



Certified Tech Developer

The Ultimate Degree

Infraestructura II

Actividad obligatoria

Dificultad: media

Ejercitación: Vemos el circuito de CD

En esta ocasión vamos a levantar un pequeño entorno para jugar con una implementación simple del circuito de CD con herramientas del mundo real.

¿Qué vamos a hacer? Seguimos este paso a paso:

- Empezamos con una app Java empaquetada: 'holamundo.jar'.
- La guardamos como *release* en un gestor de artefactos.
- Ponemos productivo el *release*. ¿Cómo? Corriendo un contenedor con un servidor web que nos muestra la salida en consola de nuestra aplicación.

¿Qué necesitaremos?

Para realizar este ejercicio necesitamos contar con:

- Docker instalado ([ayuda](#)).
- Docker-compose instalado ([ayuda](#)).
- Zip con los archivos para levantar el entorno de práctica ([link al archivo](#)).
- Conexión a internet.

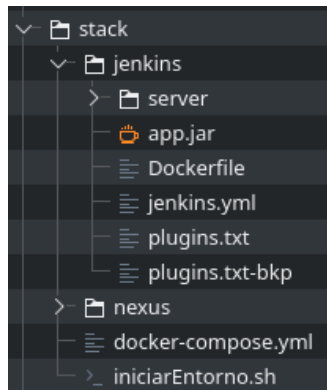
¿Qué entorno utilizaremos?

Dentro del zip vamos a encontrar distintos archivos que definen nuestro entorno:

- Un **contenedor de Jenkins** donde está implementado el circuito de CD.

- Un contenedor **de Nexus 3** que cumple la función de repositorio de artefactos (nuestra bodega).

Además podremos encontrar definiciones de un tercer contenedor que utilizaremos para levantar nuestro entorno de despliegue (producción).



¡Levantemos el entorno!

- Nos situamos en la carpeta en que descargamos el zip con los archivos y lo descomprimos ejecutando:

```
$ unzip stack.zip
```

- Ingresamos a la carpeta stack que se generó:

```
$ cd stack
```

- Para finalizar, ejecutamos el script que inicia el entorno:

```
$ ./iniciarEntorno.sh
```

Es probable que este último paso nos pida nuestra contraseña para configurar unas carpetas. Luego, demorará un tiempo en levantar todos los recursos.

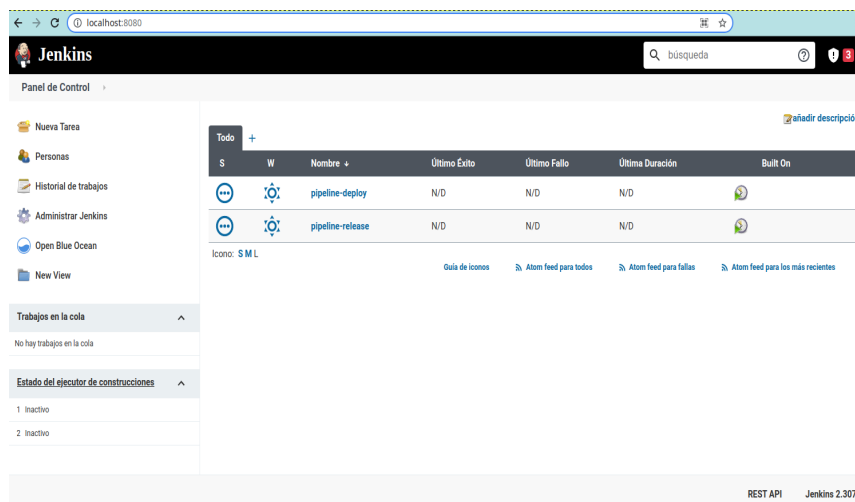
¡Verifiquemos que todo esté andando! Vemos en la consola:

```
nexus
nexus
nexus
nexus
nexus
Started Sonatype Nexus OSS 3.33.1-01
```

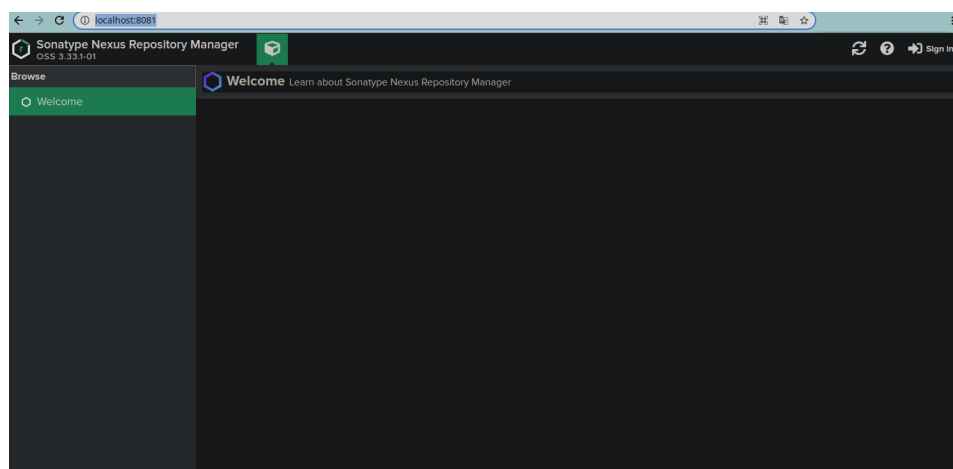
Una vez que hayamos visto en la consola, podemos seguir con la guía y visitar los links de Nexus y Jenkins. Para hacerlo, tengamos en cuenta que Nexus tarda más en inicializar que Jenkins.

Para comprobar los servicios web de cada herramienta, abrimos las siguientes url:

- **Jenkins:** <http://localhost:8080/>



- **Nexus3:** <http://localhost:8081/>



¡Manos a la obra!

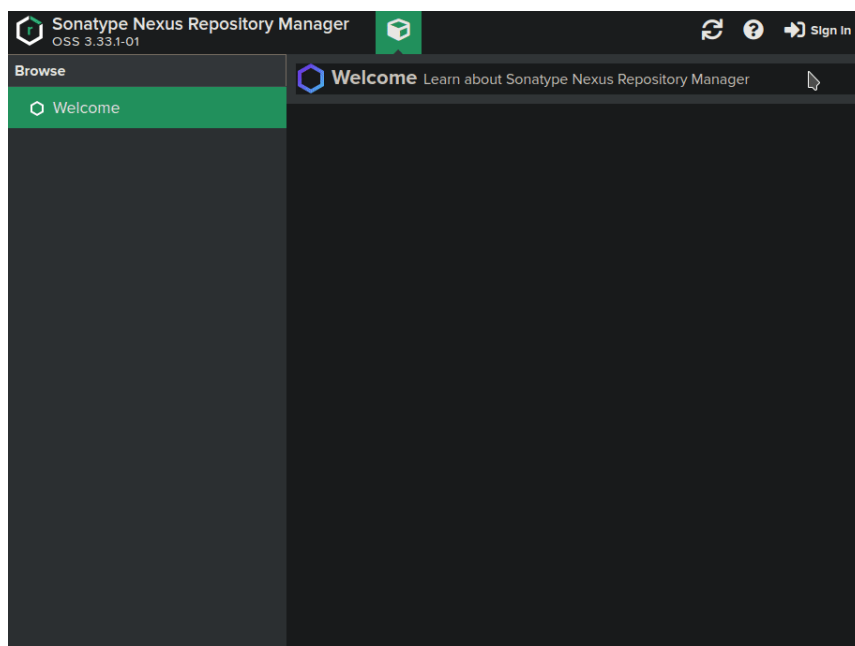
¡Muy bien! Llegamos hasta aquí con **Nexus** y **Jenkins** andando. Al iniciar estas herramientas -con **Docker compose**- se les pasaron una serie de parámetros que las dejaron configuradas y listas para usar. Vamos a explorar cada una de ellas.

Curioseando Nexus

Nexus es un administrador de repositorios de artefactos muy popular que puede incluir paquetes, librerías, ejecutables de diversos lenguajes de programación, ¡y hasta imágenes de contenedores! Nosotros lo utilizaremos para almacenar un `holamundo.jar` en su versión de release 1.0.0.

Para poder ver los repositorios en Nexus debemos iniciar sesión: el usuario y la contraseña son la palabra `nexus`. Luego nos pedirá completar un pequeño *wizard*. ¡Ya estamos listos para navegar!

En nuestro Nexus tendremos un repositorio llamado *myrepo* donde vamos a guardar nuestro artefacto. Para verlo debemos ir a “Browse” en la barra lateral izquierda. Haciendo doble click en “myrepo” entraremos en él y veremos que aún no contiene ningún artefacto.



¡Ejecutando nuestro circuito de CD en Jenkins!

Nuestro circuito de CD consta de dos etapas.

- Release: prueba del artefacto, archivado como un *release* en Nexus.
- Despliegue: obtención del release desde Nexus y despliegue en un servidor web (contenedor).

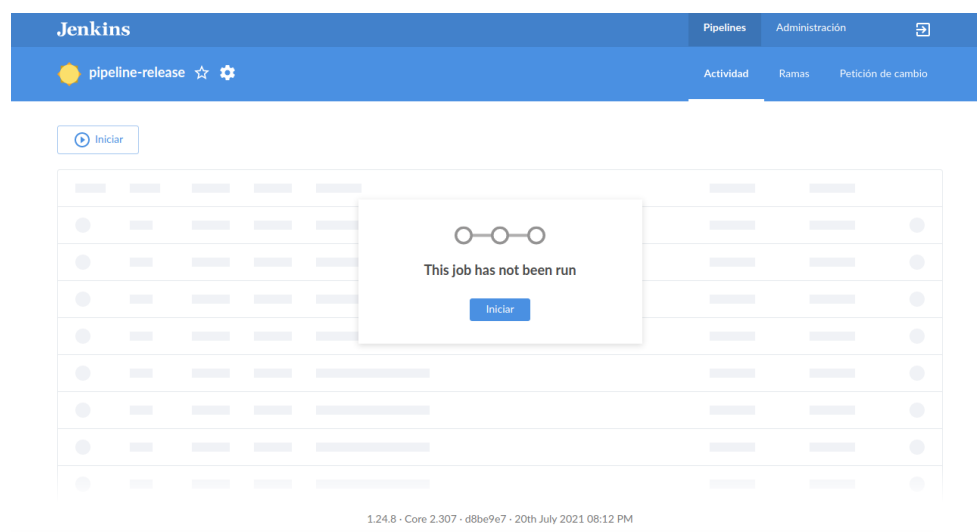
En Jenkins encontraremos dos pipelines.

- **pipeline-release**: vamos a ejecutar nuestra aplicación “*holamundo.jar*” y testearla. Si no rompe, subirá como un artefacto nuevo en Nexus.
- **pipeline-deploy**: este pipeline hará la descarga del artefacto desde Nexus, levantará un contenedor que lo incluya y disponibilice como un sitio web listo para su uso.

¡Etapas release!

Para visualizar mejor el proceso vamos a ingresar al frontend “Blue Ocean” de Jenkins mediante el siguiente [link](#).

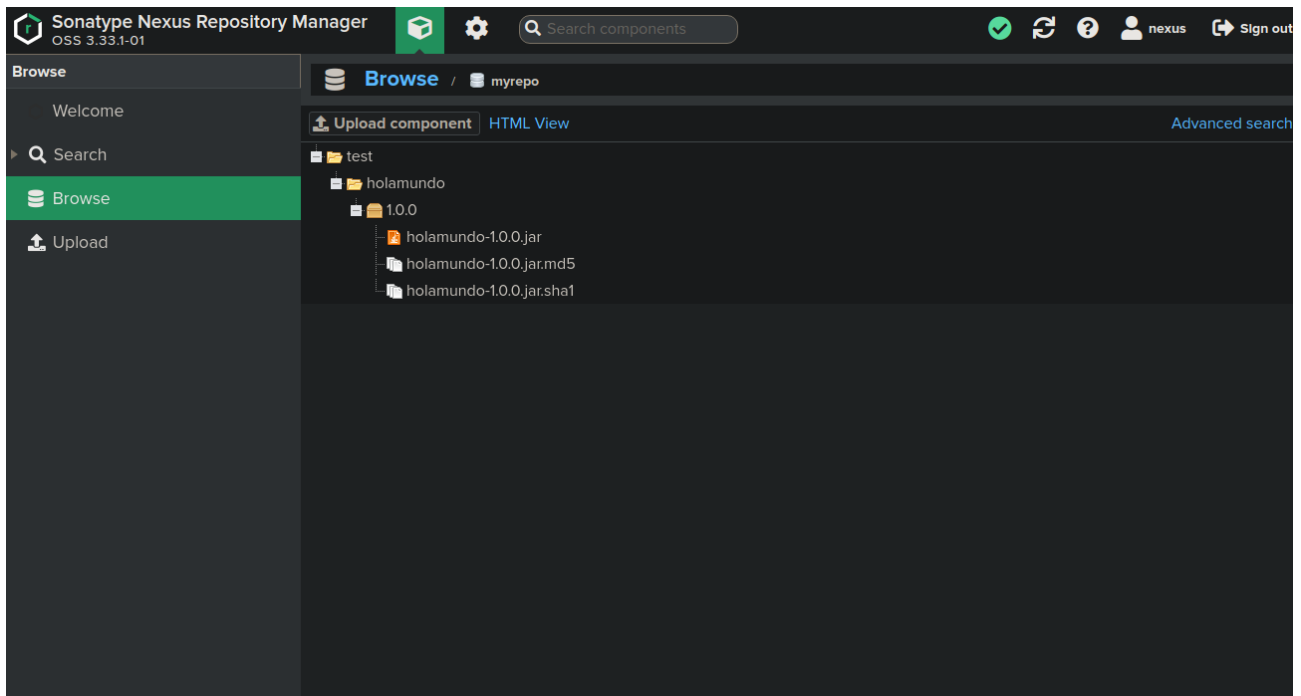
Hacemos clic en “Iniciar” para comenzar el proceso. Al instante veremos la ejecución del pipeline. Para ver el avance en detalle hacemos clic sobre el proceso que está corriendo.



Veamos el resultado en Nexus

Vamos al repositorio de nuestro Nexus para ver nuestro artefacto:

<http://localhost:8081/#browse/browse:myrepo>



¡Etapa de despliegue!

Nuevamente vamos a ingresar al frontend “Blue Ocean” de Jenkins mediante el siguiente link: <http://localhost:8080/blue/organizations/jenkins/pipeline-deploy/activity>

Hacemos clic en “Iniciar” para comenzar el pipeline. Esta ejecución tardará algo más de tiempo porque generará el container para nuestro entorno de producción.

¡Operación!

Ingresamos a la siguiente url para ver el mensaje que devuelve nuestra aplicación java:

<http://localhost:3001/cgi-bin/app>

Conclusión

En esta práctica pudimos observar y experimentar con herramientas del mundo real cómo es el proceso de Continuous Deploy / Delivery. Si bien es un proceso complejo de construir de forma completa, una vez configurado y puesto a punto ¡nos resulta de gran ventaja!

¡Imaginemos tener que hacer todo esto a mano sin pipelines! Podría ser divertido la primera vez, pero este proceso puede suceder cientos de veces al mes en una empresa y más aún si está automatizado en un pipeline. ¡He ahí el plus de esta práctica!