



## Memoria prácticas CRUD



# Jenkins

José Antonio Castillejo Lobato



# Índice

1. Qué es y para qué sirve Jenkins.....	3.
2. Qué es la integración continua.....	3.
3. Cómo instalar Jenkins con Docker.....	4/8.
4. Cómo instalar Jenkins con Docker-Compose.....	8/10.
5. Bibliografía.....	10.

## ¿Qué es y para qué sirve Jenkins?

Es un servidor open source para la integración continua. Se usa para probar y compilar proyectos de software de forma continua. Con Jenkins se facilita la incorporación de

cambios en un desarrollo y la entrega de nuevas versiones. Está escrito en Java, es multiplataforma y se puede acceder a través de una interfaz web.



Con este servidor se aceleran los procesos de progreso y entrega de los softwares. Además, cuenta con muchos plugins que se pueden utilizar durante todo el trabajo (construcción, despliegue y automatizado). Jenkins se encarga de supervisar las tareas repetitivas que surgen a lo largo de todo el proceso.

## ¿Qué es la integración continua?

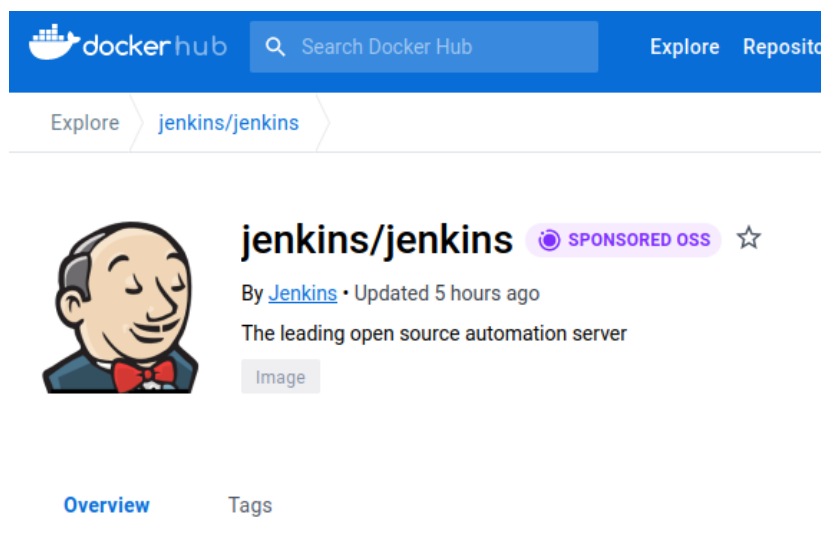
La integración continua es una práctica de desarrollo de software, en la cual los programadores suben su código a un repositorio central donde automáticamente pasan las pruebas métricas y de calidad. Esta técnica se suele realizar regularmente para detectar fallos y así mantener el código siempre actualizado.

Es decir, la integración continua es una manera de automatizar tareas cuando se sube el código, aunque también se puede utilizar para desplegar las aplicaciones. En un repositorio compartido viene bien utilizar una herramienta de integración continua para mantener una integridad en el código.

Por ejemplo desde la rama de master podemos escribir una serie de pruebas al hacer un pull o un merge request. Si las pruebas se pasan hacemos que se suba y se despliegue automáticamente en producción.

# Cómo instalar Jenkins con Docker

En este caso nos interesa instalar jenkins en docker pero también podríamos instalar su versión general. Tenemos que irnos a Docker Hub y buscar jenkins o directamente pulsar en el siguiente enlace <https://hub.docker.com/r/jenkins/jenkins>.



Abrimos la terminal y descargamos el contenedor con `docker pull jenkins/jenkins:alpine` tal y como nos indica Docker Hub, este paso se puede omitir ya que el próximo comando nos lo va a instalar de forma automática si no lo tenemos.

```
estudiante@DAW1:~$ docker pull jenkins/jenkins:alpine
alpine: Pulling from jenkins/jenkins
8921db27df28: Pull complete
76ebcf2895c4: Pull complete
f73a30177abe: Pull complete
83629d268ffb: Pull complete
cc8eb98c2e25: Pull complete
fd48e4e9ee51: Pull complete
8a8cc418517e: Pull complete
c6ceffed4402: Pull complete
30bc085cb604: Pull complete
e36e21c37e51: Pull complete
1b5376daec26: Pull complete
c8cceba9f60: Pull complete
Digest: sha256:32765336fb7e9c564a1dc7fac5d63baa60a1a0a1d685f491a13e1698739e0aa
Status: Downloaded newer image for jenkins/jenkins:alpine
docker.io/jenkins/jenkins:alpine
estudiante@DAW1:~$
```

Ahora lo arrancamos asignando el puerto con `-p` y el nombre del contenedor.



```
estudiante@DAW1:~$ docker run -it -p 8080:8080 jenkins/jenkins:alpine
Running from: /usr/share/jenkins/jenkins.war
webroot: /var/jenkins_home/war
2022-02-02 09:05:51.365+0000 [id: 1] INFO org.jenkins.war.Launcher#launch: Re
```

De esta forma jenkins se ejecuta y nos da la ruta y la contraseña para su inicio.

```
Jenkins initial setup is required. An admin user has been created and a password
generated.
Please use the following password to proceed to installation:

b0920db1a1344bf799c95f2e4bbc49b1

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
```

Para llegar a este punto es tan sencillo como poner el puerto anteriormente asignado y la ruta que nos especifica en la consola, en este caso sería **localhost:8090/var/jenkins\_home/secrets/initialAdminPassword** y ya solo tenemos que introducir la contraseña indicada en el paso anterior.

#### Getting Started

## Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (**not sure where to find it?**) and this file on the server:

```
/var/jenkins_home/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

Si hemos seguido los anteriores pasos correctamente la siguiente página que nos muestra es la siguiente, nos están preguntando si queremos los plugins por defecto o si los queremos seleccionar nosotros mismos.

# Customize 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.

En el caso de querer seleccionar a mano los plugins se nos mostrará la siguiente página donde tenemos que marcar los plugins que nos interese.

Organization and Administration

Build Features

Build Tools

Build Analysis and Reporting

Pipelines and Continuous Delivery

Source Code Management

Distributed Builds

User Management and Security

Notifications and Publishing

Languages

All | None | Suggested

Selected (19/52)

Note that the full list of plugins is not shown here. Additional plugins can be installed in the **Plugin Manager** once the initial setup is complete. [See the documentation for more information.](#)

### Organization and Administration (2/4)

☐ **Dashboard View**

Customizable dashboard that can present various views of job information.

29

☒ **Folders**

This plugin allows users to create "folders" to organize jobs. Users can define custom taxonomies (like by project type, organization type etc). Folders are nestable and you can define views within folders. Maintained by CloudBees, Inc.

5

☐ **Configuration as Code**

This plugin allows configuration of Jenkins based on human-readable declarative configuration files.

8

☒ **OWASP Markup Formatter**

Uses the [OWASP Java HTML Sanitizer](#) to allow safe-seeming HTML markup to be entered in project descriptions and the like.

7

### Build Features (4/10)

En nuestro caso vamos a dejar los plugins que vienen por defecto y se nos empezará a instalar la aplicación.



## Getting Started

# Getting Started

Folders	OWASP Markup Formatter	Build Timeout	Credentials Binding	<b>**</b> JavaBeans Activation Framework (JAF) API <b>**</b> JavaMail API
Timestampers	Workspace Cleanup	Ant	Gradle	
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View	
Git	SSH Build Agents	<input type="radio"/> Matrix Authorization Strategy	PAM Authentication	
LDAP	Email Extension	Mailer		
				<b>**</b> - required dependency

Una vez acabada la descarga nos pedirá la creación de un usuario, si no queremos crearnos un usuario lo dejaremos en blanco y nos asignará el nombre de admin

## Getting Started

# Create First Admin User

Username

Password

Confirm password

Full name

Nos preguntará la ruta para acceder al programa



## Getting Started

# Instance Configuration

Jenkins URL:

`http://localhost:8080/`

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD\_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Y ya tenemos Jenkins instalado y configurado.

# Cómo instalar Jenkins con Docker-Compose

Lo primero que tenemos que hacer es crear un archivo `.yaml` o `.yml`, donde indicamos la versión, el nombre del contenedor y la imagen, el puerto y la ruta del volumen. En este caso como ya tenemos el puerto 8080 ocupado con el anterior Jenkins vamos a poner 8081.



```
file Edit Selection View Go Run Terminal Help

EXPLORER
└─ UNTITLED (WORKSPACE)
   └─ BootstrapPeluqueria
      ├── css
      ├── img
      ├── scripts
      ├── booking.html
      ├── index.html
      └── service.html
   └─ Jenkins
      └─ docker-compose.yml

Jenkins > docker-compose.yml
1  version: "3"
2  services:
3    jenkins:
4      container_name: jenkins_managed_by_compose
5      image: jenkins/jenkins:alpine
6      ports:
7        - "8081:8080"
8        - "50001:50000"
9      volumes:
10       - jenkins_home:/var/jenkins_home
11      restart: always
12      networks:
13        - net_jenkins
14  volumes:
15    jenkins_home:
16  networks:
17    net_jenkins:
```

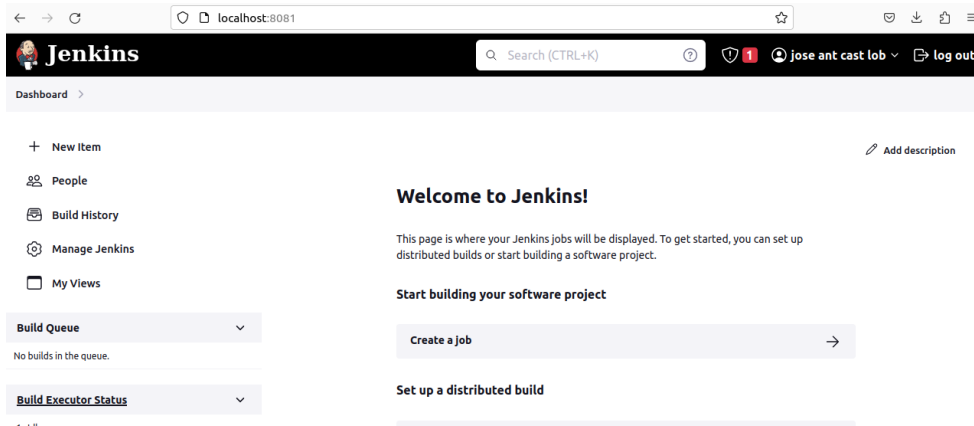
Una vez creado el archivo lo arrancamos con `docker-compose up -d`.

```
estudiante@DAW1:~/Documents/Despliegue/Jenkins$ docker-compose up -d
Creating network "jenkins_net_jenkins" with the default driver
Creating volume "jenkins_jenkins_home" with default driver
Creating jenkins_managed_by_compose ... done
estudiante@DAW1:~/Documents/Despliegue/Jenkins$
```

De esta forma no nos indica la contraseña como en el paso anterior por lo que tenemos que hacer un paso más complicado para obtenerla, tenemos que poner **docker exec** y ahora el nombre del archivo, este puede cambiar por lo que la forma más sencilla es tabular y que lo autocomplete solo, luego tenemos que poner un `cat` y la ruta donde se encuentra la contraseña. Como se ve en la imagen nos muestra la contraseña.

```
estudiante@DAW1:~/Documents/Despliegue/Jenkins$ docker exec jenkins_managed_by_compose cat /var/jenkins_home/secrets/initialAdminPassword
30f8ec607cbe41cdbc19c4aef33d603a
estudiante@DAW1:~/Documents/Despliegue/Jenkins$
```

Ya los pasos son como los anteriores y como podemos ver tenemos el Jenkins instalado en el puerto 8081.



## Bibliografía:

<https://www.bambu-mobile.com/para-que-sirve-jenkins/>

<https://codingpotions.com/jenkins-integracion-continua>

<https://aprendevops.com/instalar-jenkins-en-docker>

```
users.xml
estudiante@DAW1:~/Documents/Despliegue/Jenkins$ docker exec jenkins_managed_by_c
compose cat /var/jenkins_home/users/users.xml
<?xml version='1.1' encoding='UTF-8'?>
<hudson.model.UserIdMapper>
  <version>1</version>
  <idToDirectoryNameMap class="concurrent-hash-map">
    <entry>
      <string>castillejo</string>
      <string>castillejo_15739825004896619583</string>
    </entry>
  </idToDirectoryNameMap>
</hudson.model.UserIdMapper>estudiante@DAW1:~/Documents/Despliegue/Jenkins$
```