Jenkins CICD Pipeline

Installation procedure

Prerequisites

Jenkins Server: Ensure Jenkins is installed and running. You can set up Jenkins on a local server or use a cloud-based solution.

Docker: Install Docker on your Jenkins server or the machine where Jenkins agents will run. Ensure Docker is configured and accessible to Jenkins.

ASP.NET Core 6 Web API: Have a .NET Core 6 Web API project ready with a Dockerfile for containerization.

Install java 11,17 jdk version

Install jenkins LTS Version

Install jenkins war file

After install run the following commands:

C:\Users\Veritra>java --version

openjdk 17.0.11 2024-04-16

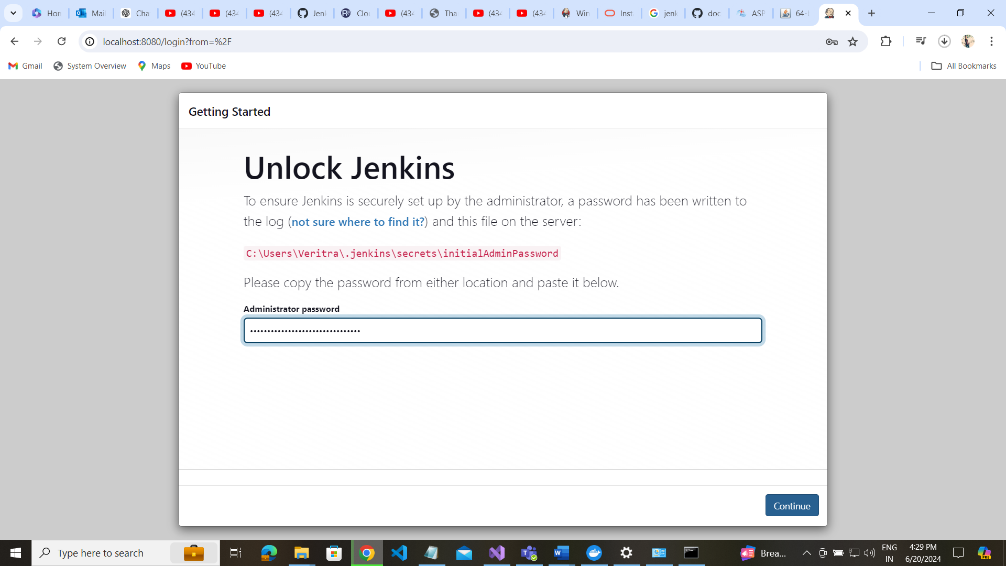
OpenJDK Runtime Environment OpenLogic-OpenJDK (build 17.0.11+9-adhoc..jdk17u)

OpenJDK 64-Bit Server VM OpenLogic-OpenJDK (build 17.0.11+9-adhoc..jdk17u, mixed mode)

C:\Users\Veritra>cd downloads

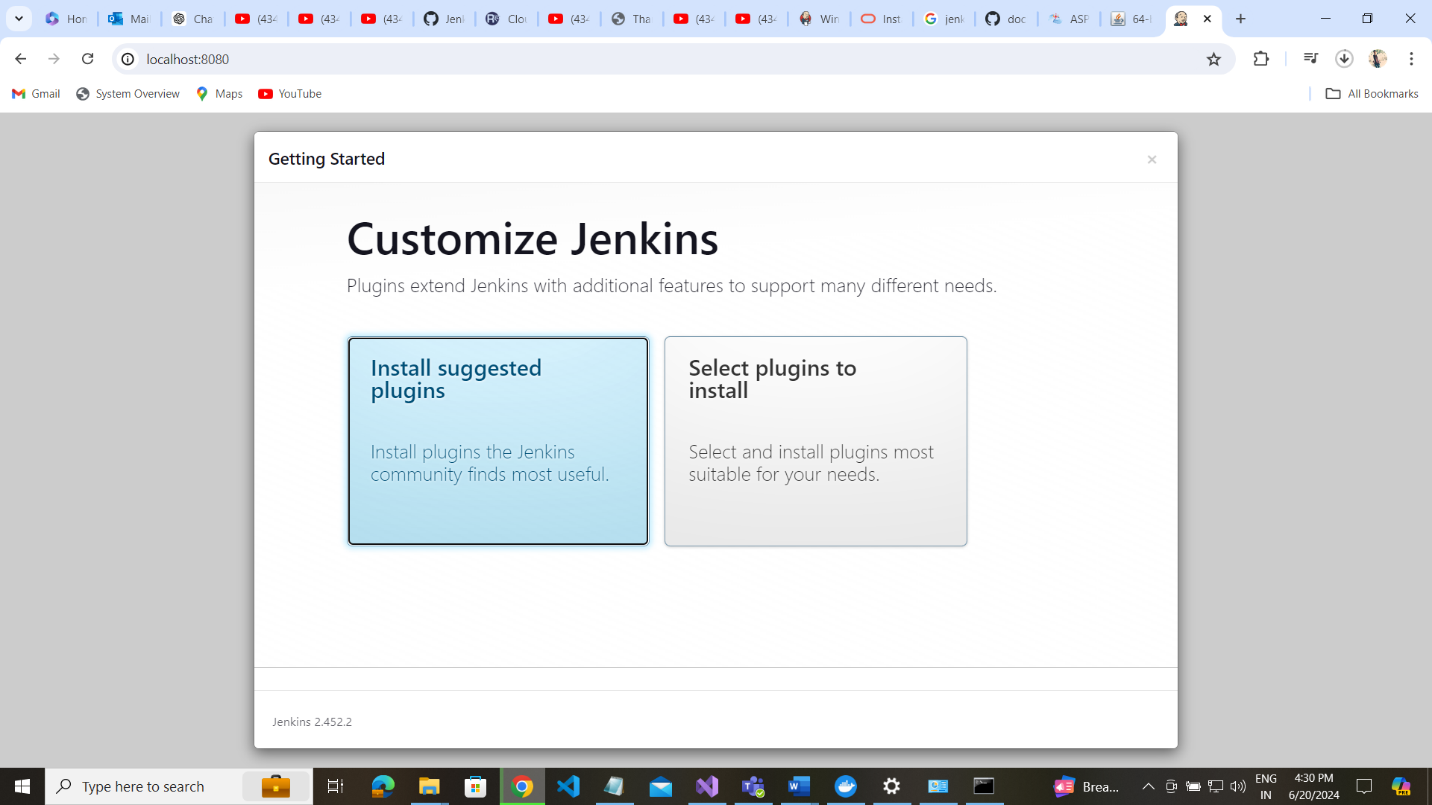
C:\Users\Veritra\Downloads>java -jar jenkins.war -httpPort=9191

copy:7241c9a536b5494bac5d49f9be7ec091 (generate password)



Paste the generated password here.

Select install suggested plugins.



Username: Saikumar

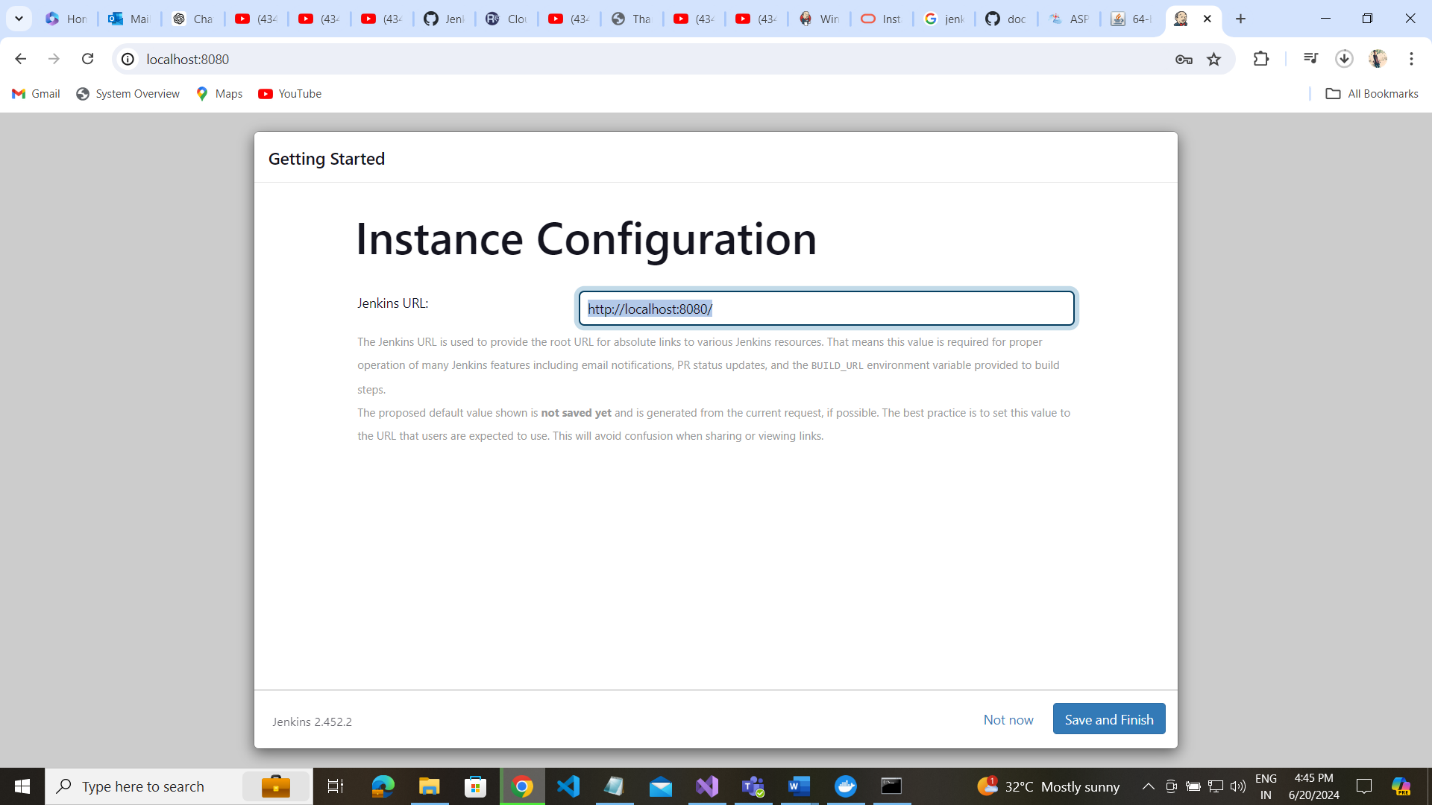
Password: S@1kumar

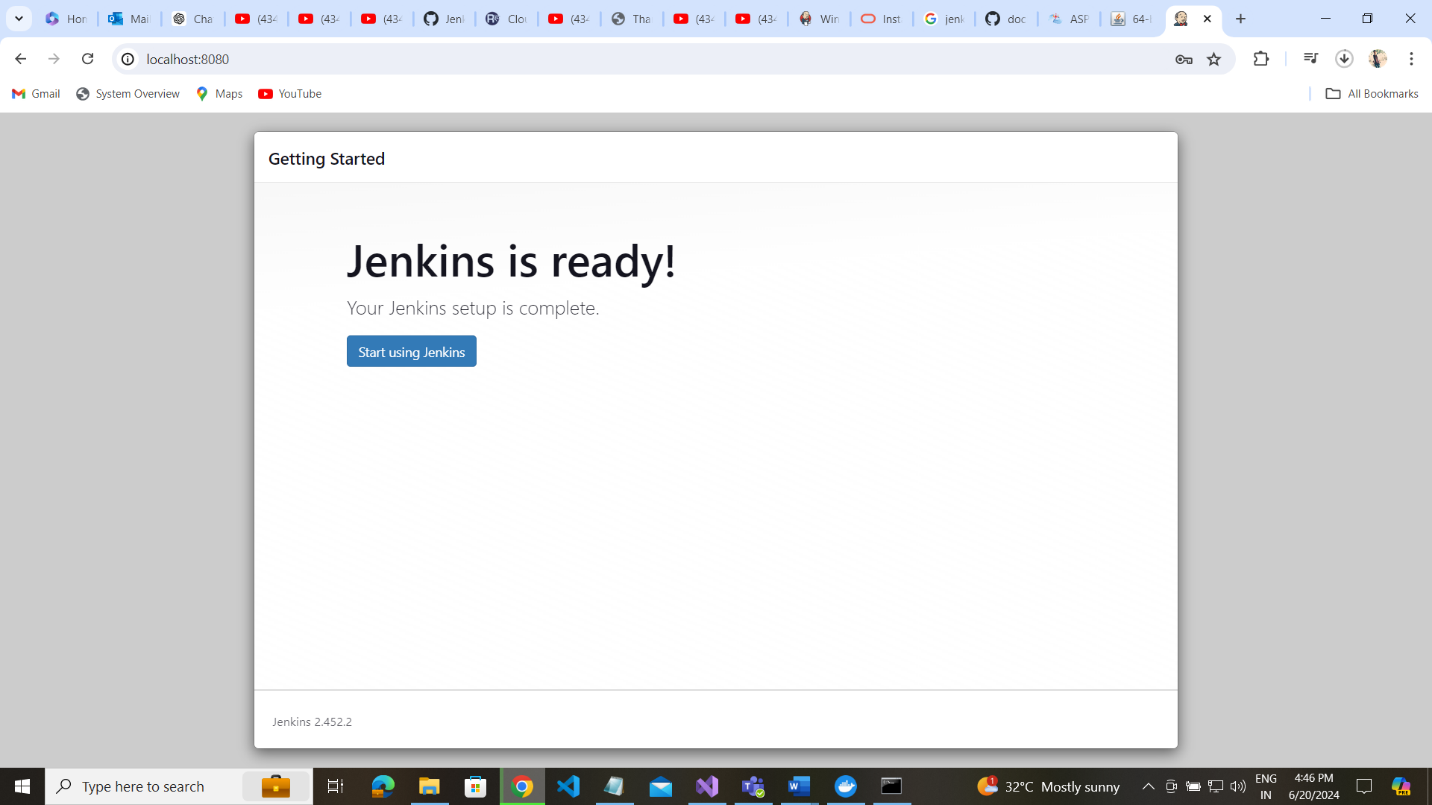
Confirm password:S@1kumar

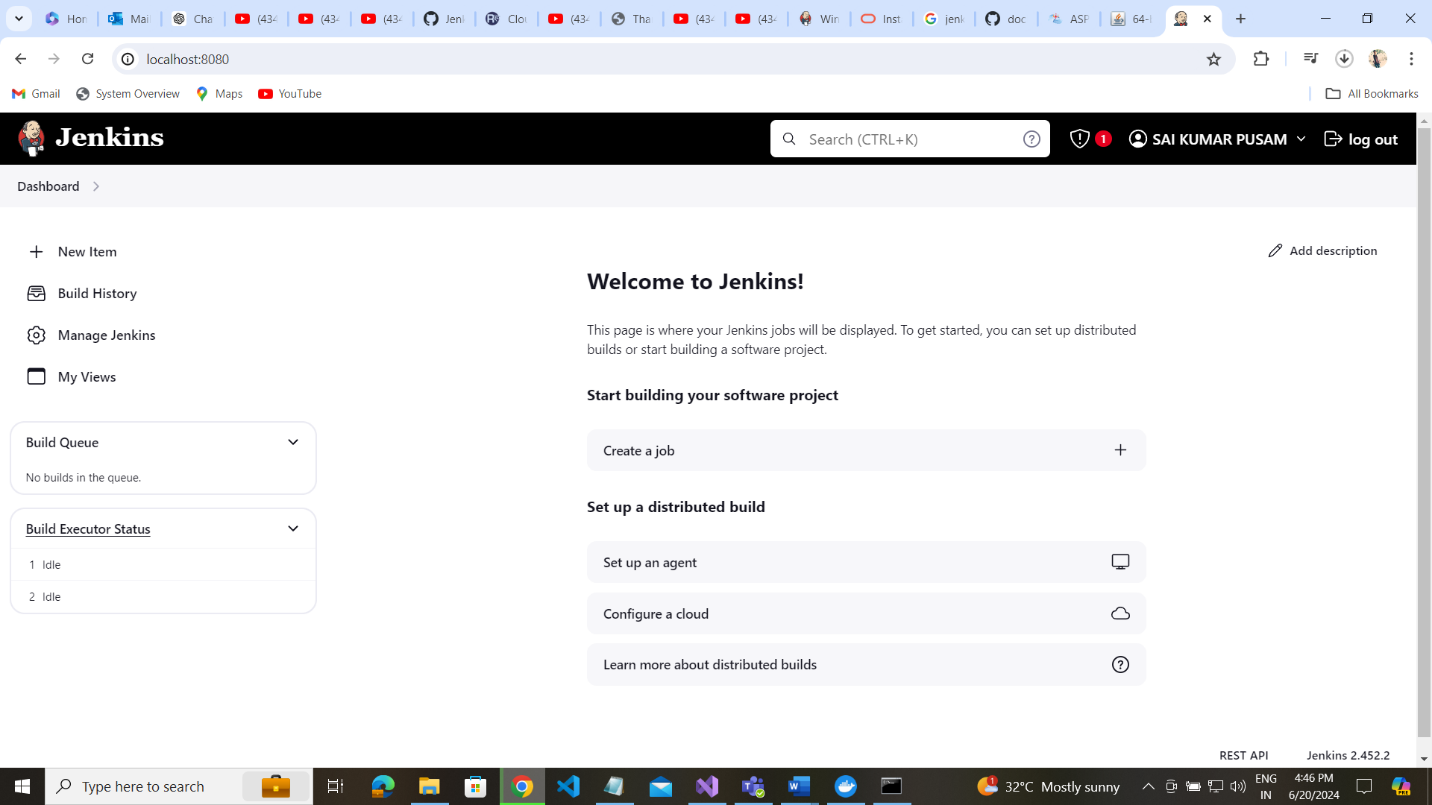
Email addredd:pusamsaikumar302@gmail.com

Then submit the form

Save url:



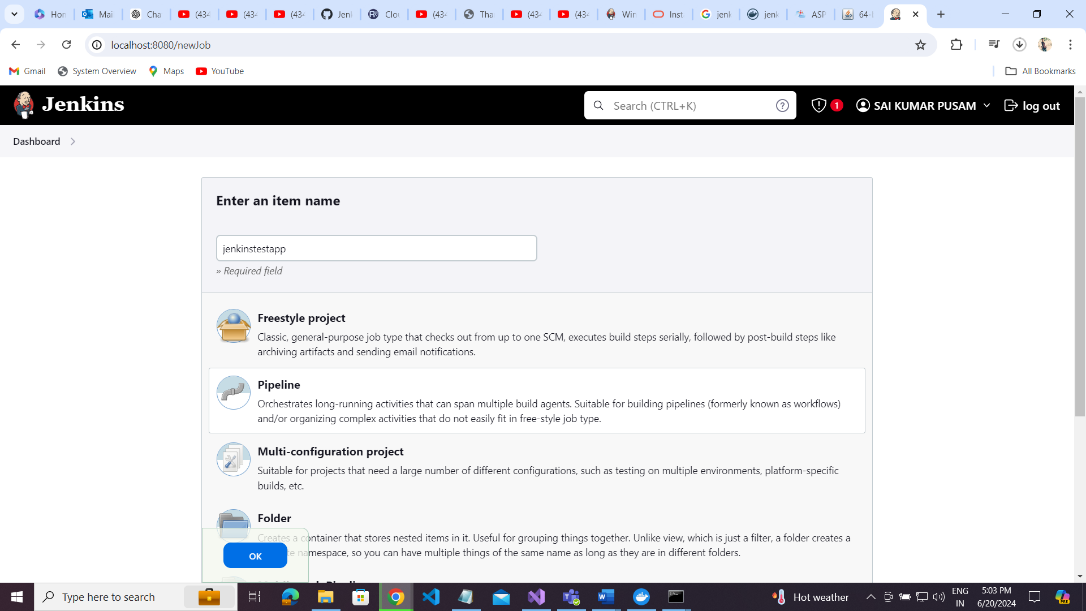




1. Create new job

Name : testjenkinsapp

Select -- pipeline options  
clike on ok button



Using Docker Container

We have to pull docker with jenkins

Error: “ it is not suitable for windows/arm 64 . its designed for linux – based contatiner we have to switch to linux docker settings ”

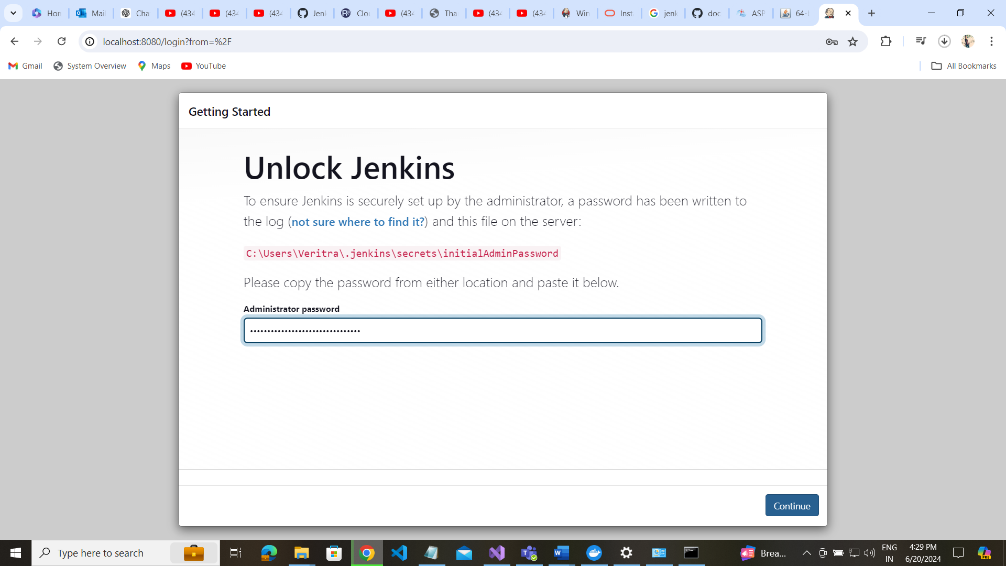
1. Docker settings --- > change to ----> Linux

C:\Users\Veritra>docker pull jenkins/jenkins

C:\Users\Veritra>docker run -p 8080:8080 -p 50000:50000 --restart=on-failure -v jenkins\_home:/var/jenkins\_home jenkins/jenkins:lts-jdk17

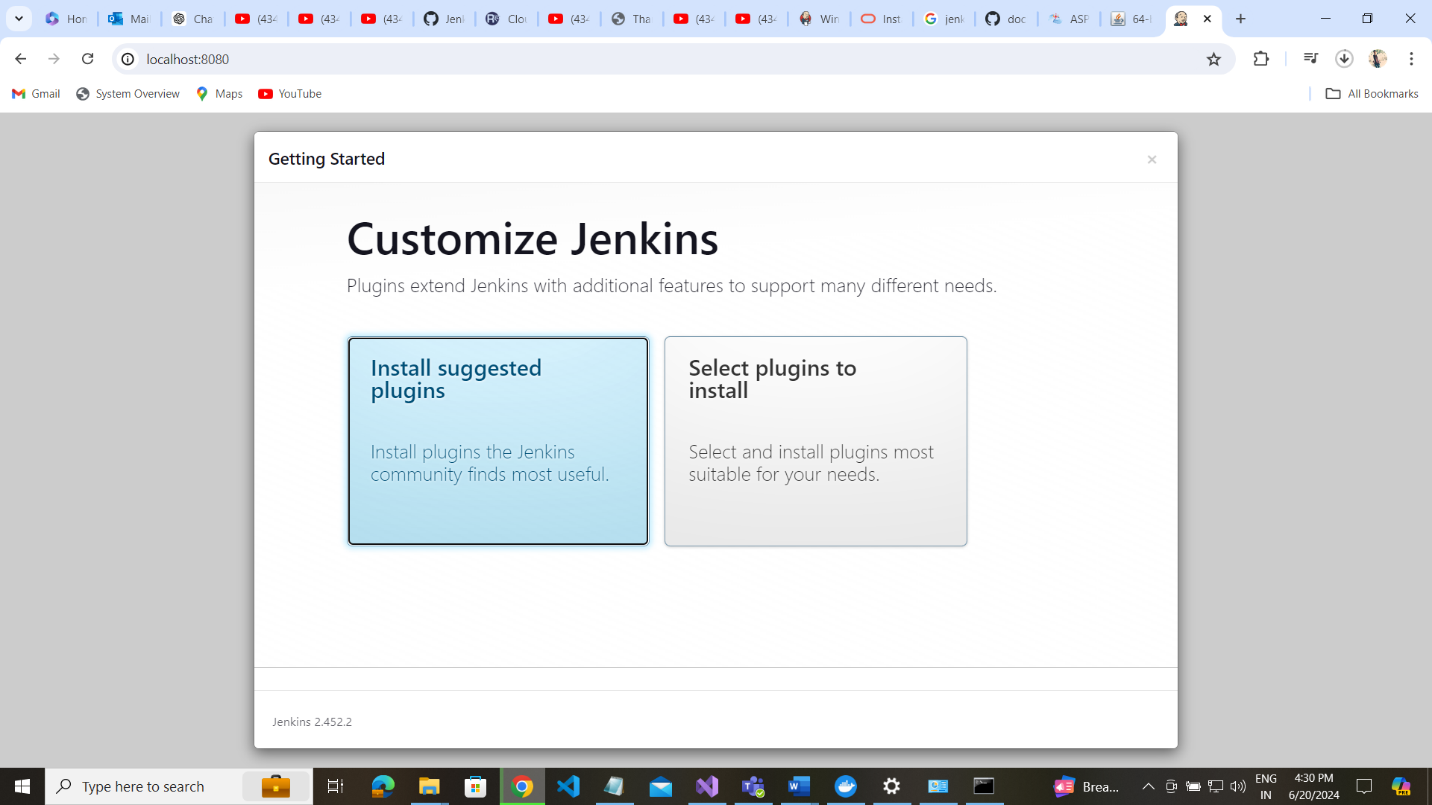
// password generated

7bf9c428db9344d8bf49129c8666da37



Paste the generated password here.

Select install suggested plugins.



Username: Saikumar

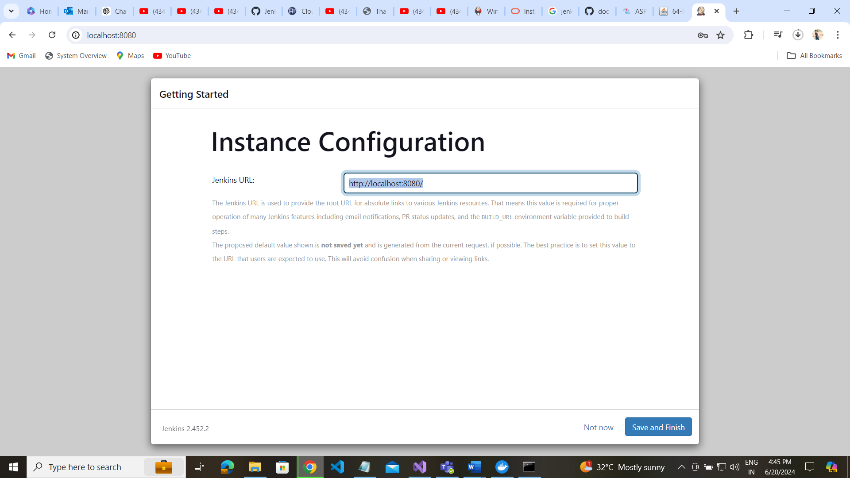
Password: S@1kumar

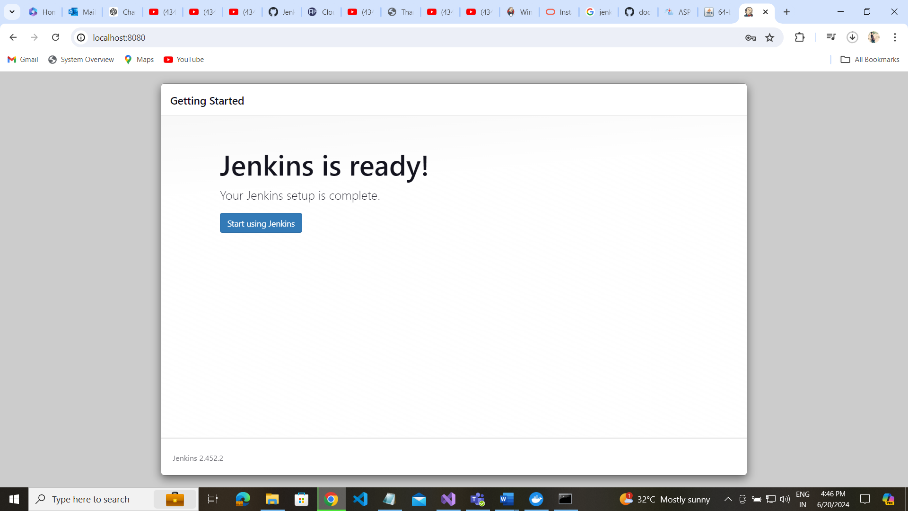
Confirm password:S@1kumar

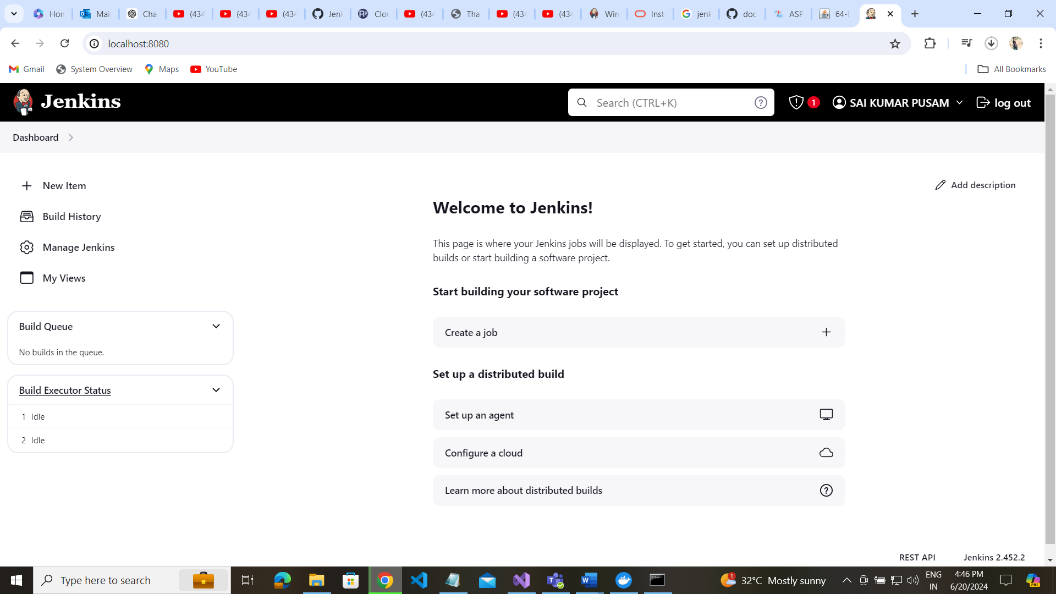
Email addredd:pusamsaikumar302@gmail.com

Then submit the form

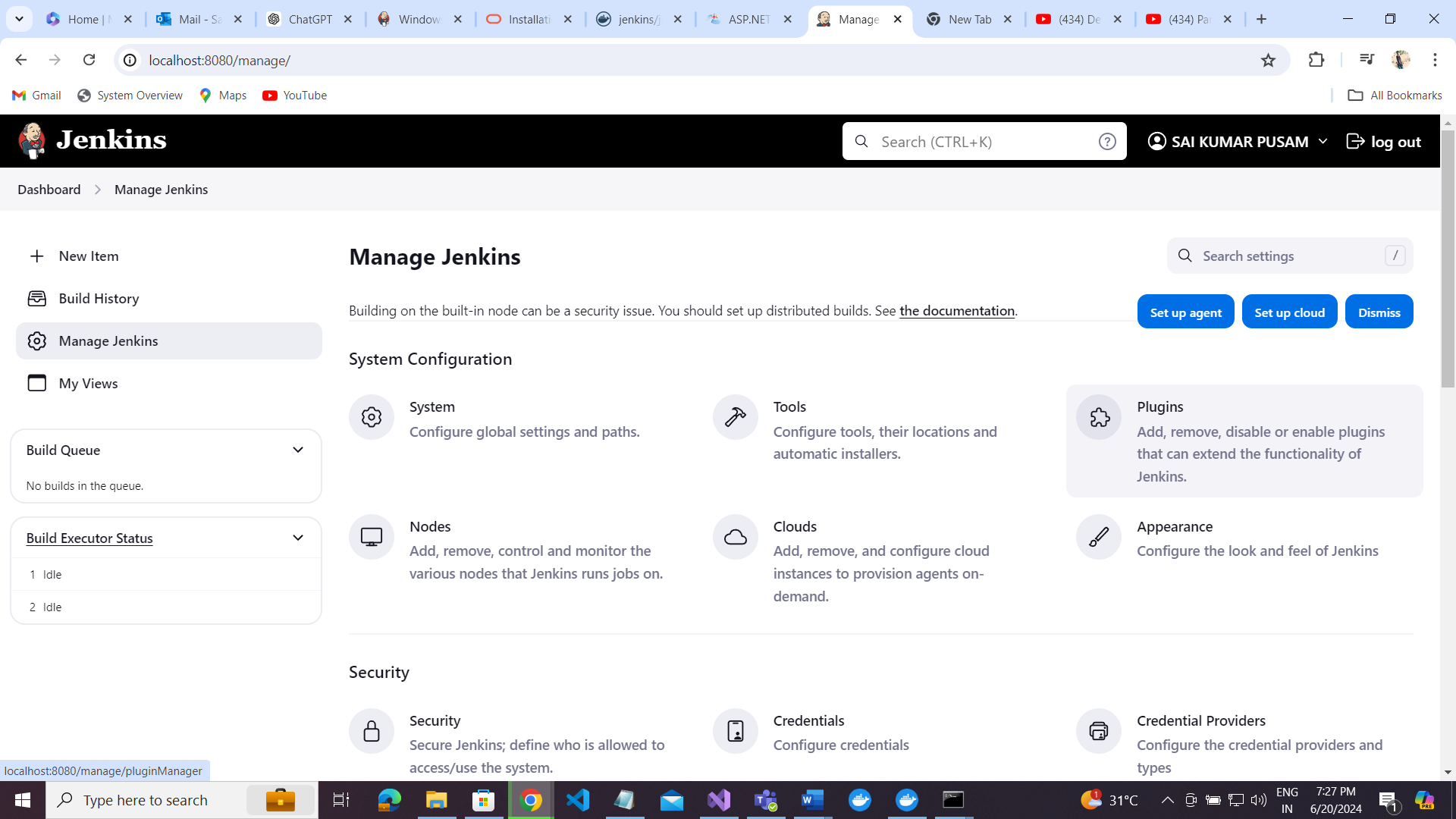
Save url:



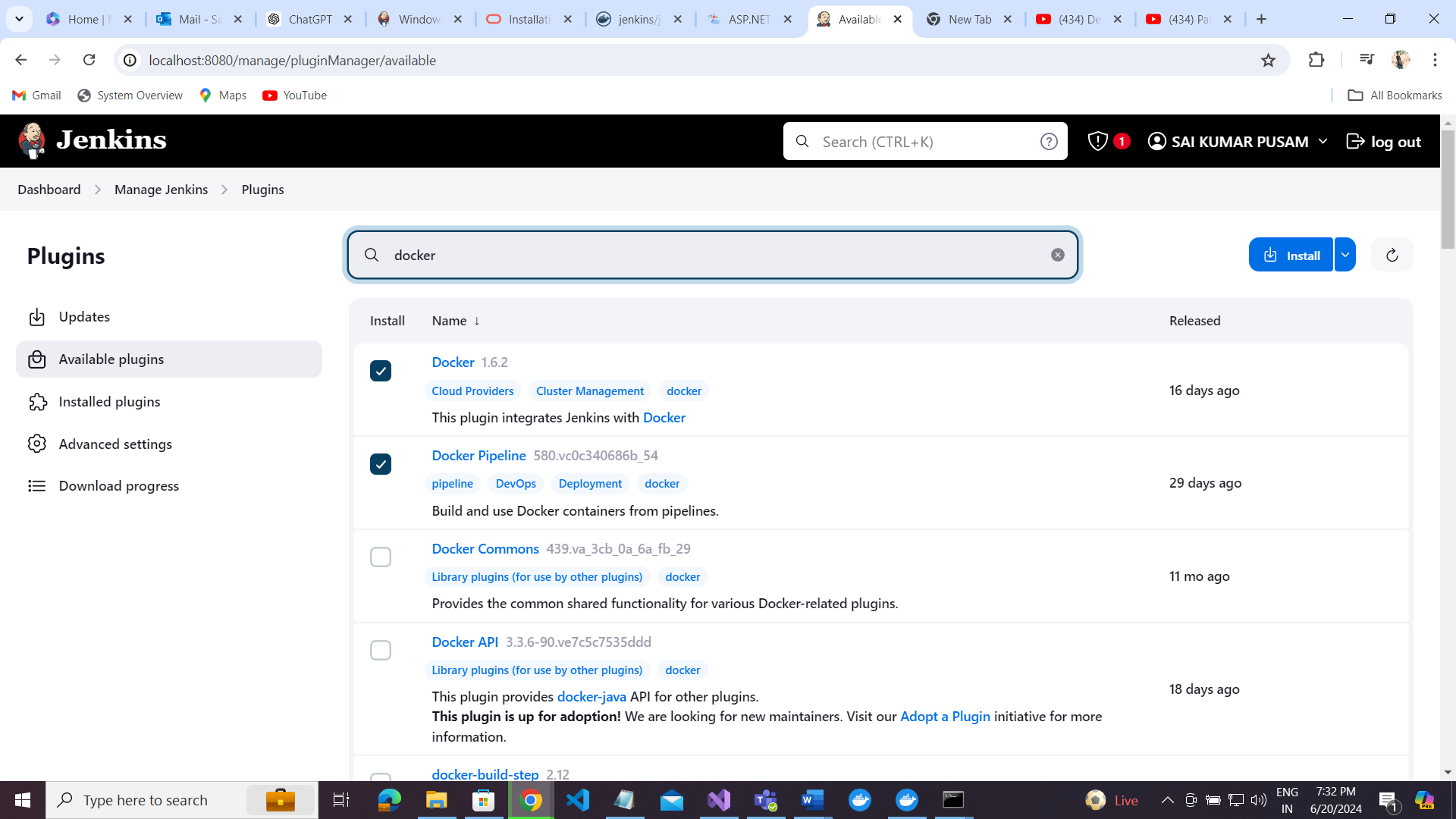




Berfore create job: manage jenkins - -- > plugins



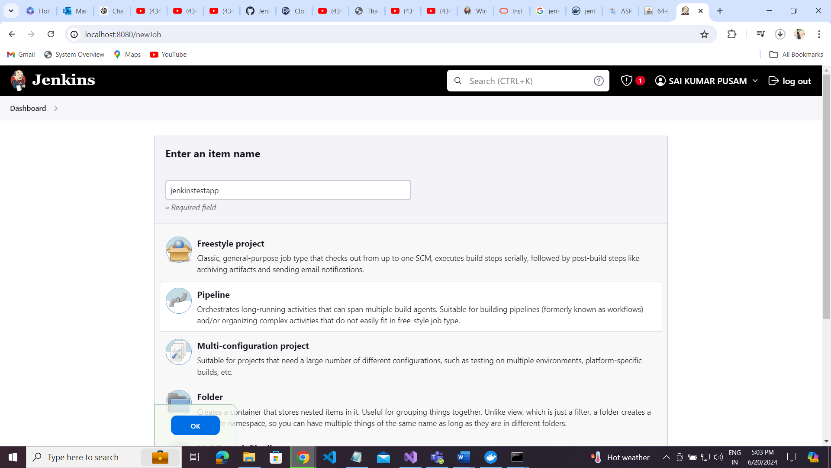
Install some jenkins: docker,docker-pipeline ---- a ddwithoutrestart



1. Create new job

Name : testjenkinsapp

Select -- pipeline options  
clike on ok button



Add pipeline

### Steps to Set Up CI/CD with Jenkins and Docker

#### 1. Set Up Jenkins

* Install necessary plugins:
  + **Docker Plugin**: Allows Jenkins to interact with Docker.
  + **Pipeline Plugin**: Provides Jenkins Pipeline support.

#### 2. Configure Jenkins Credentials

* **Docker Hub Credentials**: If pushing Docker images to Docker Hub, set up credentials in Jenkins for Docker Hub.

#### 3. Create a Jenkins Pipeline Script

* Use a Jenkinsfile to define your CI/CD pipeline. Here’s a basic example:

groovy

Copy code

pipeline {

agent any

environment {

registry = "your-docker-hub-username"

imageName = "your-image-name"

dockerfile = "./Dockerfile"

buildTag = "${env.BUILD\_NUMBER}"

}

stages {

stage('Build') {

steps {

script {

// Build the Docker image

docker.build("${registry}/${imageName}:${buildTag}", "-f ${dockerfile} .")

}

}

}

stage('Test') {

steps {

// Run tests inside the Docker container

sh "docker run --rm ${registry}/${imageName}:${buildTag} dotnet test"

}

}

stage('Push') {

steps {

script {

// Push the Docker image to Docker Hub

docker.withRegistry('https://registry.hub.docker.com', 'docker-hub-credentials') {

docker.image("${registry}/${imageName}:${buildTag}").push()

}

}

}

}

stage('Deploy') {

steps {

// Deploy to your server or Kubernetes cluster

// Example: Kubernetes deployment

sh "kubectl set image deployment/your-deployment your-container=${registry}/${imageName}:${buildTag}"

}

}

}

}

#### 4. Dockerfile for ASP.NET Core 6 Web API

* Ensure your Dockerfile is configured to build and run your ASP.NET Core Web API application. Here’s a basic example:

dockerfile

Copy code

# Use the appropriate SDK version

FROM mcr.microsoft.com/dotnet/sdk:6.0 AS build

WORKDIR /app

# Copy csproj and restore as distinct layers

COPY \*.csproj .

RUN dotnet restore

# Copy everything else and build

COPY . .

RUN dotnet publish -c Release -o out

# Build runtime image

FROM mcr.microsoft.com/dotnet/aspnet:6.0 AS runtime

WORKDIR /app

COPY --from=build /app/out .

ENTRYPOINT ["dotnet", "YourApiName.dll"]

#### 5. Configure Jenkins Pipeline Job

* Create a new Pipeline job in Jenkins.
* Link the job to your Git repository where the ASP.NET Core 6 Web API code resides.
* Use the Jenkinsfile from step 3 as your Pipeline script.
* Configure Jenkins job to trigger on Git pushes or schedule as needed.

#### 6. Run and Monitor

* Run the Jenkins Pipeline job and monitor the build/test/deploy stages.
* View console output for errors and feedback.