

JENKINS - PARTE I

Facilitador : Daniel Ojcius

- **Agenda. Un poco de DevOps**
- **CI/CD Que es?**
- **Etapas de desarrollo**
- **Como funciona CI/CD**
- **Jenkins: Arquitectura y funcionamiento**
- **Componentes de Jenkins**
- **Vamos a Instalarlo**
- **Proyectos**

Un poco de DevOps.

Los diferentes equipos que trabajan en una empresa están compuestos por varias personas. Estas personas son las encargadas de llevar a cabo cambios en el código para proveer funcionalidades.

DevOps permite que los roles que antes estaban aislados (desarrollo, operaciones de TI, ingeniería de la calidad y seguridad) se coordinen y colaboren para crear productos mejores y más confiables. Al adoptar una cultura de DevOps junto con prácticas y herramientas de DevOps, los equipos adquieren la capacidad de responder mejor a las necesidades de los clientes, aumentar la confianza en las aplicaciones que crean y alcanzar los objetivos empresariales en menos tiempo.

Que es CI/CD

CI/CD es un método para distribuir aplicaciones a los clientes con frecuencia mediante el uso de la automatización en las etapas del desarrollo de aplicaciones. Los principales conceptos que se

atribuyen a la CI/CD son la integración continua, la distribución continua y la implementación continua. La CI/CD es una solución para los problemas que puede generar la integración del código nuevo a los equipos de desarrollo y de operaciones (también conocida como "Integration Hell").

Etapas de desarrollo: automatizacion

Cuando un equipo de desarrollo introduce una nueva característica a su producto, existen varias etapas hasta llegar al cliente. Para la integración del código, podrían nombrarse 3 etapas (aunque podría haber más) las cuales permiten fusionar las nuevas construcciones con el código ya existente.

La codificación: siendo la etapa donde los desarrolladores construyen las nuevas características. Aquí es donde se desarrolla la lógica del programa.

La compilación: Donde una vez programada la nueva característica, se traduce el código al idioma de la máquina.

Las pruebas: Aquí se realizan diferentes pruebas que permiten extraer métricas acerca de la calidad del código y los posibles errores que pudiesen existir.

Como funciona CI/CD

Los desarrollos trabajan en conjunto en una aplicación y van modificando el código cada uno por su cuenta, luego esas modificaciones o cambios se envían al repositorio al source code repository y luego al server ci. Cada cambio pasa automáticamente por la herramienta de compilación y se realizan los tests pertinentes. Luego se notifica al equipo los cambios que suceden.

Es un procedimiento iterativo que persigue entregar al usuario final un producto de valor y a los equipos de desarrollo una métrica acerca de lo que construyen.

Que es Jenkins?

Jenkins es una herramienta que nos ayuda a crear pipelines, inicialmente se le conoce como servidor de integración continua o CI server, sin embargo también podemos definir los pasos para el continuous delivery y el continuous deployment. En resumen con Jenkins podemos configurar el pipeline para los componentes de software que tengamos

Arquitectura.

Jenkins como motor de automatización necesita una arquitectura para realizar sus trabajos. El mismo está compuesto por dos componentes principales:

- Maestro: sus principales tareas son programar los builds. Permite enviar las compilaciones a los workers y registrar el resultado de los builds. También ayudan a mantener la configuración de nuestro entorno de Jenkins.
- Worker: Son máquinas programadas para construir proyectos. Tienen como tarea ejecutar jobs enviados por el maestro. También recopilan métricas e información y se las envían al maestro.

Como funciona Jenkins.

Jenkins como motor de automatización permite trabajar de múltiples maneras. A modo de este ejemplo, hemos adoptado la forma de branching model (lo veremos más adelante). Sin embargo, a grandes rasgos el modo de funcionamiento es el siguiente:

1. El desarrollador sube sus cambios al repositorio de Git
2. Jenkins se pone en modo oyente de ese repositorio.
3. Si existe algún cambio en ese repositorio, Jenkins lanza el o los respectivos Jobs.

Como dijimos, Jenkins tiene diferentes formas de automatización, por lo cual la forma más clara de verlo, es como un servidor que tiene capacidad de cómputo para correr código de automatización para satisfacer los pasos de CI/CD.

Vamos a instalarlo.

Para instalarlo primero debemos verificar si tenemos docker instalado en nuestra maquina. Para verificarlo debemos correr el siguiente comando:

> Docker version

En caso de no estar instalado seguimos la siguiente guia de instalacion: <https://docs.docker.com/compose/install/>

Luego seguimos los siguientes pasos:

1. Clonamos el repositorio o lo descargamos de <https://github.com/Daniel-MundosE/Jenkins-training>:
 - a. git clone <https://github.com/Daniel-MundosE/Jenkins-training.git>
2. Dentro del proyecto que descargamos vamos a la carpeta recursos.
3. Listamos los archivos y ahi debemos ver un archivo llamado docker-compose.yml
4. Ejecutamos el comando: **docker-compose up -d**, esto comenzara a descargar la imagen de jenkins y lanzar nuestro container.
5. Luego ejecutamos el comando **docker ps** y deberiamos ver nuestro container de Jenkins corriendo. En caso de que eso no suceda debemos ejecutar el siguiente comando:

sudo chown 1000 jenkins_home

Corremos nuevamente el comando: **docker-compose up -d**

Y deberiamos ver nuestro contenedor arriba.

Luego abrimos nuestro navegador y colocamos:

<http://localhost:8080/>

en donde se nos abra una pantalla de jenkins y se nos pedira una clave. Para acceder a esta clave corremos el siguiente comando:

```
> docker exec jenkins cat  
/var/jenkins_home/secrets/initialAdminPassword
```

Copiamos la clave en el text box de jenkins y damos en continue.

A continuacion nos preguntara si queremos instalar plugins.

Presionamos en --> Install suggested plugins

Esperamos a que termine de instalarse.

Luego nos llevara a una ventana para llenar usuario, contrasenia, nombre e email, rellenamos y le damos continuar hasta llegar al boton 'start using jenkins', presionamos y listo! Ya tenemos jenkins

Proyecto estilo libre.

Característica que nos permite automatizar tareas que podrían ser repetitivas, permitiéndonos ejecutarlas cada cierto tiempo determinado, desde algún repositorio o con algún tipo de parametro. Puede servir para automatizar configuración o tareas que deber ser planificadas.

Para ello vamos a Jenkins -> Nueva Tarea -> Colocamos un nombre referencial para el proyecto -> elegimos proyecto estilo libre -> seleccionamos opciones

Para la parte de código debemos ir hasta nueva ejecución y elegimos shell, listo podemos programar una tarea!! A automatizar!

Proyecto estilo libre parametrizado.

Así como también podemos automatizar diferentes tareas para poder realizar una acción, dicha acción puede tener parametros de entrada.

Jenkins nos brinda un mecanismo para disponibilizar los nombres de los parametros y personalizarlo segun la realidad planteada

