

Integración continua con maven, tomcat y jenkins

Antonio González Luque

Para empezar debemos tener el siguiente fichero de docker:

```
FROM jenkins/jenkins:lts
USER root
RUN apt-get update && apt-get install -y Maven
```

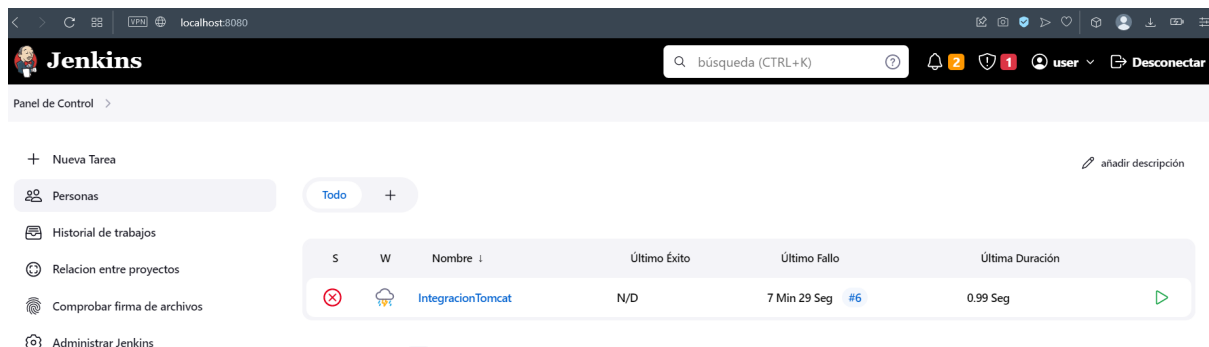
1º- Ejecutar el siguiente comando para descargar la imagen del contenedor

Docker build -t jenkins-with-maven.

2º- Ejecutamos el contenedor de jenkins:

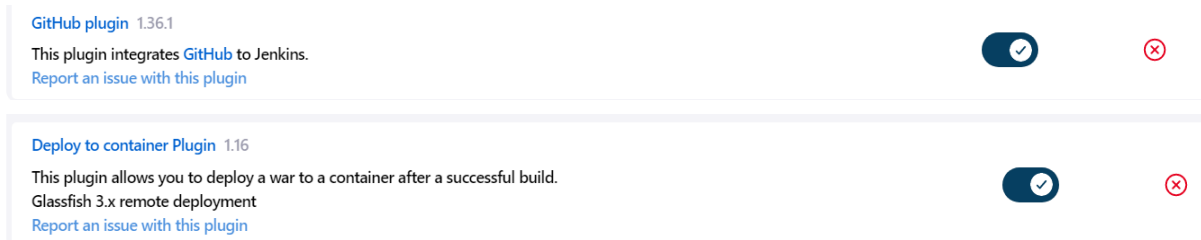
**docker run -d -p 8080:8080 -p 50000:50000 --name myjenkins -v
C:\DEV-Practices\docker\volumens\jenkins_home:/var/jenkins_home -v
C:\DEV-Practices\docker\volumens\jenkins_home\downloads:/var/jenkins_home/down
loads jenkins-with-maven**

3º- Entramos en localhost:8080 para ver que Jenkins se ha ejecutado correctamente. Una vez dentro entraremos con el Usuario: “user” y la contraseña “bitnami”.



S	W	Nombre	Último Éxito	Último Fallo	Última Duración
✖	📁	IntegracionTomcat	N/D	7 Min 29 Seg #6	0.99 Seg

4º- Lo siguiente será instalar los plugins necesarios:



5º- Ahora ejecutaremos Tomcat:

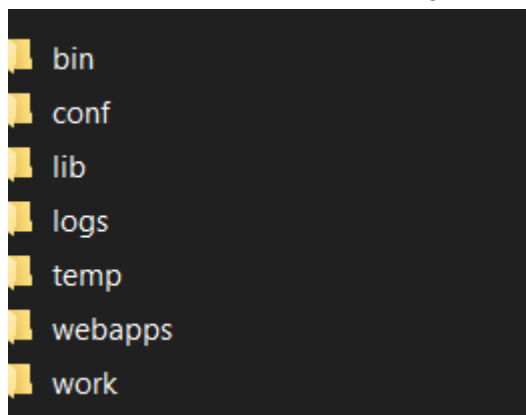
```
docker volume create --driver local --opt type=None --opt device=C:\DEV-Practices\docker\volumens\tomcatconfiguration --opt o=bind tomcatconfiguration
```

```
docker volume create --driver local --opt type=None --opt device=C:\DEV-Practices\docker\volumens\tomcat_webapps --opt o=bind tomcatwebapps
```

6º- Y ahora ejecutamos su contenedor:

```
docker run -d --name mytomcat -p 8888:8080 -v tomcatwebapps:/usr/local/tomcat/webapps -v tomcatconfiguration:/usr/local/tomcat/conf tomcat:9.0
```

7º- Ahora debemos tener descargado Tomcat y nos creará una carpeta como esta:



Pues entonces, en la carpeta de tomcat que creamos mediante los comandos debemos copiar y pegar los archivos para que quede así:

Este equipo > OS (C:) > DEV-Practices > docker > volumens > tomcat_webapps >			
	Nombre	Fecha de modificación	Tipo
	docs	10/02/2023 8:47	Carpeta de archivos
	examples	10/02/2023 8:47	Carpeta de archivos
	host-manager	10/02/2023 8:47	Carpeta de archivos
	manager	10/02/2023 8:47	Carpeta de archivos
	ROOT	10/02/2023 8:47	Carpeta de archivos

8º- Ahora debemos entrar en el fichero “tomcat-users.xml” y añadimos los siguiente:

```
<role rolename="manager-script"/>
<role rolename="manager-gui"/>
<user username="luke" password="luke"
roles="manager-gui,manager-script"/>
```

9º- Después abrimos “context.xml” dentro de host-manager/META-INF y comentamos la etiqueta de Valve:

```
<!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
      allow="127\.\d+\.\d+\.\d+|:::1|0:0:0:0:0:0:0:1" /> -->
```

10º- Entramos en Jenkins y creamos un nuevo proyecto. En los ajustes del código fuente seleccionamos la opción de Git

Configurar el origen del código fuente

☐ Ninguno

☒ Git ?

Si no aparece descarga el plugin.

Después seleccionas la url del repositorio donde esté alojado el proyecto y su rama principal.

11º- Añadimos las credenciales necesarias:

 **Git** ?

Repositories ?

Repository URL ?

Credentials ?

+ Add

 Avanzado...

El user y la password deberán ser las usadas en el xml de los usuarios.

12º- Añadimos la url de tomcat:

Acciones para ejecutar después.

☰ **Deploy war/ear to a container** ?

WAR/EAR files ?

Context path ?

Containers

Add Container ▾

☐ Deploy on failure

13º- Añadimos los Build steps:

Build Steps

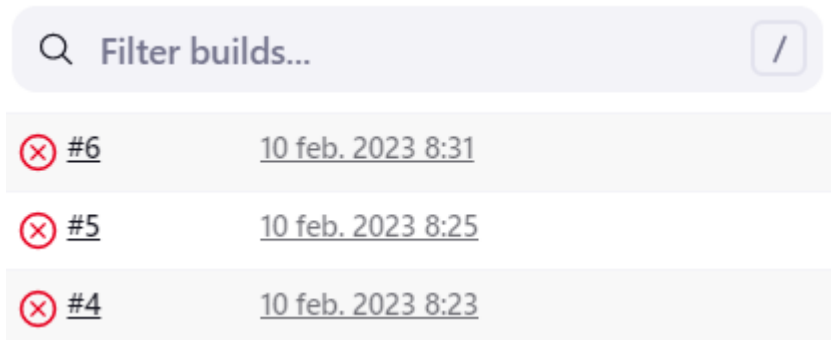
☰ **Ejecutar tareas 'maven' de nivel superior** ?

Goals

 ▾

Y listo... Deberíamos tener nuestro repositorio disponible en la url que hemos asignado para tomcat y cada cambio será testeado por Jenkins y desplegado o no en función de si hay algún fallo en el código.

En caso de crear una build fallida, jenkins te avisa de esta forma:



En caso de hacerlo bien se ejecutará y creará una nueva versión del proyecto. Y el símbolo estará en verde.

S	W	Nombre ↓	Último Éxito	Último Fallo	Última Duración	
✓	☀	jenkinsTomcat	1 Min 32 Seg #4	N/D	22 Ms	▶

En cualquier caso, da un error:

```
FATAL: command execution failed
java.io.IOException: error=2, No such file or directory
    at java.base/java.lang.ProcessImpl.forkAndExec(Native Method)
    at java.base/java.lang.ProcessImpl.<init>(ProcessImpl.java:340)
    at java.base/java.lang.ProcessImpl.start(ProcessImpl.java:271)
    at java.base/java.lang.ProcessBuilder.start(ProcessBuilder.java:1107)
Caused: java.io.IOException: Cannot run program "mvn" (in directory "/bitnami/jenkins/home/workspace/bb"): error=2, No such file or directory
    at java.base/java.lang.ProcessBuilder.start(ProcessBuilder.java:1128)
    at java.base/java.lang.ProcessBuilder.start(ProcessBuilder.java:1071)
    at hudson.Proc$LocalProc.<init>(Proc.java:254)
    at hudson.Proc$LocalProc.<init>(Proc.java:223)
    at hudson.Launcher$LocalLauncher.launch(Launcher.java:997)
    at hudson.Launcher$ProcStarter.start(Launcher.java:509)
    at hudson.Launcher$ProcStarter.join(Launcher.java:520)
    at hudson.tasks.Maven.perform(Maven.java:369)
    at hudson.tasks.BuildStepMonitor$1.perform(BuildStepMonitor.java:20)
    at hudson.model.AbstractBuild$AbstractBuildExecution.perform(AbstractBuild.java:818)
    at hudson.model.Build$BuildExecution.build(Build.java:199)
    at hudson.model.Build$BuildExecution.doRun(Build.java:164)
    at hudson.model.AbstractBuild$AbstractBuildExecution.run(AbstractBuild.java:526)
    at hudson.model.Run.execute(Run.java:1900)
    at hudson.model.FreeStyleBuild.run(FreeStyleBuild.java:44)
    at hudson.model.ResourceController.execute(ResourceController.java:107)
    at hudson.model.Executor.run(Executor.java:449)
Build step 'Invoke top-level Maven targets' marked build as failure
[DeployPublisher][INFO] Build failed, project not deployed
Finished: FAILURE
```

Bibliografía

[Tutorial de integración](#)