

Práctica Integración Continúa

Subir código a git

Crear repositorio en git

```
git remote add origin https://github.com/fabiancontigliani/um2020.git
```

```
git status
```

```
git add .
```

```
git commit -m "deploy con jenkins"
```

```
git branch -M master
```

```
git push -u origin master
```

Crear el contenedor Jenkins

```
docker container run -d --name jenkins2 -p 8090:8080 -p 50000:50000 jenkins/jenkins
```

```
docker exec -it jenkins2 bash
```

```
cd /var/jenkins_home/secrets/
```

```
cat initialAdminPassword
```

```
7cc40b44b92e4f35ba2b1a91a0373848
```

```
jhipster ci-cd
```

<https://www.jhipster.tech/setting-up-ci-jenkins2/>

Crear proyecto Jhipster

jhipster file

Replace double quotes "" with single quotes '.

```
sh " apt-get update //... "
```

Configurar jenkins file

```
def dockerImage
```

```
stage('publish docker') {
```

```
    withCredentials([usernamePassword(credentialsId: 'myregistry-login', passwordVariable: 'DOCKER_REGISTRY_PWD',  
usernameVariable: 'DOCKER_REGISTRY_USER')) {
```

```
        sh "./mvnw -ntp jib:build"
```

```
    }
```

```
}
```

Modificar el pom

```
<to>
```

```
<auth>
```

```
<username>${DOCKER_REGISTRY_USER}</username>
```

```
<password>${DOCKER_REGISTRY_PWD}</password>
```

```
</auth>
```

```
<image>um2020/jenkinstest:latest</image>
```

```
</to>
```

Configurar Jenkins

- 1) crear nueva tarea
- 2) Nombre= build um2020
- 3) Pipeline -> Pipeline script from SCM

Configuramos repositorio y credenciales

- 4) Configurar las credenciales de dockerhub que dejamos en el POM y en el Jenking file

Administrar Jenkins -> Manage credentials

Agregar credenciales tipo username with password

Lanzar el build

Deploy Heroku

<https://www.jhipster.tech/heroku/>

jhipster heroku

What tasks/integrations do you want to include ? Deploy to ***Heroku*** (requires **HEROKU_API_KEY** set on CI service)

<https://dashboard.heroku.com/account>

git push heroku master

Configurar pipe en Heroku