

# Trabajo Práctico 7 - Servidor de Build (de integración continua).

## 1- Objetivos de Aprendizaje

- Adquirir conocimientos acerca de las herramientas de integración continua.
- Configurar este tipo de herramientas.
- Implementar procesos de construcción automatizado simples.

## 2- Unidad temática que incluye este trabajo práctico

Este trabajo práctico corresponde a la unidad N°: 3 (Libro Continuous Delivery: Cap 3)

## 3- Consignas a desarrollar en el trabajo práctico:

- Para una mejor evaluación del trabajo práctico, incluir capturas de pantalla de los pasos donde considere necesario.

## 4- Desarrollo:

### 1- Poniendo en funcionamiento Jenkins

- Bajar la aplicación y ejecutarla (ejemplo para Linux):

```
export JENKINS_HOME=~/.jenkins
```

```
mkdir -p $JENKINS_HOME  
cd $JENKINS_HOME
```

```
wget http://mirrors.jenkins.io/war-stable/latest/jenkins.war
```

```
java -jar jenkins.war --httpPort=8081  
pass 27dc5a1660f245009b4c1e02abb63db9
```

Please use the following password to proceed to installation:

27dc5a1660f245009b4c1e02abb63db9

This may also be found at: /home/juan-pablo/jenkins/secrets/initialAdminPassword

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

```
2022-09-20 18:11:12.673+0000 [id=37]    INFO    jenkins.InitReactorRunner$1#onAt
tained: Completed initialization
2022-09-20 18:11:12.686+0000 [id=24]    INFO    hudson.lifecycle.Lifecycle#onRea
dy: Jenkins is fully up and running
2022-09-20 18:11:13.476+0000 [id=66]    INFO    h.m.DownloadService$Downloadable
#load: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
2022-09-20 18:11:13.476+0000 [id=66]    INFO    hudson.util.Retrier#start: Reali
zada la acción check updates server con éxito en el intento #1
2022-09-20 18:11:13.478+0000 [id=66]    INFO    hudson.model.AsyncPeriodicWork#l
ambda$doRun$1: Finished Download metadata. 18.112 ms
```



# Bienvenido a Jenkins

Plugins extend Jenkins with additional features to support many different needs.

## Install suggested plugins

Install plugins the Jenkins community finds most useful.

## Select plugins to install

Select and install plugins most suitable for your needs.

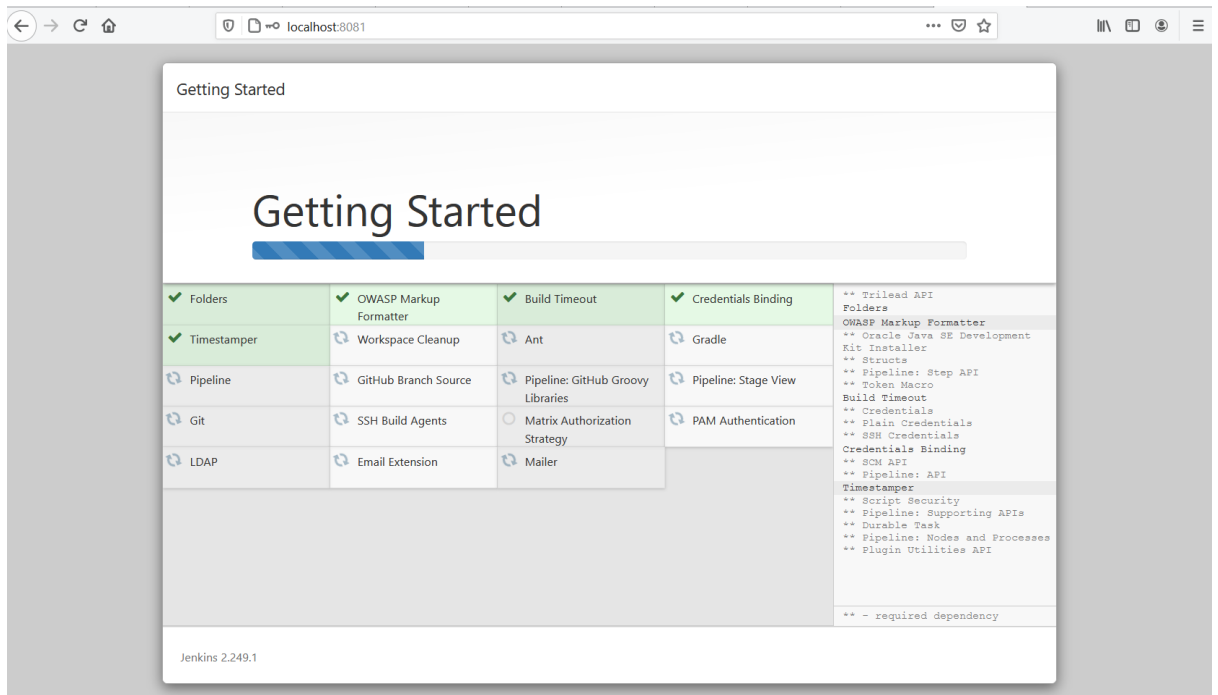


Jenkins 2.361.1

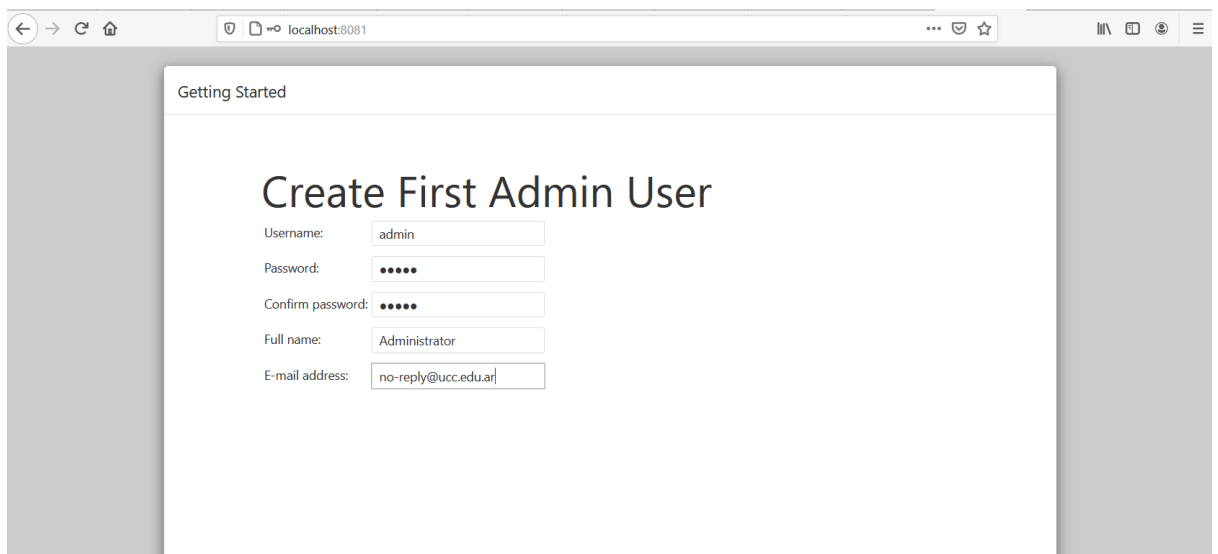
- Se puede también ejecutar en contenedor de Jenkins (pero para construir imágenes de Docker, el proceso se complica un poco):
- Una vez en ejecución, abrir <http://localhost:8081>
- Inicialmente deberá especificar el texto dentro del archivo  
~/jenkins/secrets/initialAdminPassword

cat ~/jenkins/secrets/initialAdminPassword

- Instalar los plugins por defecto



- Crear el usuario admin inicial. Colocar cualquier valor que considere adecuado.




- Se aconseja persistir la variable **JENKINS\_HOME**, ya sea por ejemplo en .bashrc o en las variables de entorno de Windows.

# Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

**Jenkins**

Q Buscar (CTRL+K) ? ! 1

Panel de Control >

+ Nuevo Tarea

Personas

Historial de Construcción

Administrar Jenkins

Mis vistas

Trabajos en cola

No hay trabajos en cola

Estado del Ejecutor de Tareas

1 Inactivo

2 Inactivo

[añadir descripción](#)

## Bienvenido a Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

### Start building your software project

Create a Job →

### Set up a distributed build

## 2- Conceptos generales

- Junto al Jefe de trabajos prácticos:
- Explicamos los diferentes componentes que vemos en la página principal:
  - En la pagina principal tenemos varias opciones, en el lado izquierdo tenemos un panel de control que nos muestra:

Panel de Control >

+ Nuevo Tarea

Personas

Historial de Construcción

Administrar Jenkins


Mis vistas

- 
- 


- **nueva tarea:** muestra como crear nuevos procesos,items,jobs,pipelines y determina como va a actuar jenkins ante algun cambio

Enter an item name


» This field cannot be empty, please enter a valid name

**Crear un proyecto de estilo libre**


Esta es la característica principal de Jenkins, la de ejecutar el proyecto combinando cualquier tipo de repositorio de software (SCM) con cualquier modo de construcción o ejecución (make, ant, mvn, rake, script ...). Por tanto se podrá tanto compilar y empaquetar software, como ejecutar cualquier proceso que requiera monitorización.

**Pipeline**


Gestiona actividades de larga duración que pueden abarcar varios agentes de construcción. Apropiado para construir pipelines (conocidas anteriormente como workflows) y/o para la organización de actividades complejas que no se pueden articular facilmente con tareas de tipo freestyle.

**Crear un proyecto multi-configuración**


Adecuado para proyectos que requieran un gran número de configuraciones diferentes, como testear en multiples entornos, ejecutar sobre plataformas concretas, etc.

**Folder**

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**Multibranch Pipeline**

Creates a set of Pipeline projects according to detected branches in one SCM repository.

**Organization Folder**

Creates a set of multibranch project subfolders by scanning for repositories.

OK

- - **Personas:** se pueden administrar y gestionar lls usuarios conocidos, identidades con acceso al dominio de seguridad actual, etc
- Personas**

Incluye todos los "usuarios" conocidos, incluyendo las identidades de acceso que se pueden enumerar en el dominio de seguridad actual, así como las personas mencionadas en los mensajes de confirmación en los registros de cambios registrados.

ID del usuario	Nombre	Last Commit Activity ↑	En
 <a href="#">juanrojas09</a>	juan pablo	N/A	

- 
- **Historial de construccion:** lleva a cabo un registro de las tareas ejecutadas de jenkins.
- **Administrar jenkins:** sirve para configurar el sistema, configurar herramientas e instalaciones, nodos, plugins, administracion en base a la seguridad, el estado de la informacion, logs, trouble shooting, acciones y herramientas a llevar a cabo

## Administrar Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#)

Set up agent

Set up cloud

Dismiss

### System Configuration



#### Configurar el Sistema

Configurar variables globales y rutas.



#### Global Tool Configuration

Configure tools, their locations and automatic installers.



#### Administrar Plugins

Añadir, borrar, desactivar y activar plugins que extienden la funcionalidad de Jenkins.



#### Administrar Nodos

Añadir, borrar, gestionar y monitorizar los nodos sobre los que Jenkins ejecuta tareas.

### Security



#### Configuración global de la seguridad

Seguridad en Jenkins. Define quién tiene acceso al sistema (autenticación) y qué puede hacer (autorización)



#### Manage Credentials

Configure credentials



#### Configure Credential Providers

Configure the credential providers and types



#### Gestión de usuarios

Crear/borrar/editar usuarios que puedan utilizar Jenkins

### Status Information



#### Información del sistema

Muestra información del entorno que puedan ayudar a la solución de problemas.



#### System Log

El log del sistema captura la salida de la clase `java.util.logging` en todo lo relacionado con Jenkins.



#### Estadísticas de Carga

Comprobar la utilización de los recursos y comprobar si es necesario añadir nuevos nodos para la ejecución de tareas.



#### Acerca de Jenkins

Eche un vistazo a la información sobre la versión y la licencia.

### Troubleshooting



#### Datos antiguos

Scrub configuration files to remove remnants from old plugins and earlier versions.

### Tools and Actions



#### Actualizar configuración desde el disco duro.

Descartar todos los datos cargados en memoria y actualizar todo nuevamente desde los ficheros del sistema. Útil cuando se modifican ficheros de configuración directamente en el disco duro.



#### Jenkins CLI

Accede y administra Jenkins desde la consola, o desde scripts



#### Consola de scripts

Ejecutar script para la administración, diagnóstico y solución de problemas.



#### Preparar Jenkins para apagar el contenedor

Detener la ejecución de nuevas tareas para que el sistema pueda apagarse de manera segura.

- Analizamos las opciones de administración de Jenkins

## 3- Instalando Plugins y configurando herramientas

- En Administrar Jenkins vamos a la sección de Administrar Plugins



## Plugin Manager

Actualizar Disponible Instalado Avanzado

🔍 Filtrar

<input type="checkbox"/>	Nombre ↓	Released	Instalado
No hay actualizaciones pendientes			

- 
- De la lista de plugins disponibles instalamos **Docker Pipeline**

## Plugin Manager

Actualizar Disponible Instalado Avanzado

🔍 Docker Pipeline

Install	Name ↓	Released
<input type="checkbox"/>	<b>Docker Pipeline</b> 521.v1a_a_dd2073b_2e pipeline DevOps Deployment docker Build and use Docker containers from pipelines.	Hace 1 Mes 12 días

- 
- Instalamos sin reiniciar el servidor.

## Instalando Plugins/Actualizaciones

### Preparación

- Probando conectividad con Internet
- Probando conectividad con jenkins-ci.org
- Correcto

Authentication Tokens API	✓ Actualizado
Docker Commons	✓ Actualizado
Docker Pipeline	✓ Actualizado
Loading plugin extensions	✓ Success

→ [Volver al inicio de la página](#)  
(puedes empezar a utilizar los plugins instalados inmediatamente)

- → ☐ Reiniciar Jenkins cuando se complete la instalación y no haya trabajos corriendo
- Abrir nuevamente página de Plugins y explorar la lista, para familiarizarse qué tipo de plugins hay disponibles.

Install	Name ↓	Released
<input type="checkbox"/>	<b>Git server</b> 99.va_0826a_b_cdfa_d <a href="#">git</a> <a href="#">Library plugins (for use by other plugins)</a> Allows Jenkins to act as a Git server.	Hace 1 Mes 23 días
<input type="checkbox"/>	<b>Pipeline: Deprecatcd Groovy Libraries</b> 588.v576c103a_ff86 <a href="#">Varios</a> Hosting of Pipeline Groovy libraries inside a Jenkins Git server. <b>Deprecatcd.</b> Use <a href="#">Pipeline: Groovy Libraries</a> instead. If you see this plugin installed just because you upgraded, you can probably uninstall it now. This plugin should only be used if you have historically <i>pushed</i> libraries to a Git server inside Jenkins.	Hace 3 Mes 29 días
<input type="checkbox"/>	<b>Lockable Resources</b> 2.16 <a href="#">pipeline</a> <a href="#">Plugins para la administración de clusters y trabajos distribuidos</a> <a href="#">Plugins para el control de nodos</a> This plugin allows to define external resources (such as printers, phones, computers) that can be locked by builds. If a build requires an external resource which is already locked, it will wait for the resource to be free.	Hace 2 Mes 0 días
<input type="checkbox"/>	<b>Javadoc</b> 226.v71211feb_e7e9 This plugin adds Javadoc support to Jenkins.	Hace 1 Mes 12 días

- 
- En la sección de administración abrir la opción de configuración de herramientas
- Agregar maven con el nombre de **M3** y que se instale automáticamente.

≡

Maven

×

Nombre

M3

☒ Instalar automáticamente ?

≡

Instalar desde Apache

×

Versión

3.8.6

▼

Añadir un instalador ▼

Añadir Maven

- 

#### 4- Creando el primer Pipeline Job

- Crear un nuevo ítem, del tipo Pipeline con nombre **hello-world**

Enter an item name

hello-world

» Required field

-

- Una vez creado el job, en la sección Pipeline seleccionamos **try sample Pipeline** y luego **Hello World**

## Pipeline

### Definition

### Pipeline script

Script ?

```
1 pipeline {
2     agent any
3
4     stages {
5         stage('Hello') {
6             steps {
7                 echo 'Hello World'
8             }
9         }
10    }
11 }
12
```

☒ Use Groovy Sandbox ?


## Pipeline Syntax

Guardar

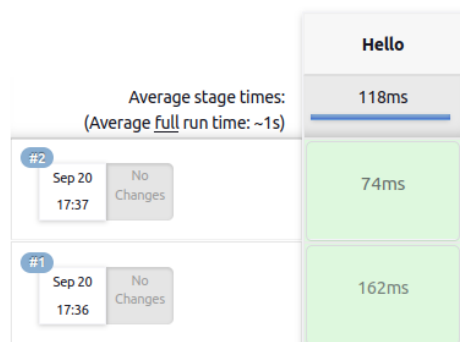
### Apply

- Guardamos y ejecutamos el Job

▶ Construir ahora

- 
- Analizar la salida del mismo

## Stage View



-

Stage Logs (Hello)

Print Message -- Hello World (self time 5ms)

Hello World

#1

20 sep. 2022 17:36

Atom feed para todos

Atom feed para fallos

Pipeline hello-world

añadir descripción

Desactivar el Proyecto

</>

Recent Changes

Stage View

Average stage times:

(Average full run time: ~1s)

#2

Sep 20 17:37

No Changes

#1

Sep 20 17:36

No Changes

Success Hello

Logs

74ms

162ms

## 5- Creando un Pipeline Job con Git y Maven

- Similar al paso anterior creamos un ítem con el nombre **simple-maven**

Enter an item name

» Required field

Crear un proyecto de estilo libre

Esta es la característica principal de Jenkins, la de ejecutar el proyecto combinando cualquier tipo de repositorio de software (SCM) con cualquier modo de construcción o ejecución (make, ant, mvn, rake, script ...). Por tanto se podrá tanto compilar y empaquetar software, como ejecutar cualquier proceso que requiera monitorización.

Pipeline

Gestiona actividades de larga duración que pueden abarcar varios agentes de construcción. Apropiado para construir pipelines (conocidas anteriormente como workflows) y/o para la organización de actividades complejas que no se pueden articular fácilmente con tareas de tipo freestyle.

Crear un proyecto multi-configuración

- Elegir **Git + Maven** en la sección **try sample Pipeline**

## Pipeline

### Definition

Pipeline script

#### Script ?

```
1 pipeline {
2   agent any
3
4   tools {
5     // Install the Maven version configured as "M3" and add it to the path.
6     maven "M3"
7   }
8
9   stages {
10    stage('Build') {
11      steps {
12        // Get some code from a GitHub repository
13        git 'https://github.com/jglick/simple-maven-project-with-tests.git'
14
15        // Run Maven on a Unix agent.
16        sh "mvn -Dmaven.test.failure.ignore=true clean package"
17
18        // To run Maven on a Windows agent, use
19        // bat "mvn -Dmaven.test.failure.ignore=true clean package"
```

GitHub + Maven

☒ Use Groovy Sandbox ?

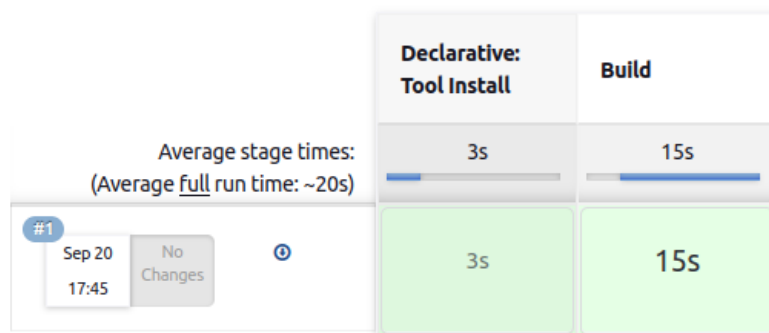
[Pipeline Syntax](#)

Guardar

Apply

- 
- Guardar y ejecutar el Job
- Analizar el script, para identificar los diferentes pasos definidos y correlacionarlos con lo que se ejecuta en el Job y se visualiza en la página del Job.

### Stage View



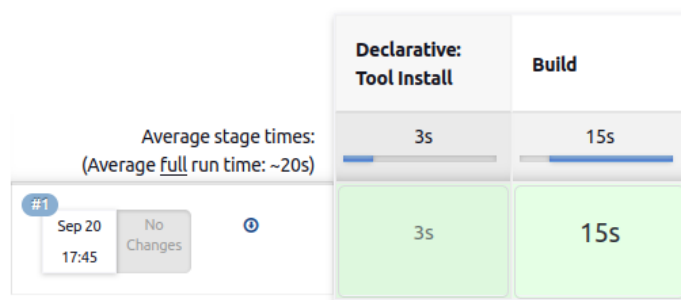
- - Script: stage view, vemos dos stages, el declarative y el build, en el build instala maven mediante la herramienta definida anteriormente M3 que instalaba el recurso automáticamente, luego hace fetch y se baja el proyecto de test de maven desde github, después corre maven (comienza la parte de post del script) y guarda los resultados del test mediante junit(sirve para tests, como xunit) con los resultados y por ultimo guarda los archivos que se

generaron en la carpeta target en jenkins para después usarlos, en archiveArtifacts

- **Resultado:**

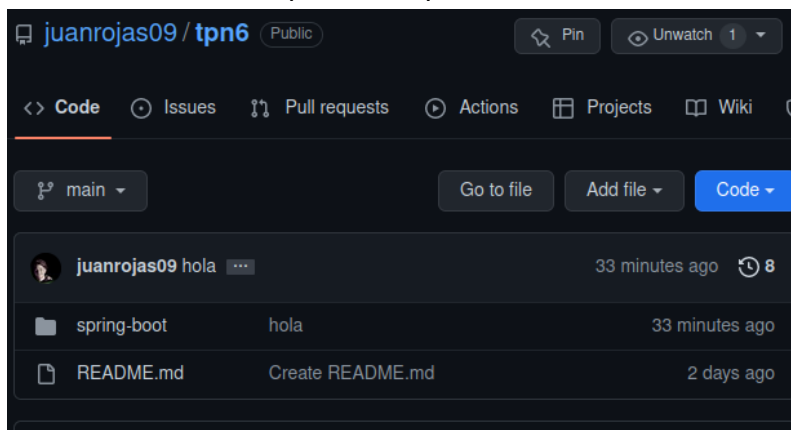


### Stage View



## 6- Utilizando nuestros proyectos

- Utilizando lo aprendido en el ejercicio 5
- Crear un Job que construya el proyecto **spring-boot** del [trabajo práctico 6](#).
- Obtener el código desde el repositorio de cada alumno (se puede crear un repositorio nuevo en github que contenga solamente el proyecto maven).
- En mi caso cree un repo con lo que necesitaba



- Generar y publicar los artefactos que se producen.
- Como resultado de este ejercicio proveer el script en un archivo **spring-boot/Jenkinsfile**

## Script ?

```
1 pipeline{
2   agent any
3   tools{
4     maven "M3"
5   }
6   stages {
7     stage('Build'){
8       steps{
9         //traerme el proyecto de spring boot
10        //con mi repo personal, al fetchear me salia algun error con las credenciales
11        //siempre al hacer push me pide contraseña, por lo que al no configurar eso con
12        //puede que no me detecte, como solucion al momento, utilizo el repo de mi comp
13        //proyecto debido a mi problema de credencial con mi git.
14        git 'https://github.com/juanrojas09/tpn6.git'
15        sh "ls"
16        dir('tp6/spring-boot'){
17          sh("mvn clean package spring-boot:repackage")
18        }
19      }
20    }
21  }
22  post{
23    success{
24      dir('tp6/spring-boot'){
25        archiveArtifacts 'target/*.jar'
26      }
27    }
28  }
29 }
30 }
31 }
32 }
33 }
34 }
35 }
36 }
37 }
```

despues de un par de intentos, solucionando el problema de mi git y mis credenciales, salio.

Status

</> Changes

► Construir ahora

⚙️ Configurar

🗑️ Eliminar Pipeline

🔍 Full Stage View

✎ Rename

🔍 Pipeline Syntax

Pipeline spring-boot

</> Recent Changes

Stage View

Historia de tareas

tendencia ▼

Filter builds...

#38 20 sep. 2022 19:46

#37 20 sep. 2022 19:44

#36 20 sep. 2022 19:42

#35 20 sep. 2022 19:42

#34 20 sep. 2022 19:41

#33 20 sep. 2022 19:40

#32 20 sep. 2022 19:39

#31 20 sep. 2022 19:38

#30 20 sep. 2022 19:37

#29 20 sep. 2022 19:36

#28 20 sep. 2022 19:35

#27 20 sep. 2022 19:34

#26 20 sep. 2022 19:33

Average stage times:  
(Average full run time: ~5s)

#38 Sep 20 19:46 No Changes

#37 Sep 20 19:44 No Changes

#36 Sep 20 19:42 No Changes

#35 Sep 20 19:42 No Changes

#34 Sep 20 19:41 No Changes

#33 Sep 20 No

Declarative: Tool Install	Build
54ms	2s
48ms	4s
51ms	7s failed
52ms	811ms failed
54ms	815ms failed
51ms	2s failed

## 7- Utilizando nuestros proyectos con Docker

- Extender el ejercicio 6
- Generar y publicar en Dockerhub la imagen de docker además del Jar.
- Se puede utilizar el [plugin de docker](#) o comandos de shell.
- No poner usuario y password en el pipeline en texto plano, por ejemplo para conectarse a DockerHub, utilizar [credenciales de jenkins](#) en su lugar.
- Como resultado de este ejercicio proveer el script en un archivo **spring-boot/Jenkinsfile**
- Referencia: <https://tutorials.releaseworksacademy.com/learn/building-your-first-docker-image-with-jenkins-2-guide-for-developers>
- [En el script trate de usarlo con el plugin de docker pero tardaba mucho entonces use sh y los comandos usados en el anterior tp para subirlo a docker hub y buildear la imagen](#)



```

1 pipeline {
2   agent any
3   tools {
4     naven "M3"
5   }
6   stages {
7     stage('Build') {
8       steps {
9         //traerme el proyecto de spring boot
10        //con mi repo personal, al fetchear me salia algun error con las credenciales
11        //siempre al hacer push me pide contraseña, por lo que al no configurar eso con la credencial
12        //puede que no me detecte, como solucion al momento, utilizo el repo de mi compañero para acceder a los archivos del
13        //proyecto debido a mi problema de credencial con mi git.
14        git 'https://github.com/juanrojas09/tpn6.git'
15        sh "ls"
16        dir('tp6/spring-boot') {
17
18          sh("mvn clean package spring-boot:repackage")
19        }
20
21      }
22      post {
23        success {
24          dir('tp6/spring-boot') {
25            archiveArtifacts 'target/*.jar'
26          }
27        }
28      }
29    }
30  }
31  stage('Build image') {
32    environment {
33      //usuario y contra del docker
34      DOCKER_CREDS = credentials('0e549c42-67a5-4259-b7d4-3b5ed0bf0209')
35    }
36    steps {
37      dir('tp6/spring-boot') {
38        // la funcion credentials te genera el usr y psw
39        sh 'docker login -u $DOCKER_CREDS_USR -p $DOCKER_CREDS_PSW'
40
41        //bildeo la imagen de docker
42        sh 'docker build -t "test-node" .'
43        sh 'docker tag test-node juanrojas09/test-node:latest'
44        sh 'docker push juanrojas09/test-node:latest'
45      }
46    }
47    /*steps{
48      dir('tp6/spring-boot'){
49        sh 'ls'
50        script{
51          def img=docker.build("test-spring-boot")
52          img.inside{
53            sh "make test"
54          }
55        }
56      }
57    }
58
59    */
60  }
61 }
62 }
63 }
64 }
65 }

```

```
Step 1/13 : FROM maven:3.5.2-jdk-8-alpine AS MAVEN_TOOL_CHAIN
--> 293423a981a7
Step 2/13 : COPY pom.xml /tmp/
--> Using cache
--> 3c818f8ac207
Step 3/13 : RUN mvn -B dependency:go-offline -f /tmp/pom.xml -s /usr/share/maven/ref/settings-docker.xml
--> Using cache
--> 528070ef64d1
Step 4/13 : COPY src /tmp/src/
--> Using cache
--> 828f2c91ca77
Step 5/13 : WORKDIR /tmp/
--> Using cache
--> 59f073066316
Step 6/13 : RUN mvn -B -s /usr/share/maven/ref/settings-docker.xml package
--> Using cache
--> af6058ba0adc
Step 7/13 : FROM openjdk:8-jre-alpine
--> f7a292bbb70c
Step 8/13 : EXPOSE 8080
--> Using cache
--> 106814f08b06
Step 9/13 : RUN mkdir /app
--> Using cache
--> eb212b41334e
Step 10/13 : COPY --from=MAVEN_TOOL_CHAIN /tmp/target/*.jar /app/spring-boot-application.jar
--> Using cache
--> cbeld782b04d
Step 11/13 : ENV JAVA_OPTS="-Xms32m -Xmx128m"
--> Using cache
--> 736607b3e53c
Step 12/13 : ENTRYPOINT exec java $JAVA_OPTS -Djava.security.egd=file:/dev/./urandom -jar /app/spring-boot-application.jar
--> Using cache
--> 512cf2f33c11
Step 13/13 : HEALTHCHECK --interval=1m --timeout=3s CMD wget -q -T 3 -s http://localhost:8080/actuator/health/ || exit 1
--> Using cache
--> f40bcfd78d8f
Successfully built f40bcfd78d8f
Successfully tagged test-node:latest
[Pipeline] sh
+ docker tag test-node juanrojas09/test-node:latest
[Pipeline] sh
+ docker push juanrojas09/test-node:latest
The push refers to repository [docker.io/juanrojas09/test-node]
a496e07f264f: Preparing
1ce9888d8bbf: Preparing
edd61588d126: Preparing
9b9b7f3d56a0: Preparing
f1b5933fe4b5: Preparing
edd61588d126: Mounted from library/openjdk
f1b5933fe4b5: Mounted from library/openjdk
```

```
+ docker push juanrojas09/test-node:latest
The push refers to repository [docker.io/juanrojas09/test-node]
a496e07f264f: Preparing
1ce988d8bbf: Preparing
edd61588d126: Preparing
9b9b7f3d56a0: Preparing
f1b5933fe4b5: Preparing
edd61588d126: Mounted from library/openjdk
f1b5933fe4b5: Mounted from library/openjdk
9b9b7f3d56a0: Mounted from library/openjdk
1ce988d8bbf: Pushed
a496e07f264f: Pushed
latest: digest: sha256:aed1c96dff125cbb0930c3dec1d7c7d1f517757263b575da4b12d81003870b09 size: 1366
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Status

</> Changes

Construir ahora

Configurar

Eliminar Pipeline

Full Stage View

Rename

Pipeline Syntax

Pipeline spring-boot

añadir descripción

Desactivar el Proyecto

Last Successful Artifacts

spring-boot-sample-actuator-2.0.2.jar

24,14 MB

view

Recent Changes

Historia de tareas

tendencia

Q. Filter builds...

#21

22 sep. 2022 15:54

#20

22 sep. 2022 15:42

#19

22 sep. 2022 15:39

#18

22 sep. 2022 15:37

#17

22 sep. 2022 15:35

#16

22 sep. 2022 15:33

#15

22 sep. 2022 15:29

#14

22 sep. 2022 15:27

#13

22 sep. 2022 15:26

#12

22 sep. 2022 15:22

#11

22 sep. 2022 15:07

#10

22 sep. 2022 15:07

Stage View

Average stage times:

(Average full run time: ~46s)

	Declarative: Tool Install	Build	Build Image
#19 Sep 22 15:54 No Changes	63ms	4s	41s
#18 Sep 22 15:42 No Changes	53ms	4s	6min 18s failed
#15 Sep 22 15:39 No Changes	50ms	5s	4s failed
#13 Sep 22 15:37 No Changes	51ms	470ms failed	26ms failed
#12 Sep 22 15:37 No Changes	51ms	470ms failed	26ms failed

- Podemos ver que se pusheo la imagen en docker hub y aparece con una actualización de hace 3 minutos.

des

Navegador web Firefox

22 de sep 15:58

ING SW3

Merging v

Mi unidad

Resumen

ing-sof

juanjrojas0

Edit perso

¿Cómo con

Docker Hub

Simple gro

Edit perso

spring-bo

spring-bo

Docker

← → ↻

https://hub.docker.com/repository/docker/juanrojas09/test-node

☆

🔒

docker hub

Search for great content (e.g., mysql)

Explore

Repositories

Organizations

Help

Upgrade

juanjrojas09

juanjrojas09

Repositories

test-node

Using 0 of 1 private repositories. [Get more](#)

General

Tags

Builds

Collaborators

Webhooks

Settings

🔔

Add a short description for this repository

The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.

Update

🌐

juanjrojas09 / test-node

Description

This repository does not have a description [✎](#)

🕒 Last pushed: 3 minutes ago

Docker commands

To push a new tag to this repository,

`docker push juanjrojas09/test-node:tagname`

Public View

Tags and scans

This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
latest		---	3 minutes ago

[See all](#)[Go to Advanced Image Management](#)

VULNERABILITY SCANNING - DISABLED

[Enable](#)

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions.

UpgradeLearn more

README

Repository description is empty. Click [here](#) to edit.