Maven Web App Deployment using Jenkins

Step 1 : Download Maven Project from Git.
Configure the Same in Source Code M

Step 2 : Execute Shell in Build

```
> mvn clean compile exec:java
                    package
```

Step 3 : Configuration of Tomcat Server

STEP I : Change root user password

```
> sudo passwd root
    -- Type New password
```

Sudo su

Switch to the Root User

```
> ~~sudo~~ su -
```

Step II Add user and roles in Config

Path

```
> cd /opt/tomcat/config
> nano tomcat-users.xml
```

Add the below tags (Hyphen)

```
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<user username="admin"
      password="admin"
      rolename="manager-gui, manger-s
/>
```

Stop 4 Come back to Jenkins and install "Deploy to Container" plugin

Stop 5. Set the Java JDK Environment variable path in Global Tools Configuration

JDK → Not JRE

Name : Oracle_JDK
JAVA_HOME : /usr/lib/jvm/java-8-oracle

↳ get the environment Variable's value by using
↳ printenv

Stop 7. Come back to the Project Select Post Build Trigger as "Deploy war/ear to a container"

WAR/EAR Files : **/*.war
Context Path : /javawebapp
Container : Tomcat 9.x Remote
URL : http://127.0.0.1:8090
password : Refer Stop3 → Stop II

Stop 8. Verify

http://127.0.0.1:8090/ ~~we~~ javawebapp

Building & Publishing Docker image to Docker Hub

1) Build Java Project (Maven) as a <u>pipeline</u>

→ General Tab
  → GitHub Project (Where we can given a SCM)

<u>Pipeline</u>

Definition: Pipeline Script

<u>Scripts</u>

```
pipeline {
    agent any
    tools {
        maven 'MAVEN'
    }
    stages {
        stage ('Build Maven') {
            steps {
                checkout : check Out from version control
                sh "mvn clean compile package"
            }
        }
    }
}
```

(using pipeline syntax)

2) Build Docker Image

Sripts

```
pipeline{
    agent any
    stages{
        stage('Build Docker Image'){
            steps{
                scripts{
                    sh 'docker build -t devop/my-app .'
                }
            }
        }
        stage ('Push Docker Image'){
            steps{
                script{
                    withCredential : Bind ~~Variabl~~ credentials to variables
                    {
                        Binding : Secret test
                        Credential : kind : Secret test
                        Secret : Password
                        ID : username

                        sh 'docker login -u devops
                            -p ${dockerhubpwd}'
                    }
                    sh 'docker push devops/my-app'
                }
            }
        }
    }
}
```

This is from SCM

Install Plugin

1. Maven integration
2. Pipeline maven integration
3. Pipeline utility steps

⊗ Docker Permission Issue in Jenkins Solution

(1) > sudo usermod -a -G docker jenkins
=> (2) > sudo service jenkins restart
(3) (jenkins_URL)/restart

<u>Notes</u>

⊗ Docker image tag name should be

dockerUserName/ImageName: version

ex

$\boxed{\text{civildinesh04/my-app:v1}}$

Docker

1) Nginx Image Dockerfile

> docker pull nginx
> docker run -d --name nginx:v1 -p 80:80 nginx

Node JS

1) Create an empty folder
> mkdir nodewebapp
> cd nodewebapp

2) >npm init

3) >Create a dockerfile and docker ignore file
> touch Dockerfile
> touch .dockerignorefile
> touch ~~ide~~ index.js

> Open the index.js file.
> vi index.js (or) nano index.js

```
const express = require("express");
const app = express ();
app.get ("/", (req, res) => {
    res.send ("<h1> Hello Node </h1>");
});
app. listen (3000 , () => {
    console.log (" App running on port 3000...");
});
```

Dockerfile

```
FROM node: latest
WORKDIR /usr/src/app
COPY package*.json ./          ← ./ X
RUN npm install
COPY . .
EXPOSE 3000
CMD ["node", "app.js"]
```

```
> docker build -t webapp .
> docker run -d -p 3000:3000 webapp
> Sudo apt-get update
> sudo apt-get install npm
```

Nginx DockerFile

```
FROM   Ubuntu: Latest
RUN   apt-get -y update &
      apt-get -y install nginx
EXPOSE   80/tcp
CMD ["/usr/sbin/nginx", "-g", "daemon off;"]
```

Tomcat Env Path in Docker

```
CATALINA_HOME = /usr/local/tomcat
```

Docker Exec Command

```
> docker exec -ti <container_name> bash
```

Deploy ~~D~~ WebApp using Script Pipeline

Plugin

* ssh agent plugin

Pipeline Script — Invoke anothe Job

```
Stage (' Copy War File'
{
    steps {
        build job:- 'WebAppTomcat'   <- Job name
    }
}
```