

NOTE: I'm using wsl, so all commands are coming from there.

Show the created, and the repository:

```
WSL at [icon] [icon] 9ms
>> cd DevOps/miniProjects/4thWeek/
WSL at [icon] [icon] 4thWeek [icon] JavaMaven [icon] ~1 6ms
>> pwd
/home/brayanmarin/DevOps/miniProjects/4thWeek
WSL at [icon] [icon] 4thWeek [icon] JavaMaven [icon] ~1 5ms
>> ls
Dockerfile Jenkinsfile helloJavaMavenDeploy
WSL at [icon] [icon] 4thWeek [icon] JavaMaven [icon] ~1 6ms
>> cd ..
WSL at [icon] [icon] miniProjects 6ms
>> cd 4thWeek/
WSL at [icon] [icon] 4thWeek [icon] JavaMaven [icon] ~1 6ms
>> git branch
* JavaMaven
main
onlyJava
WSL at [icon] [icon] 4thWeek [icon] JavaMaven [icon] ~1 17ms
```

Installing minikube:

```
WSL at [icon] [icon] 4thWeek [icon] JavaMaven [icon] ?1 ~1 1s 691ms
>> sudo dpkg -i minikube_latest_amd64.deb
Selecting previously unselected package minikube.
(Reading database ... 68781 files and directories currently installed.)
Preparing to unpack minikube_latest_amd64.deb ...
Unpacking minikube (1.32.0-0) ...
Setting up minikube (1.32.0-0) ...
WSL at [icon] [icon] 4thWeek [icon] JavaMaven [icon] ?1 ~1 2s 67ms
>> minikube version
minikube version: v1.32.0
commit: 8220a6eb95f0a4d75f7f2d7b14cef975f050512d
WSL at [icon] [icon] 4thWeek [icon] JavaMaven [icon] ?1 ~1 144ms
```

I use SSH to connect the pipeline of Jenkinsfile to my wsl machine because in my WSL i have my cluster of minikube

SSH Pipeline Steps 2.0.68.va_d21a_12a_6476

Jenkins pipeline steps which provides SSH facilities such as command execution or file transfer for continuous delivery.

[Report an issue with this plugin](#)



This is the pipeline that I use the apps (NOTE: There are 3 Jenkinsfile, but ALL OF THEM are on different branches)

The screenshot shows the GitHub interface for the repository **BrayanMarin / Week4Minikube1**. At the top, there are buttons for **Unwatch** (1), **Star** (0), and **Fork** (0). Below this is a navigation bar with tabs for **Code**, **Issues**, **Pull Requests**, **Packages**, **Projects**, **Releases**, **Wiki**, and **Activity**. The main content area features two green notification banners: "You pushed on branch **JavaMaven** 1 hour ago" and "You pushed on branch **onlyJava** 1 hour ago", each with a **New Pull Request** button. Below these is a section for "No Description" and "Manage Topics". A summary bar shows **3 Commits**, **3 Branches**, and **0 Tags**. The **JavaMaven** branch is selected, with options to **Go to file** or **Add File**. The file list shows the **HTTP** protocol with the URL **http://localhost:3000/BrayanMari**. The commit history lists four items: a commit by **Brayan Ma...** with hash **40fb1a9f4f**, and three files: **helloJavaMavenDeploy**, **Dockerfile**, and **Jenkinsfile**, all pushed 1 hour ago.

```

1  def remote={}
2  remote.name = 'brayanmarin'
3  remote.host = '172.28.92.49'
4  remote.allowAnyHosts = true
5  pipeline{
6      environment{
7          dockerImageName = "merxxaz/helloworldjavamaven:v1.1.0"
8          dockerImage = ""
9          SSH_CREDS = credentials('23f6de3f-0ddf-4ca8-9c0d-18960773cd0e')
10     }
11     agent any
12     stages{
13         stage('SSH Connection'){
14             steps{
15                 script{
16                     remote.user = env.SSH_CREDS_USR
17                     remote.password = env.SSH_CREDS_PSW
18                 }
19             }
20         }
21         stage('Checkout Source') {
22             steps {
23                 sh 'ls'
24             }
25         }
26         stage('Build image') {
27             steps ||
28             script {
29                 // Using Docker Pipeline plugin syntax
30                 dockerImage = docker.build dockerImageName
31             }
32         }
33         stage('Setting up files'){
34             steps{
35                 sshPut(remote: remote, from:'helloJavaMavenDeploy/deployment.yml', into: '/home/brayanmarin/DevOps/miniProjects/4thWeek/helloJavaMavenDeploy/')
36                 sshPut(remote: remote, from:'helloJavaMavenDeploy/service.yml', into: '/home/brayanmarin/DevOps/miniProjects/4thWeek/helloJavaMavenDeploy/')
37             }
38         }
39         stage('Deploying API to Kubernetes'){
40             steps{
41                 sshCommand(remote: remote, command: 'minikube kubectl -- apply -f /home/brayanmarin/DevOps/miniProjects/4thWeek/helloJavaMavenDeploy/deployment.yml')
42                 sshCommand(remote: remote, command: 'minikube kubectl -- apply -f /home/brayanmarin/DevOps/miniProjects/4thWeek/helloJavaMavenDeploy/service.yml')
43                 sshCommand(remote: remote, command: 'minikube service hello-javamaven')
44             }
45         }
46     }
47 }
48

```

Now, showing the same with the deployment.yml and service.yml that as I said, there are on different branches:

```
deployment.yml helloJavaMavenDeploy/deployment.yml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: hello-javamaven #The name of the Kubernetes Deployment to be created in the Kubernetes cluster
5    labels:
6      app: hello-javamaven
7  spec:
8    replicas: 2 #The number of pods to be created in the Kubernetes cluster for the application container
9    selector:
10     matchLabels:
11       app: hello-javamaven
12    template:
13     metadata:
14       labels:
15         app: hello-javamaven
16     spec:
17       containers:
18         - name: hello-javamaven #The name of the application container
19           image: merxxaz/helloworldjavamaven:v1.1.0 #The Docker image for building the application container
20           ports:
21             - containerPort: 8003 #The port for the application container
22
```

```
service.yml helloJavaMavenDeploy/service.yml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: hello-javamaven #The name of the Kubernetes Service to be created in the Kubernetes cluster
5  spec:
6    selector:
7      app: hello-javamaven
8    type: ClusterIP #Type of the Kubernetes Service
9    ports:
10     - protocol: TCP
11       port: 8003 #Service port
12       targetPort: 8003 #The port for the application container
13
```

Executing the pipeline as a job in Jenkins:


```
WSL at [refresh] [home] [back] [forward] [search] helloJavaMavenDeploy [JavaMaven] 277ms bash [53.62%] [19:35:59]
>> kubectl get svc
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
hello-javamaven  ClusterIP     10.103.143.181 <none>         8003/TCP         117m
hello-minikube  NodePort      10.105.71.65   <none>         8081:31766/TCP   6h18m
hello-onlyjava  ClusterIP     10.106.66.170  <none>         8002/TCP         179m
kubernetes     ClusterIP     10.96.0.1      <none>         443/TCP          11h
```