

Integración continua con Jenkins



DESPLIEGUE DE APLICACIONES WEB

SANDRA RUIZ JIMÉNEZ
2º DAW

Lista de contenidos

1. INTEGRACIÓN CONTINUA	1
2. ¿QUÉ ES JENKINS?	2
3. ¿CÓMO FUNCIONA JENKINS?	3
4. INSTALACIÓN DE JENKINS CON DOCKER-COMPOSE	4
4.1. CREACIÓN DEL FICHERO DOCKER-COMPOSE	4
4.2. LANZAR EL DOCKER-COMPOSE	4
4.3. ACCEDER A JENKINS	5
5. CONFIGURACIÓN DE JENKINS	5
BIBLIOGRAFÍA	10

1. INTEGRACIÓN CONTINUA

La integración continua (*Continuous Integration*, CI) consiste en la automatización de las labores de compilación, *testing* y análisis estático del código.

Cada cambio que realiza un desarrollador (resolución de un bug, creación de una nueva funcionalidad, etc.) se comprueba compilando el código fuente y obteniendo un ejecutable (llamado *build*). Si es validado, será incorporado al código fuente y desplegado.

La CI es un requisito necesario e innegociable para hacer software de calidad, ya que realizar *builds* frecuentemente y comprobar su funcionamiento ayuda a conseguir un producto final más fiable, al prevenir fallos en producción.

Hay muchas maneras de hacer integración continua, pero independientemente del sistema que utilicemos para iniciarlo, el proceso como mínimo seguirá el siguiente camino:

1. Descargar el código fuente desde el repositorio de control de versiones.
2. Compilar el código según sea necesario.
3. Realizará las pruebas unitarias y/o de integración.
4. Publicará los resultados de modo que sean accesibles.



Las principales ventajas que tiene la CI son:

- Detectar rápidamente los posibles errores de compilación del código.
- Detectar funcionamientos anómalos en el software.
- Mejorar la calidad del producto.
- Compilar/testear el código en diferentes plataformas.

Una de las principales herramientas utilizadas para la integración continua es Jenkins.

2. ¿QUÉ ES JENKINS?

Jenkins es un servidor *open source* gratuito para la integración continua, es decir, es una herramienta que se utiliza para compilar y probar proyectos de software de forma continua. Escrito en Java, es multiplataforma y accesible mediante interfaz web.

Con Jenkins, se acelera el proceso de desarrollo y entrega de software a través de la automatización.

Tiene disponible muchos plugins, que permiten que se pueda implementar en diferentes etapas del ciclo de vida del desarrollo, como la compilación, la documentación, el testeo o el despliegue.

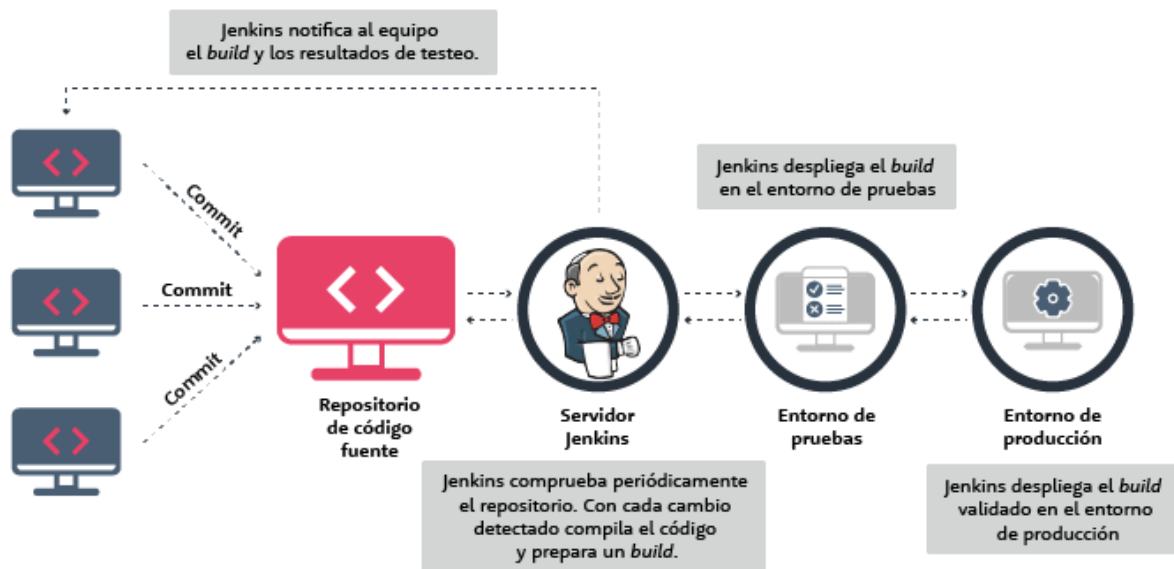
Jenkins permite:

- Automatizar la compilación y testeo de software.
- Notificar a los equipos correspondientes la detección de errores.
- Desplegar los cambios en el código que hayan sido validados.
- Hacer un seguimiento de la calidad del código y de la cobertura de las pruebas.
- Generar la documentación de un proyecto.

Esto hace que el ciclo de desarrollo sea más rápido, por lo que se entregan más funcionalidades y más frecuentemente a los usuarios, con lo que los beneficios son mayores, algo esencial en las metodologías AGILE.

3. ¿CÓMO FUNCIONA JENKINS?

1. El desarrollador hace un commit en el repositorio del código fuente.
2. El servidor de Jenkins hace comprobaciones periódicas para detectar cambios en el repositorio.
3. Jenkins detecta los cambios que se han producido en el código fuente. Compila el código y prepara un *build*.
4. Si el build falla, envía una notificación al equipo. Si resulta exitoso, lo despliega en el servidor de testeo.
5. Después de la prueba, Jenkins genera un *feedback* y notifica al equipo el build y los resultados del testeo.
6. Se repite el proceso en bucle.



4. INSTALACIÓN DE JENKINS CON DOCKER-COMPOSE

Podríamos instalar Jenkins directamente con el comando:

```
docker run -it -p 8086:8080 jenkins/jenkins:lts
```

Pero de esta forma, perderíamos los datos cada vez que se parase el contenedor.

Para que esto no sea así, vamos a instalarlo con docker-compose e indicarle un volumen para que persista los datos:

4.1. CREACIÓN DEL FICHERO DOCKER-COMPOSE

En nuestra carpeta Home, creamos un directorio llamado jenkins y dentro creamos el siguiente fichero docker-compose.yml

```
docker-compose.yml x
1  version: '3.8'
2
3  services:
4    jenkins:
5      image: jenkins/jenkins:lts
6      privileged: true
7      user: root
8      ports:
9        - 8092:8080
10       - 50000:50000
11      container_name: jenkins
12      volumes:
13        - /home/sandra/jenkins/jenkins_configuration:/var/jenkins_home
14        - /var/run/docker.sock:/var/run/docker.sock
```

4.2. LANZAR EL DOCKER-COMPOSE

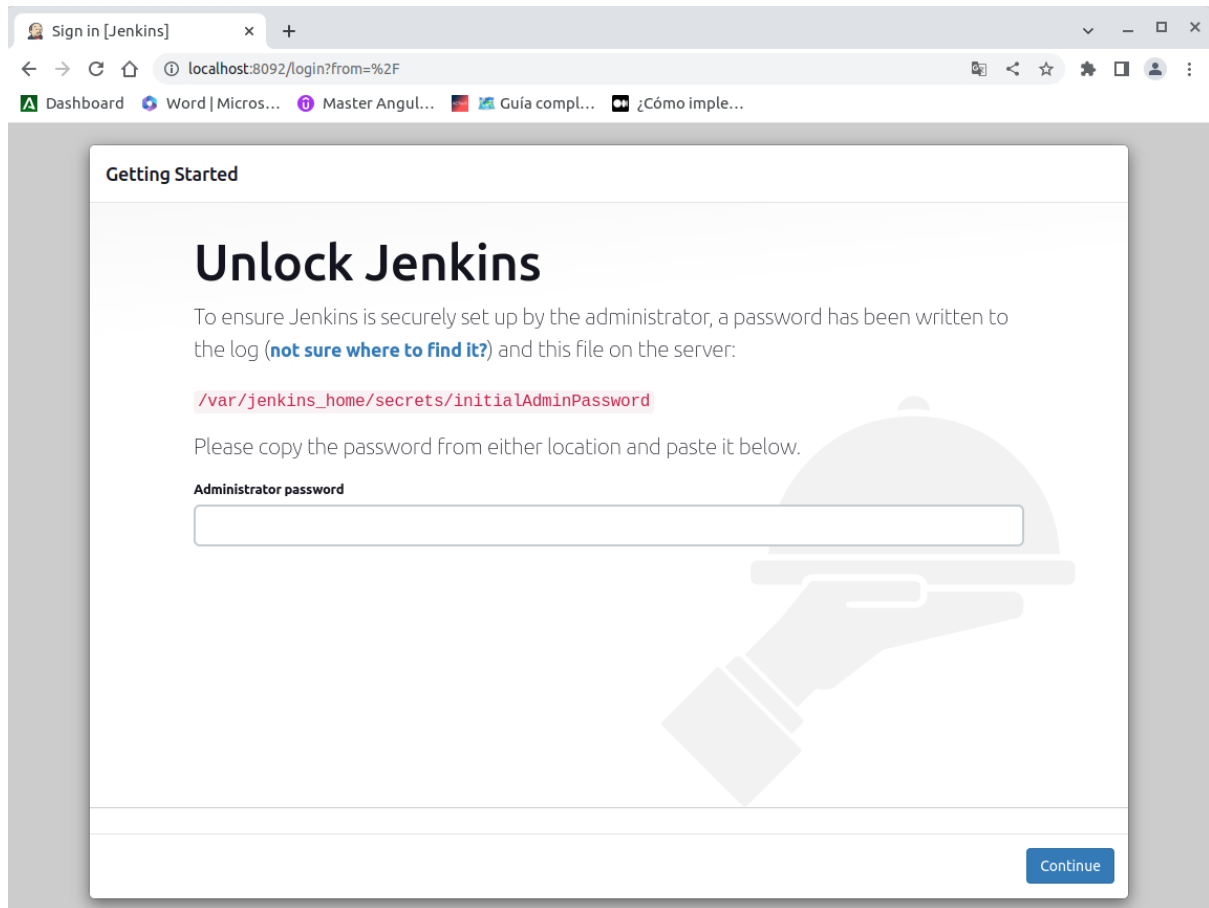
Lanzamos el fichero docker-compose con el comando:

```
docker-compose up -d
```

```
sandra@sandra-UX410UAK:~/jenkins$ docker-compose up -d
Recreating jenkins ... done
sandra@sandra-UX410UAK:~/jenkins$
```

4.3. ACCEDER A JENKINS

En el navegador, accedemos a Jenkins a través del puerto de redirección que le hayamos indicado, en nuestro caso el 8092:



5. CONFIGURACIÓN DE JENKINS

Lo primero que debemos hacer es configurar el usuario de Jenkins. Cuando entramos al navegador por primera vez nos pedirá una clave. Dicha clave se genera en la instalación del contenedor y podemos recuperarla, como nos dice en el asistente, en `/var/jenkins_home/secrets/initialAdminPassword`

Para ello, accedemos al contenedor con el comando:

`docker exec -it jenkins /bin/bash`

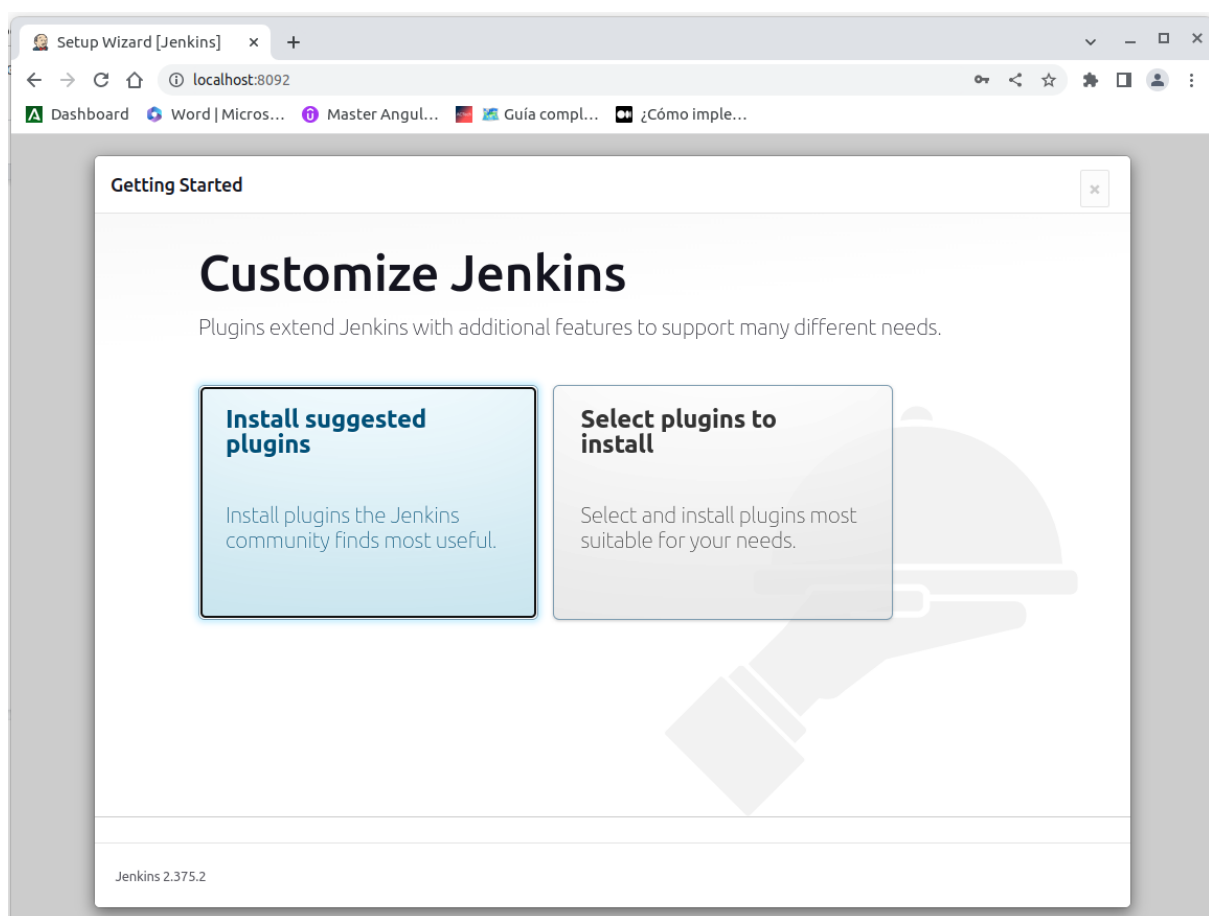
Y podemos ver la contraseña ejecutando:

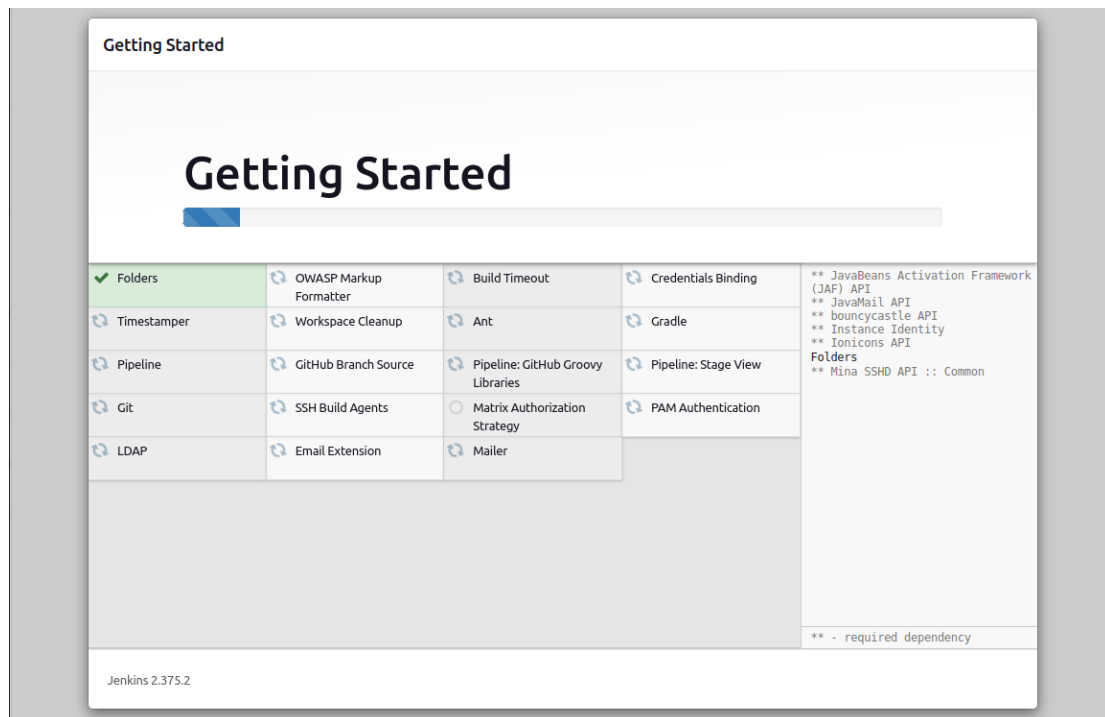
`tail /var/jenkins_home/secrets/initialAdminPassword`

```
sandra@sandra-UX410UAK: ~  
sandra@sandra-UX410UAK:~$ docker exec -it jenkins /bin/bash  
root@f81662d2186d:/# tail /var/jenkins_home/secrets/initialAdminPassword  
9b4aeac4120a432183849d2a282d2a04  
root@f81662d2186d:/#
```

Copiamos la contraseña y la introducimos en el navegador.

A continuación Jenkins nos pedirá que plugins queremos instalar. Escogemos la opción de plugins por defecto y comienza su proceso de instalación.





Al terminar la instalación de los plugins, nos pide que creamos los datos de la cuenta de administrador:

The screenshot shows the 'Create First Admin User' form in Jenkins 2.375.2. The form has five input fields: Username, Password, Confirm password, Full name, and E-mail address. The Username field contains 'user', the Password field contains '****', the Confirm password field contains '****', the Full name field contains 'Sandra', and the E-mail address field contains 'sandra.ruizjimenez86@gmail.com'. At the bottom right, there are two buttons: 'Skip and continue as admin' and 'Save and Continue'.

Username

user

Password

Confirm password

Full name

Sandra

E-mail address

sandra.ruizjimenez86@gmail.com

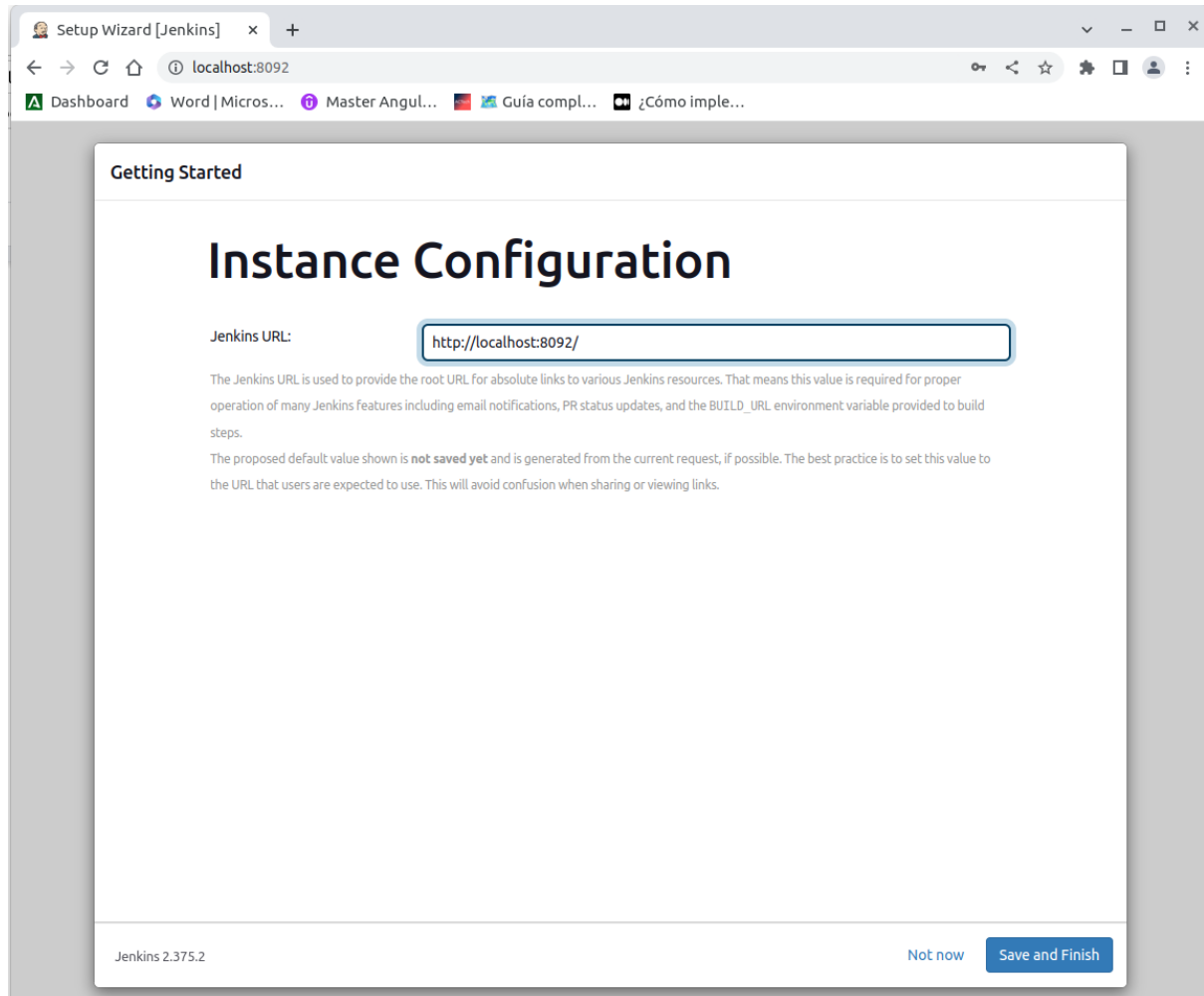
Jenkins 2.375.2

Skip and continue as admin

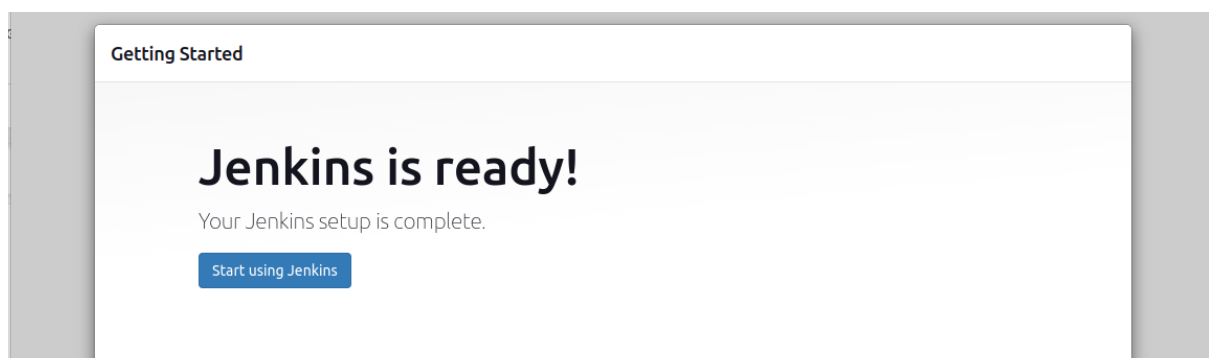
Save and Continue

Nuestro nombre de usuario y contraseña serán “user”.

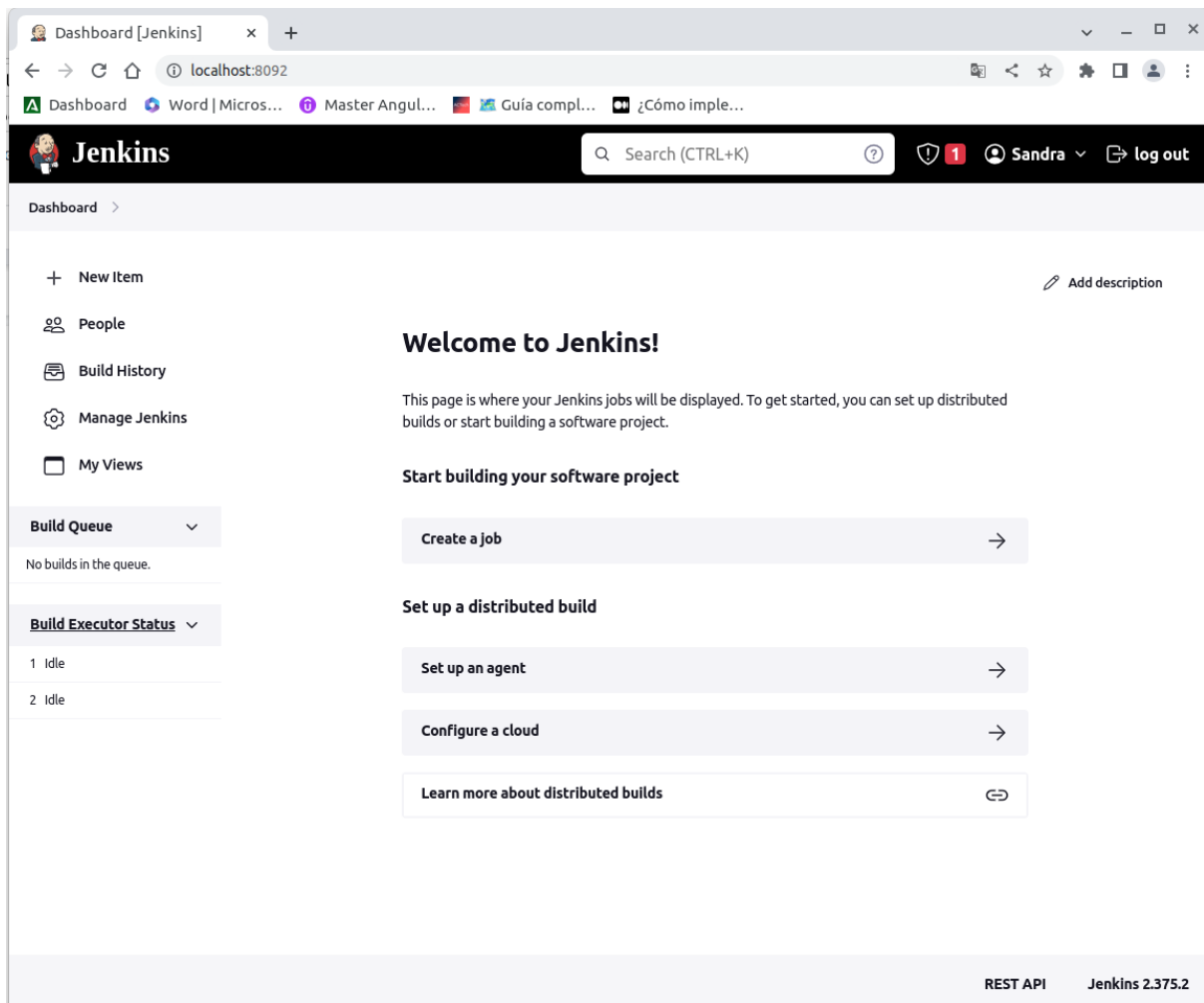
El último paso es configurar la ruta por defecto de Jenkins. Vamos a dejar la que aparece por defecto:



Guardamos y nos dice que Jenkins está listo.



Hacemos clic en el botón Start using Jenkins y ya podemos usar la herramienta.



The screenshot shows the Jenkins Dashboard in a web browser. The browser's address bar indicates the URL is `localhost:8092`. The Jenkins logo is in the top left, and a search bar with the text "Search (CTRL+K)" is in the top right. Below the search bar, there's a notification icon with a red "1" and a user profile for "Sandra" with a "log out" button. The main content area is titled "Welcome to Jenkins!" and includes a description: "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." Below this, there's a section "Start building your software project" with a "Create a job" button. Another section "Set up a distributed build" contains three buttons: "Set up an agent", "Configure a cloud", and "Learn more about distributed builds". On the left sidebar, there are links for "New Item", "People", "Build History", "Manage Jenkins", and "My Views". Below these, there's a "Build Queue" section showing "No builds in the queue." and a "Build Executor Status" section showing two idle executors. At the bottom right, there are links for "REST API" and "Jenkins 2.375.2".

Dashboard [Jenkins] x +

localhost:8092

Dashboard Word | Micros... Master Angul... Guía compl... ¿Cómo imple...

Jenkins Search (CTRL+K) ? 1 Sandra log out

Dashboard >

+ New Item Add description

People

Build History

Manage Jenkins

My Views

Build Queue ▾

No builds in the queue.

Build Executor Status ▾

1 Idle

2 Idle

Welcome to 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

Set up an agent →

Configure a cloud →

Learn more about distributed builds ↗

REST API Jenkins 2.375.2

BIBLIOGRAFÍA

ApplInventiv (2022). *How to Install Jenkins using Docker Compose – The Process*. Disponible en: <https://appinventiv.com/blog/jenkins-installation-using-docker-compose/>.

Consultado: 03/02/2023

Campus MVP (s.f.). *Integración continua: qué es y por qué deberías aprender a utilizarla cuanto antes*. Disponible en:

<https://www.campusmvp.es/recursos/post/integracion-continua-que-es-y-por-que-deberias-aprender-a-utilizarla-cuanto-antes.aspx>. Consultado: 03/02/2023

Sentrio (2021). *Introducción a Jenkins: ¿qué es, para qué sirve y cómo funciona?* Disponible en: <https://sentrio.io/blog/que-es-jenkins/>. Consultado: 03/02/2023