# **Características**

* Es Open Source.
* Escrito en java sobre JVM.
* Mayormente se corre en Linux.
* Es extensible.
* Se puede escribir plugins en java.
* La comunidad aporta tanto escribiendo un sin fin de Plugins.
* Jenkins es amigable y flexible por la cantidad de comunidad que tiene.
* Compañías enormes tiene un solo Jenkins.
* Jenkins puede crecer horizontalmente añadiendo más hadware ya que soporta esto con el termino **slaves**.
* Jenkins es un servicio de automatización.
* CircleCI realiza lo mismo que jenkins.
* Actualmente se puede escribir tu job con código y no solo con la interfaz.

## **Escalabilidad Vertical**

Relacionado con el hardware del servidor, lo conseguimos aumentando los recursos del servidor (procesamiento, memoria, almacenamiento).

## **Escalabilidad Horizontal**

Agregar nodos adicionales para adaptarse a la carga de trabajo. Si la aplicación o el sistema están llegando a su punto crítico, entonces se agregan nodos adicionales y se divide la carga entre los distintos nodos. En Jenkins, lo podemos conseguir mediante "Slaves".

# **Manejo Básico de Usuarios**

* Se pueden crear nuevos usuarios y asignarles diferentes permisos, esto con el fin de poder saber en todo momento, o auditorías quien hizo que.
* Lo ideal es no compartir mismos usuarios ni misma contraseña.
* La autenticación se puede dar por medio de login con Github o Google, esto con el uso de plugins.

Para crear, eliminar, editar un usuario: Ir a Manage Jenkins/ Manage Users/ Create user (en caso de crear). Para editar o borrar solo se debe dar clic en ++user id++ deseado, después elegir opción Configurar o Borrar.

# **¿Qué es un Job?**

La parte más importante de Jenkins, los Jobs (los trabajos que ejecuta). Jenkins puede hacer varios trabajos al tiempo, esto es controlado por el Build Executor. Por cada job, Jenkins crea un folder dentro de su workspace (/var/lib/jenkins/workspace/). Un build es una ejecución de un Job. Cada job tiene su Build History.

# **Conectándonos a GitHub**

Es posible conectar un repositorio de GitHub a Jenkins para que cada vez que exista un push se haga un build del source code. Para que esto sea posible debemos realizar cambios tanto en Jenkins como en GitHub.

**En Jenkins**:

1. Debemos tener el GitHub plugin instalado.
2. Al crear el Job, debemos marcar el SCM la opción de **Git**, y pegar la URL del repositorio. (Para esta acción el host de Jenkins debe tener instalado Git)
3. En el apartado "**branches to build**" si dejamos en blanco tomara en cuenta cualquier branch.
4. En "**Build Triggers**" debemos marcar la opción "**GitHub hook trigger for GITScm polling**"

**En GitHub**:

1. Vamos al repositorio de GitHub.
2. Entramos en **Settings** -> **Webhooks**.
3. Añadimos un nuevo **Webhook**.
4. Añadimos la Payload URL. (Si la URL no acaba en **/github-webhook/** GItHub lanzara un error.)
5. Marcar "**Just the push event**"

# **¿Qué es un Pipeline?**

Es un conjunto de procesos automatizados que se ejecutan de manera secuencial para **construir**, **probar** y **desplegar** software de manera eficiente y confiable. Un pipeline de CI/CD permite a los equipos de desarrollo automatizar tareas repetitivas y eliminar errores humanos al implementar cambios en el código.

# **¿Qué son los Slaves?**

* Permite la ejecución de los Jobs distribuidamente.
* Se conecta a Jenkins master para delegar Jobs.
* Escala horizontalmente.