**You must use the following tools:**

* Docker: To build the application from a Dockerfile and push it to Docker Hub
* Docker Hub: To store the Docker image
* GitHub: To store the application code and track its revisions
* Git: To connect and push files from the local system to GitHub
* Linux (Ubuntu): As a base operating system to start and execute the project
* Jenkins: To automate the deployment process during continuous integration

**Following requirements should be met:**

* Document the step-by-step process from the initial installation to the final stage
* Track the versions of the code in the GitHub repository
* Availability of the application in the Docker Hub
* Track the build status of Jenkins for every increment of the project

To install maven I have created DockerFile and created new image .Jenkin container is created from this docker image

1.create custom docker file

FROM jenkins/jenkins:lts

# if we want to install via apt

USER root

RUN apt-get update && apt-get install -y maven

# drop back to the regular jenkins user - good practice

USER Jenkins

Text

Description automatically generated

2.run docker file

docker build -t jenkin\_maven:1.1 .

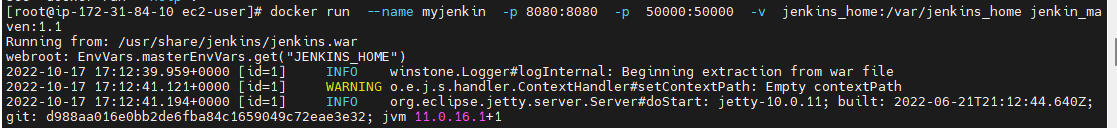
3. docker images

Text

Description automatically generated

4. execute below command to run jenkin on port 8080 from docker image created via dockerfile

docker run --name myjenkin -p 8080:8080 -p 50000:50000 -v jenkins\_home:/var/jenkins\_home jenkins\_maven:1.1



To get password for below run command

cat /jenkins\_home/secrets/initialAdminPassword

6561af64c2d54c92a8887165c8ed1519—enter this password

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

5. verify Java, maven, git in jenkin container is present

Text

Description automatically generated

6.

Update JAVA\_HOME,MAVEN\_HOME as below

JAVA\_HOME= /opt/java/openjdk

MAVAN\_HOME= /usr/share/maven

Graphical user interface, text, application

Description automatically generated

7. pipeline script is as below

pipeline {

agent any

tools {

maven 'MyMaven'

jdk 'MyJava'

}

stages {

stage ('Initialize') {

steps {

sh '''

echo "PATH = ${PATH}"

echo "JAVA\_HOME = ${JAVA\_HOME}"

'''

sh "mvn --version"

}

}

stage ('Build') {

steps {

echo 'This is a minimal pipeline.'

}

}

}

}

Graphical user interface, application

Description automatically generated

Console OUTPUT:

**Console Output**

Started by user [vihaan](http://3.88.10.34:8080/user/vihaan)

[Pipeline] Start of Pipeline

[Pipeline] node

Running on [Jenkins](http://3.88.10.34:8080/computer/(built-in)/) in /var/jenkins\_home/workspace/pipeline

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Tool Install)

[Pipeline] tool

[Pipeline] envVarsForTool

[Pipeline] tool

[Pipeline] envVarsForTool

[Pipeline] }

[Pipeline] // stage

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Initialize)

[Pipeline] tool

[Pipeline] envVarsForTool

[Pipeline] tool

[Pipeline] envVarsForTool

[Pipeline] withEnv

[Pipeline] {

[Pipeline] sh

+ echo PATH = /opt/java/openjdk/bin:/usr/share/maven/bin:/opt/java/openjdk/bin:/usr/share/maven/bin:/opt/java/openjdk/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

PATH = /opt/java/openjdk/bin:/usr/share/maven/bin:/opt/java/openjdk/bin:/usr/share/maven/bin:/opt/java/openjdk/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

+ echo JAVA\_HOME = /opt/java/openjdk

JAVA\_HOME = /opt/java/openjdk

[Pipeline] sh

+ mvn --version

[1mApache Maven 3.6.3[m

Maven home: /usr/share/maven

Java version: 11.0.16.1, vendor: Eclipse Adoptium, runtime: /opt/java/openjdk

Default locale: en, platform encoding: UTF-8

OS name: "linux", version: "5.10.135-122.509.amzn2.x86\_64", arch: "amd64", family: "unix"

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Build)

[Pipeline] tool

[Pipeline] envVarsForTool

[Pipeline] tool

[Pipeline] envVarsForTool

[Pipeline] withEnv

[Pipeline] {

[Pipeline] echo

This is a minimal pipeline.

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // node

[Pipeline] End of Pipeline

Finished: SUCCESS

8. Create pipeline with GIT Code

pipeline{

agent any

tools{

maven 'MyMaven'

jdk 'MyJava'

}

stages{

stage("checkout with priyanka today"){

steps{

git url: 'https://github.com/TheoryGit444/DevOpsClassCodes'

}

}

stage("Codecompile by priyanka"){

steps{

sh "mvn compile"

}

}

stage("Code testing by priyanka"){

steps{

sh "mvn test"

}

}

stage("qa the code"){

steps{

sh "mvn pmd:pmd"

}

}

stage("creating package"){

steps{

sh "mvn package"

}

}

}

}

Graphical user interface, application, table

Description automatically generated

Table

Description automatically generated

.war file is created at the end

Table

Description automatically generated

8. To create Docker image from the container

-Docker commit 795cac292c43 priyankasahu50/jenkinwithmaven:1.1

-Docker images

Output is as below:

A screenshot of a computer

Description automatically generated with medium confidence

9.Login to Docker as below

Text

Description automatically generated

10.To push image to docker hub ,run below command

Text

Description automatically generated

11. Login to Docker Hub and verify the image is pushed

Graphical user interface, text, application, email, website

Description automatically generated