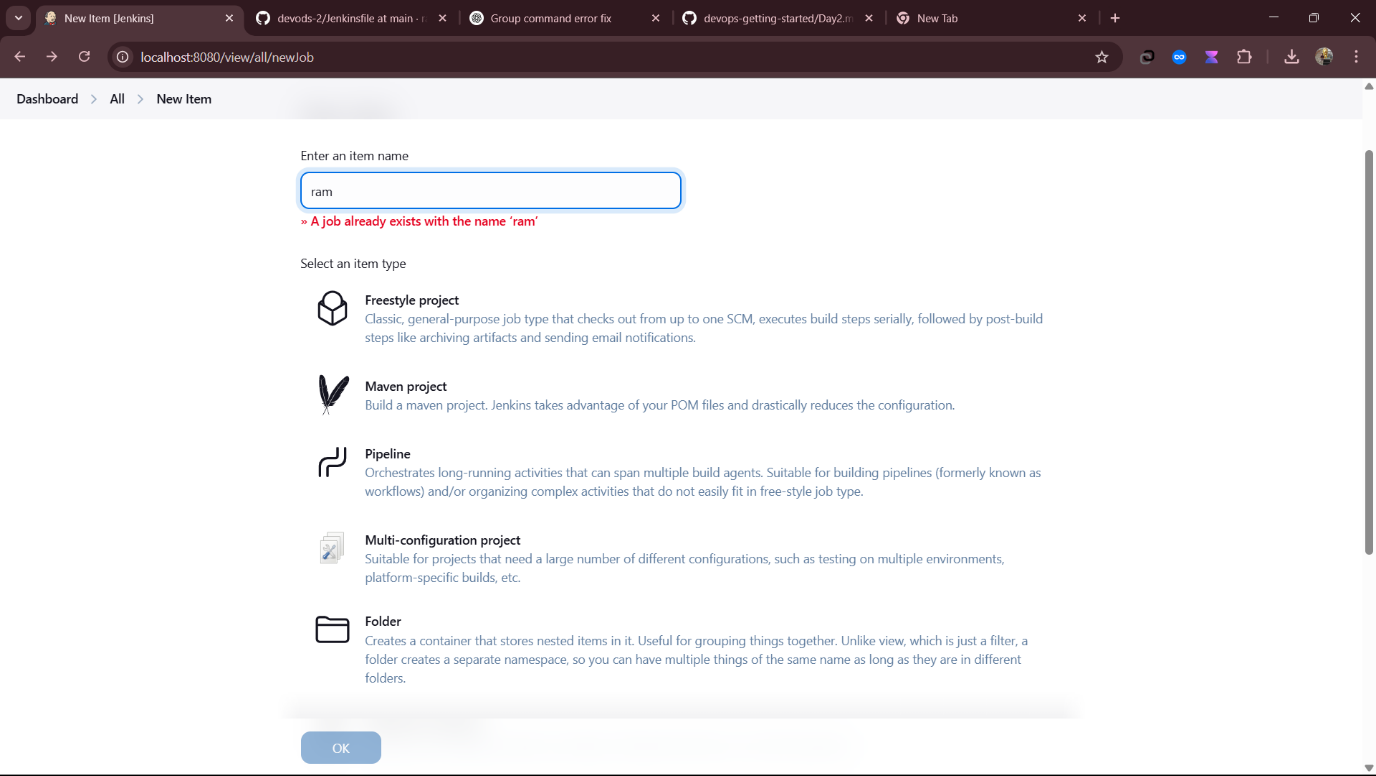
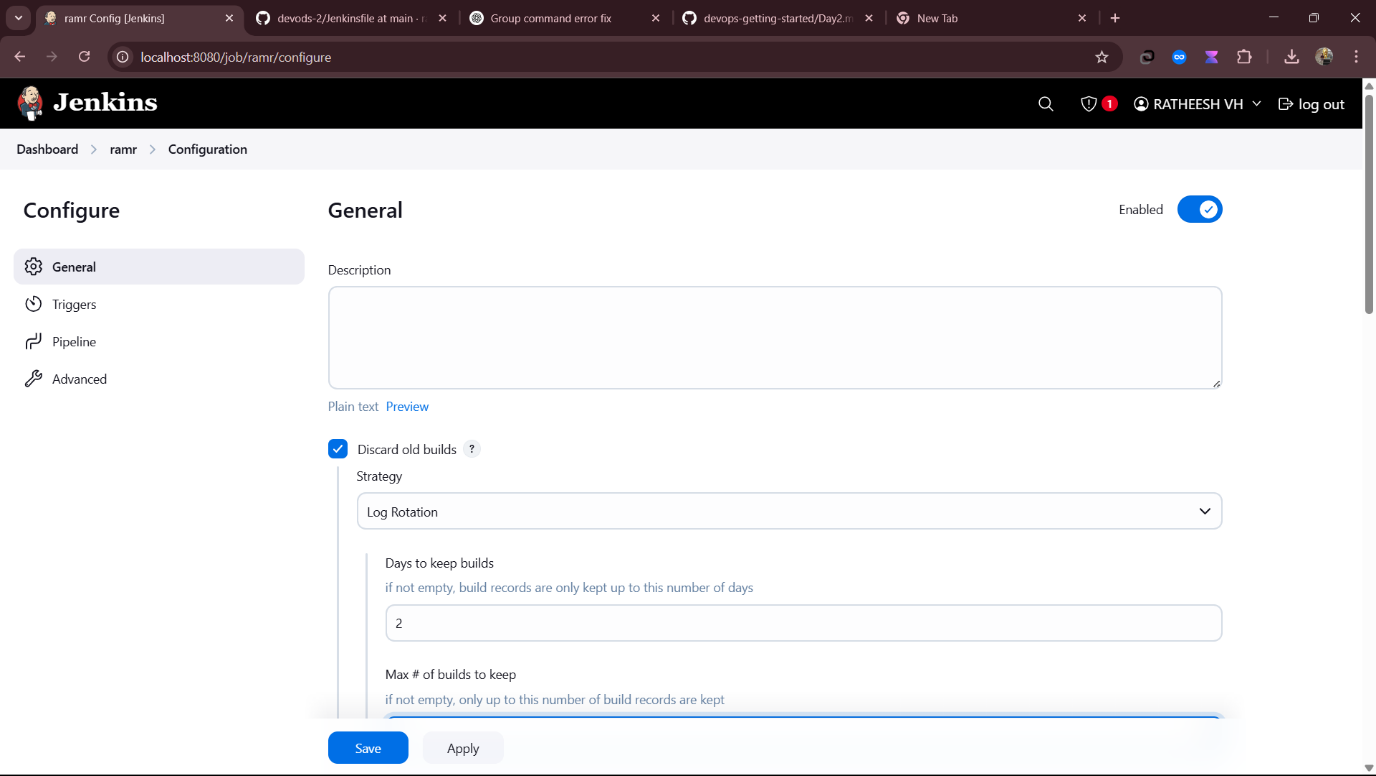
}

}

```

 - click `save`

 - click `build`



 - Go to Docker Hub to see your image pushed there