

Jenkins -Cl



Jenkins

Vicente Villarreal Hinojo
2ºDAW

Introducción:

En esta práctica vamos a explicar un poco que es Jenkins y para que sirve, así como su proceso de instalación en Windows, por que es tan utilizado y que es la Integración continua.

¿Qué es Jenkins?

Jenkins es un servidor open source para la integración continua. Es una herramienta que se utiliza para compilar y probar proyectos de software de forma continua, lo que facilita a los desarrolladores integrar cambios en un proyecto y entregar nuevas versiones a los usuarios. Escrito en Java, es multiplataforma y accesible mediante interfaz web. Es el software más utilizado en la actualidad para este propósito.

Con Jenkins, las organizaciones aceleran el proceso de desarrollo y entrega de software a través de la automatización. Mediante sus centenares de plugins, se puede implementar en diferentes etapas del ciclo de vida del desarrollo, como la compilación, la documentación, el testeo o el despliegue.

 ¿Qué es JENKINS y para qué sirve?

¿Qué es la integración continua (CI)?

La integración continua o Continuous Integration (CI) es una práctica habitual en desarrollo de software que consiste en integrar frecuentemente mejoras en el código de un proyecto una vez han sido validadas, normalmente varias veces al día, con el objetivo de detectar errores lo antes posible.

Cada cambio que realiza un desarrollador (ya sea una resolución de un bug, la 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.

Realizar builds frecuentemente y comprobar su funcionamiento ayuda a conseguir un producto final más fiable, al prevenir fallos en producción. Gracias a la integración continua los desarrolladores pueden detectar y corregir errores constantemente, evitar despliegues traumáticos y reducir notablemente el time to market.

Además, gracias a herramientas como Jenkins, se puede conocer el estado del software en cada momento, monitorizar la calidad del código, la cobertura de las pruebas y reducir la deuda técnica y los costes.

La integración continua es parte integral de DevOps y trabaja en la línea de entregar valor a los usuarios en el menor tiempo posible. Se vincula habitualmente con el Extreme Programming (XP) y las metodologías ágiles.

¿Qué se puede hacer con Jenkins?

Con Jenkins podemos automatizar multitud de tareas que nos ayudarán a reducir el time to market de nuestros productos digitales o de nuevas versiones de ellos. Concretamente, con esta herramienta podemos:

- 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.

Podemos ampliar las funcionalidades de Jenkins a través de múltiples plugins creados por la comunidad, diseñados para ayudarnos en centenares de tareas, a lo largo de las diferentes etapas del proceso de desarrollo.

¿Cómo funciona Jenkins?

Para entender cómo funciona Jenkins vamos a ver un ejemplo de cómo sería el flujo de integración continua utilizando esta herramienta:

- Un desarrollador hace un commit de código en el repositorio del código fuente.
- El servidor de Jenkins hace comprobaciones periódicas para detectar cambios en el repositorio.
- Poco después del commit, Jenkins detecta los cambios que se han producido en el código fuente. Compila el código y prepara un **build**. Si el build falla, envía una notificación al equipo en cuestión. Si resulta exitoso, lo despliega en el servidor de testeo.
- Después de la prueba, Jenkins genera un feedback y notifica al equipo el build y los resultados del testeo.
- Jenkins continúa revisando el repositorio frecuentemente y todo el proceso se repite.

¿Cómo Instalamos Jenkins en Windows?

1. Primero descargamos jenkins desde su página oficial seleccionando la versión de Windows.

<https://www.jenkins.io/download/>

📄 Download Jenkins 2.375.2 LTS for:

Generic Java package (.war)

SHA-256: e572525f7fa43b082e22896f72570297d88daec4f36ab4f25fdadca885f95492

Docker

Ubuntu/Debian

CentOS/Fedora/Red Hat

Windows

openSUSE

FreeBSD 📦

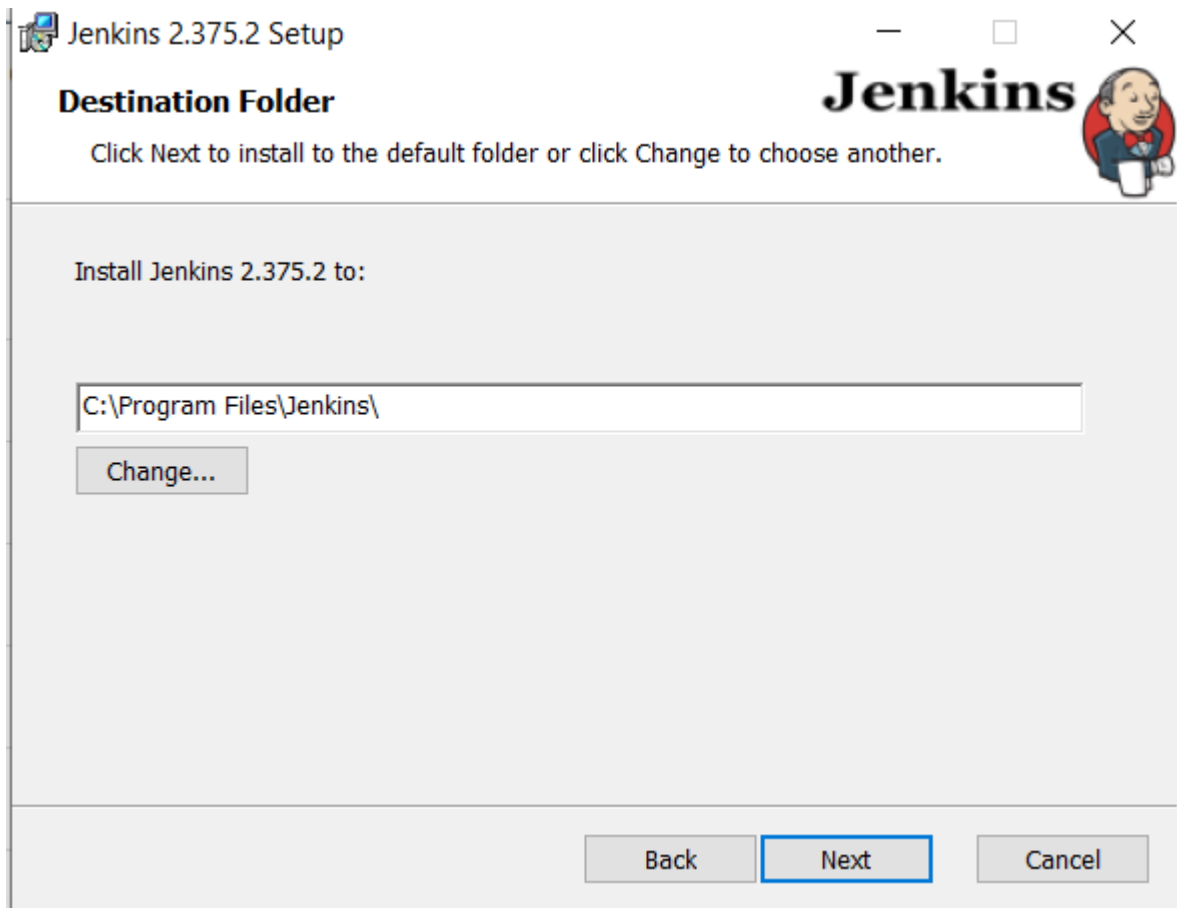
Gentoo 📦

macOS 📦

OpenBSD 📦

2. Después lo ejecutamos y seguimos el proceso de instalación.





Carpeta donde se instalará todo lo relacionado con Jenkins



Service Logon Credentials

Enter service credentials for the service.

Jenkins 2.375.2 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.375.2 to run successfully.

Logon Type:

- ☒ Run service as LocalSystem (not recommended)
☐ Run service as local or domain user:

Account:

Password:

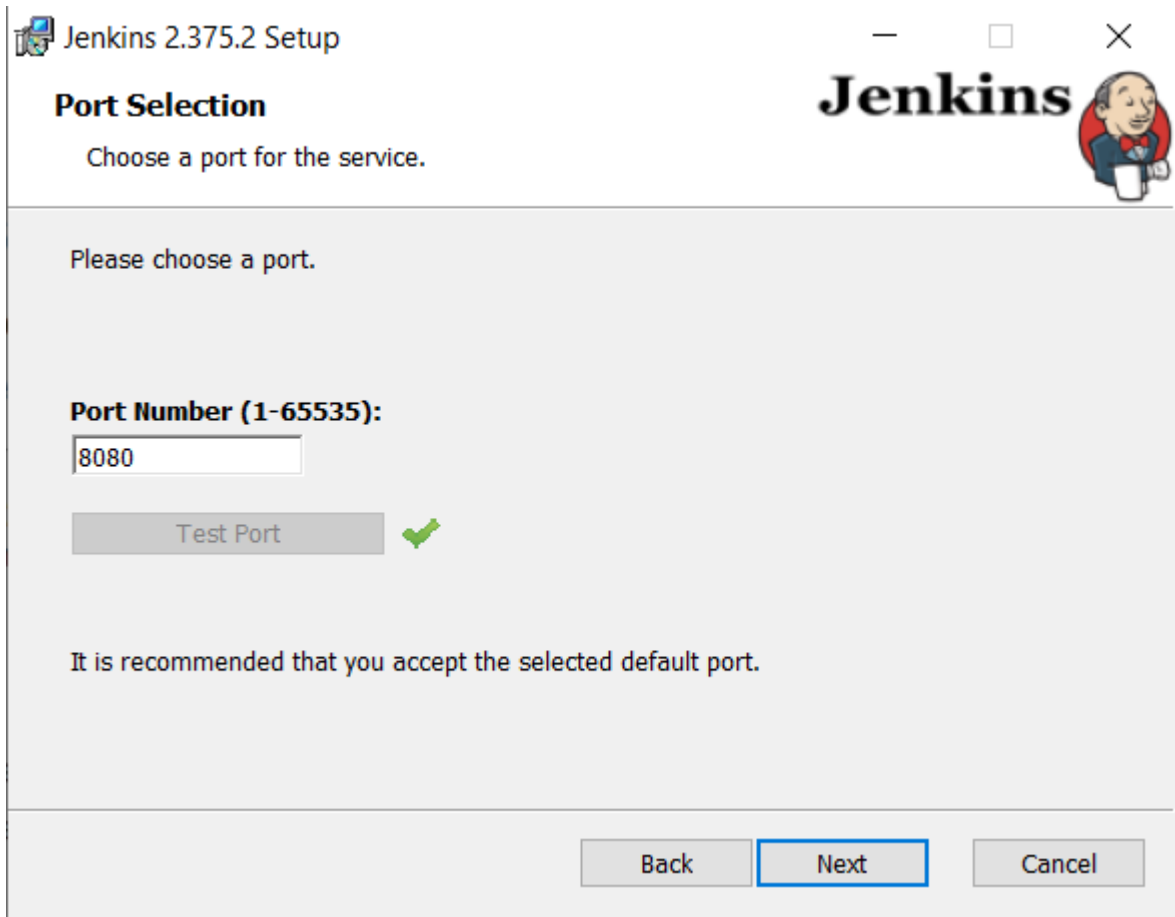
Test Credentials

Back

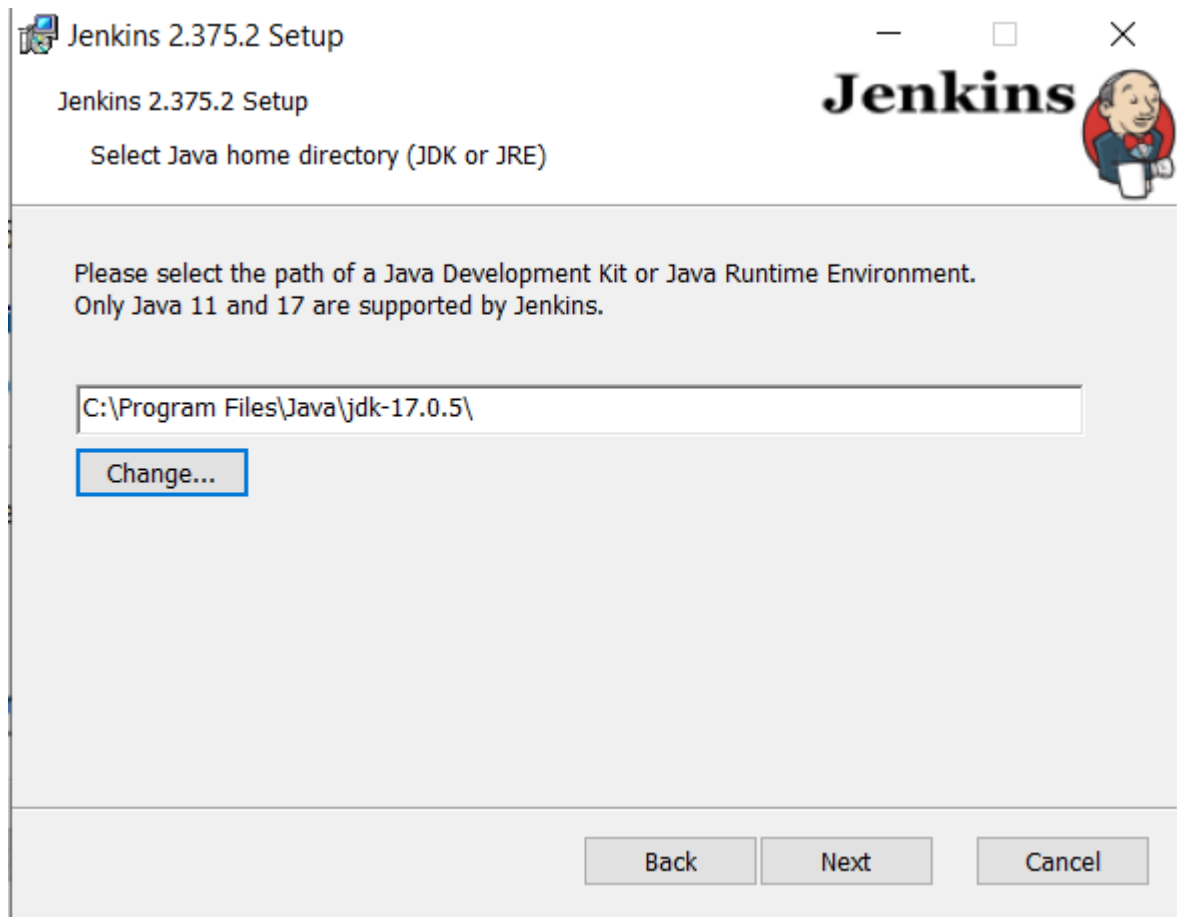
Next

Cancel

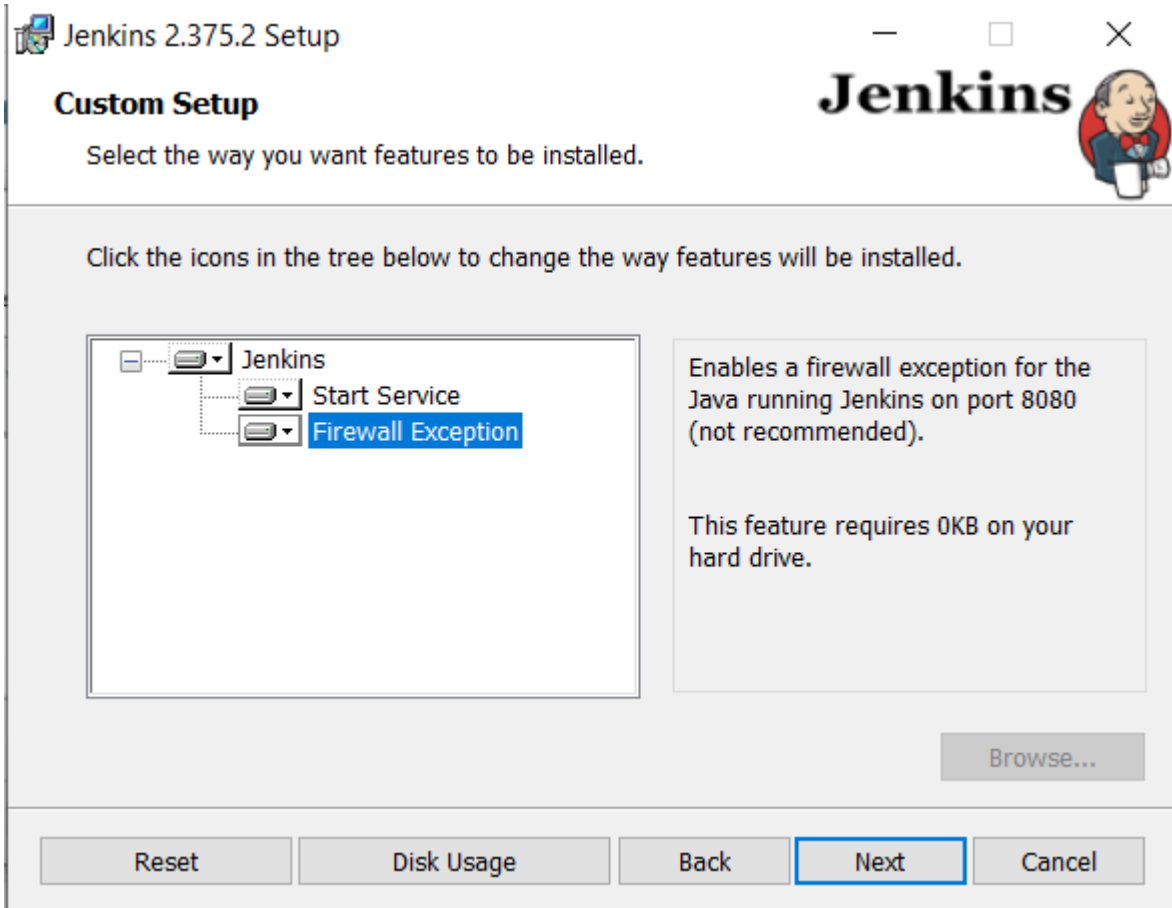
le damos a la primera opcion



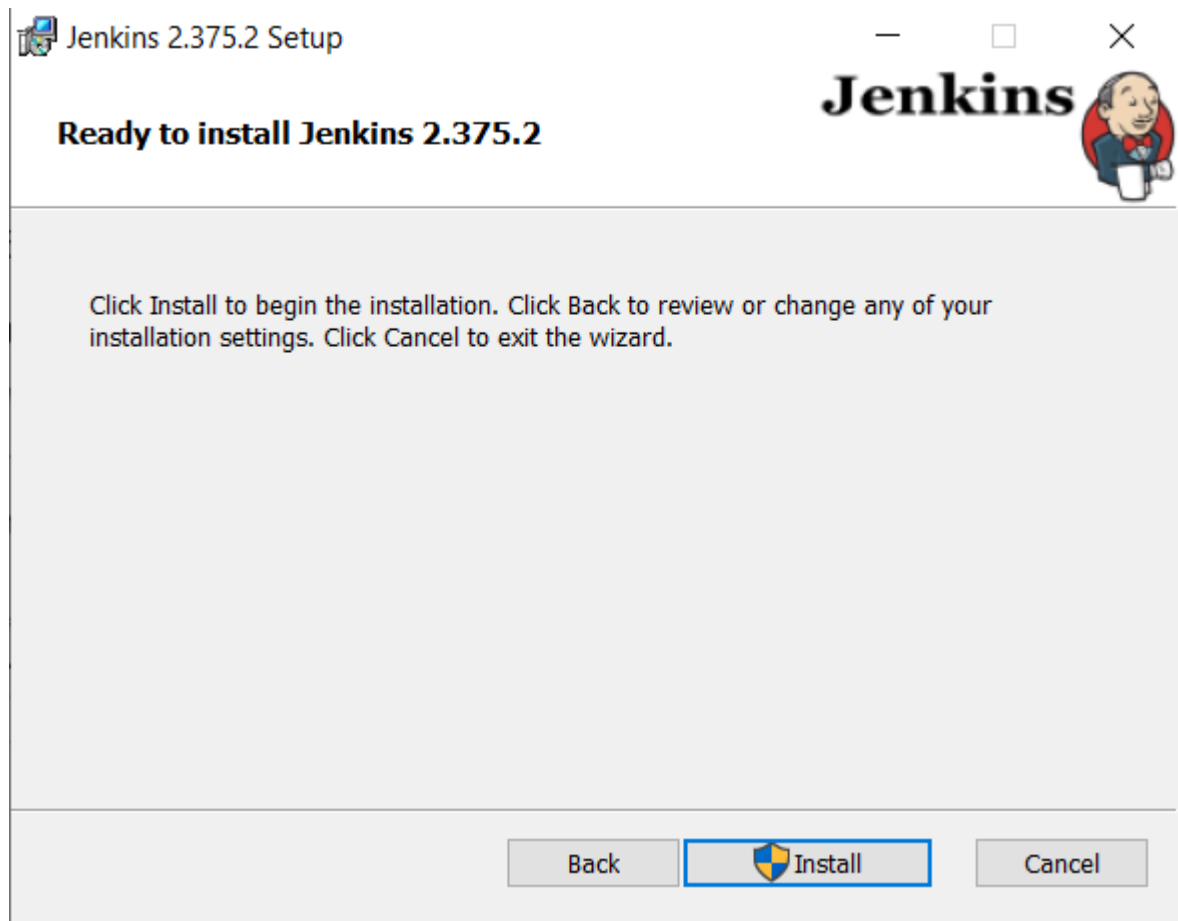
Test para ver si funciona el puerto.



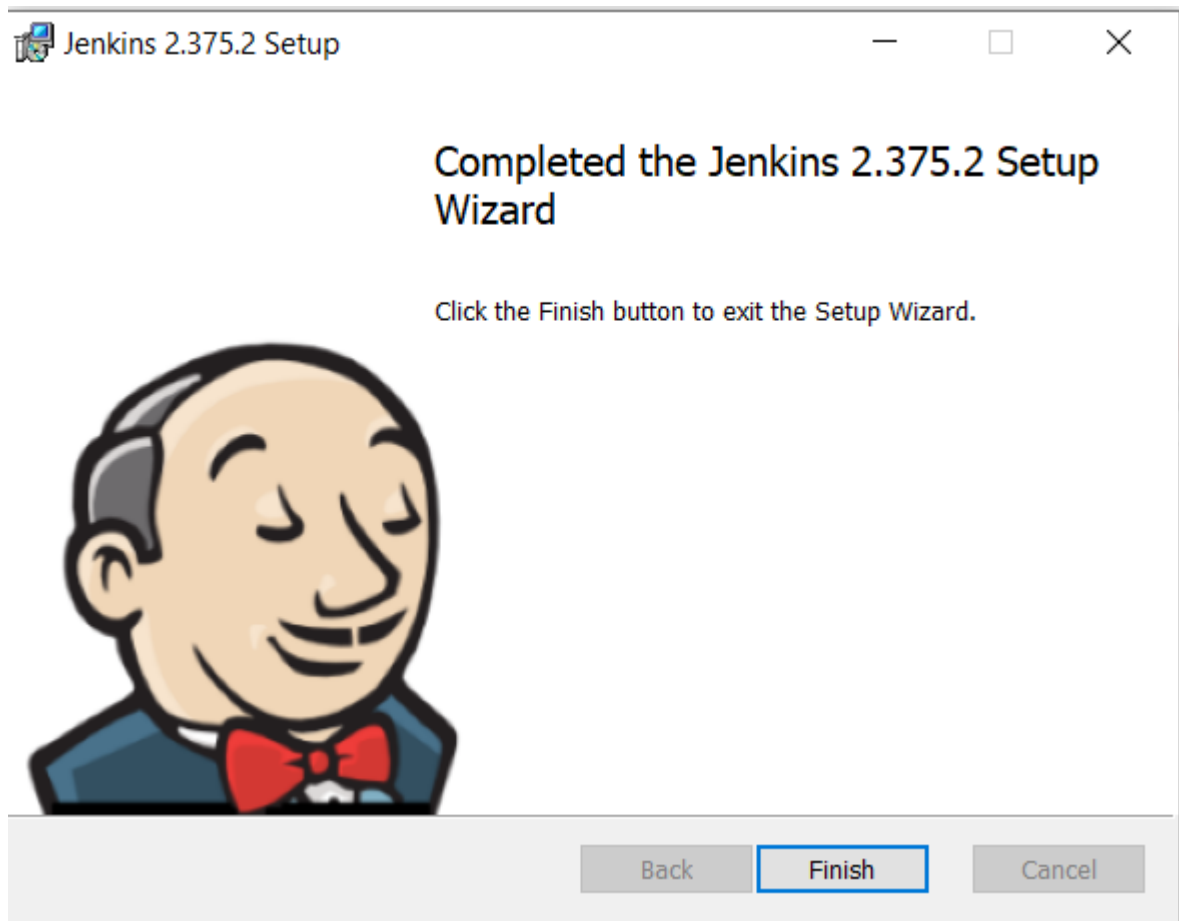
Le decimos dónde está nuestro jdk



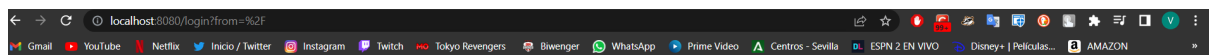
habilitamos el firewall.



le damos a instalar.



se ha completado, accedemos al puerto.



Por favor espera hasta que Jenkins esté listo. ...

Su navegador recargará automáticamente esta página cuando Jenkins esté listo.



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:

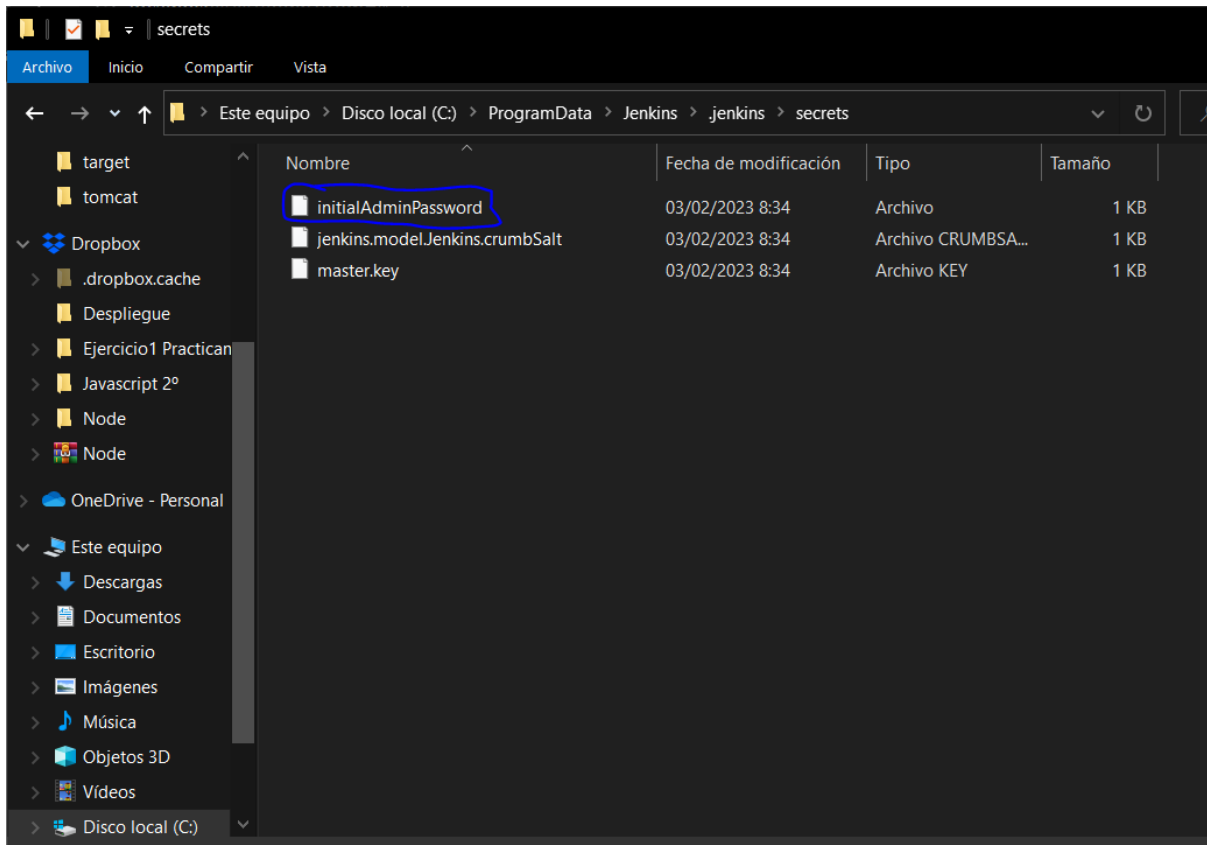
`C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword`

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

Administrator password

Continue

Para la password inicial vamos al directorio de la ruta.



La copiamos



initialAdminPassword: Bloc de notas

Archivo Edición Formato Ver Ayuda

61b9096f08dd4186a39c6f8062b92a1c

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:

`C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword`

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

Administrator password

.....

Continue

La pegamos

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 2.375.2

Getting Started



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

Jenkins 2.375.2

Instalamos los plugins recomendados.

Getting Started

Usuario

Contraseña

Confirma la contraseña

Nombre completo

Dirección de email

Jenkins 2.375.2

[Skip and continue as admin](#)

[Save and Continue](#)

Nos configuramos nuestro usuario.

Getting Started

Instance Configuration

Jenkins URL:

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.

Jenkins 2.375.2

[Not now](#)

[Save and Finish](#)

Getting Started

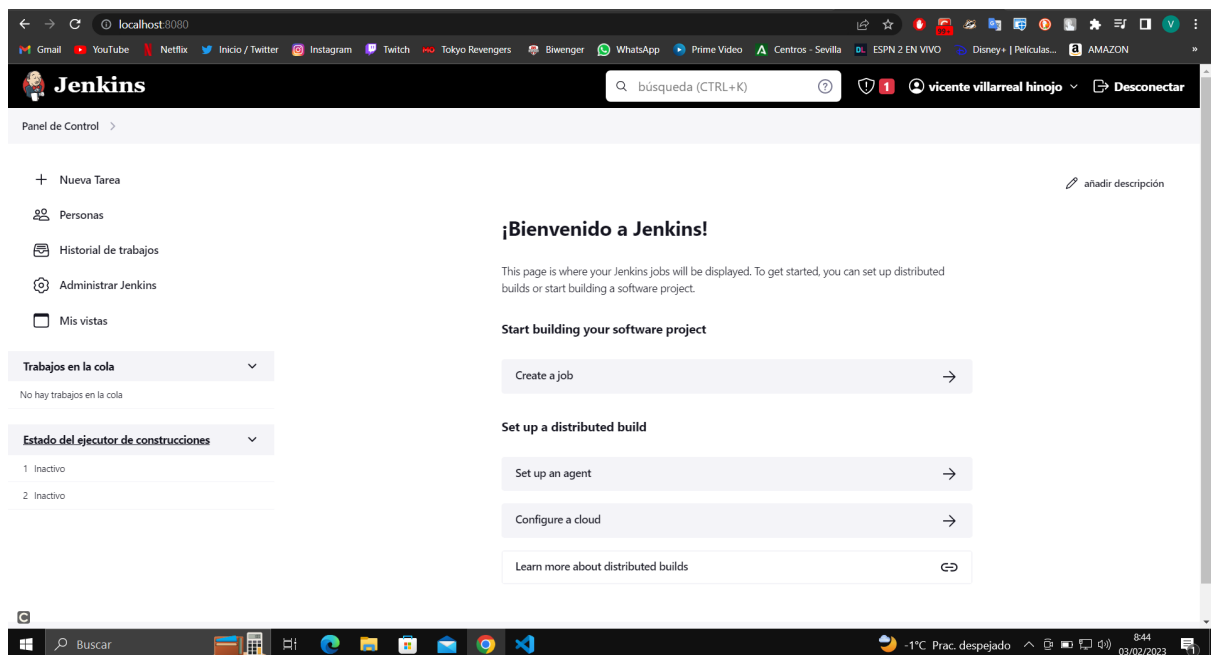
Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

Jenkins 2.375.2

Listo.



Tendríamos jenkins listo.

CON DOCKER COMPOSE:

Creamos un directorio y creamos un archivo docker-compose.yml abriendo ese directorio con Visual Studio, y a continuación le copiamos este texto.

```
version: '2'
```

```
services:
```

```
  jenkins:
```

```
    image: docker.io/bitnami/jenkins:2
```

```
    ports:
```

```
      - '80:8080'
```

```
    environment:
```

```
      - JENKINS_PASSWORD=bitnami
```

```
    volumes:
```

```
      - 'jenkins_data:/bitnami/jenkins'
```

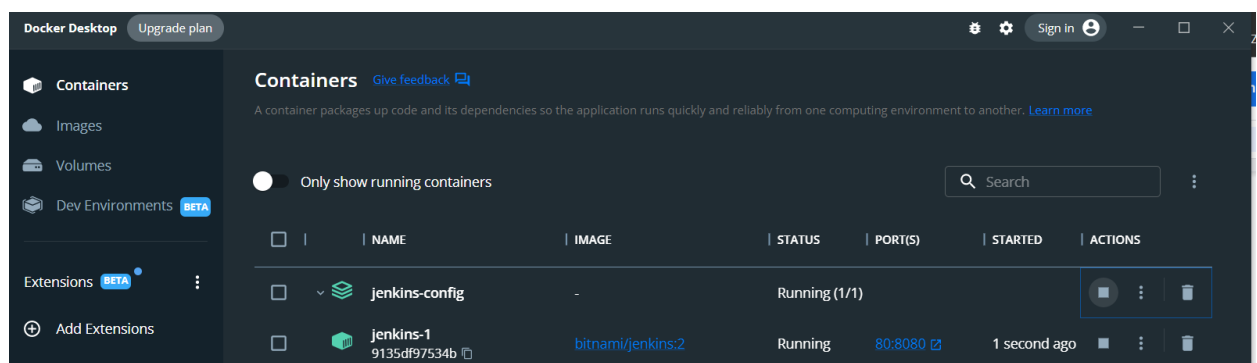
```
volumes:
```

```
  jenkins_data:
```

```
    driver: local
```

Cuando lo tengamos listo, abrimos la powershell y nos dirigimos hacia ese directorio donde tenemos el docker-compose.yml,

ejecutamos docker-compose up -d y cuando se crea.



buscamos el puerto 8080 y para acceder el usuario es: **user** y la contraseña: **bitnami**



Welcome to Jenkins!

☐ Keep me signed in

Sign in

Y listo, lo tendremos instalado con docker-compose.