



Certified Tech Developer

The Ultimate Degree

Infraestructura II

Actividad obligatoria e individual

Dificultad: Media

Guía: Jenkins y Docker

Configuración de Docker en Jenkins

En esta guía veremos cómo configurar Jenkins para utilizar Docker como un ambiente sencillo de prueba local.

Configuración Jenkins en contenedor

En este caso, tenemos andando Jenkins en un contenedor servido por Docker en nuestro entorno. Ya tenemos Docker funcionando, ahora tenemos que hacer que el contenedor de Jenkins pueda correr los comandos de Docker, para esto, necesitamos los binarios de Docker en el contenedor y acceso al servicio de Docker del host.

Necesitamos ingresar al cliente de Docker y conectarnos al servidor que maneja todos los contenedores en nuestro host o máquina. Con este objetivo, vamos a levantar Jenkins nuevamente creando una imagen propia con el siguiente Dockerfile:

```
FROM docker.io/jenkins/jenkins:2.307-jdk11
USER root
ENV JENKINS_UC_DOWNLOAD http://mirrors.jenkins-ci.org
ENV CURL_OPTIONS -sSfLk -C -
RUN apt-get update && apt-get install -y tree nano curl sudo ; \
    curl https://get.docker.com/builds/Linux/x86_64/docker-latest.tgz | tar
```

```
xvz -C /tmp/ && mv /tmp/docker/docker /usr/bin/docker ; \
    curl -L
"https://github.com/docker/compose/releases/download/1.23.1/docker-compose-$(
uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose ; \
    chmod 755 /usr/local/bin/docker-compose ; \
    apt-get clean --dry-run; \
    usermod -a -G sudo jenkins ; \
    echo "jenkins ALL=(ALL:ALL) NOPASSWD:ALL" >> /etc/sudoers ;

USER jenkins
```

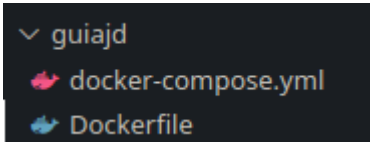
Las líneas agregadas están instalando una serie de herramientas, incluidos los binarios de Docker.

Ahora nos falta hacer que el contenedor tenga acceso al server de Docker. Vamos a hacerlo en el momento de levantarlo, montando el socket del server como un volumen. Tal como se ve en el siguiente **docker-compose.yaml**:

```
version: '3'
services:
  jenkins:
    container_name: jenkins
    build:
      context: .
      dockerfile: Dockerfile
    restart: always
    privileged: true
    user: root
    ports:
      - "8080:8080"
      - "50000:50000"
    volumes:
      - ./jenkins_home:/var/jenkins_home
      - /var/run/docker.sock:/var/run/docker.sock
```

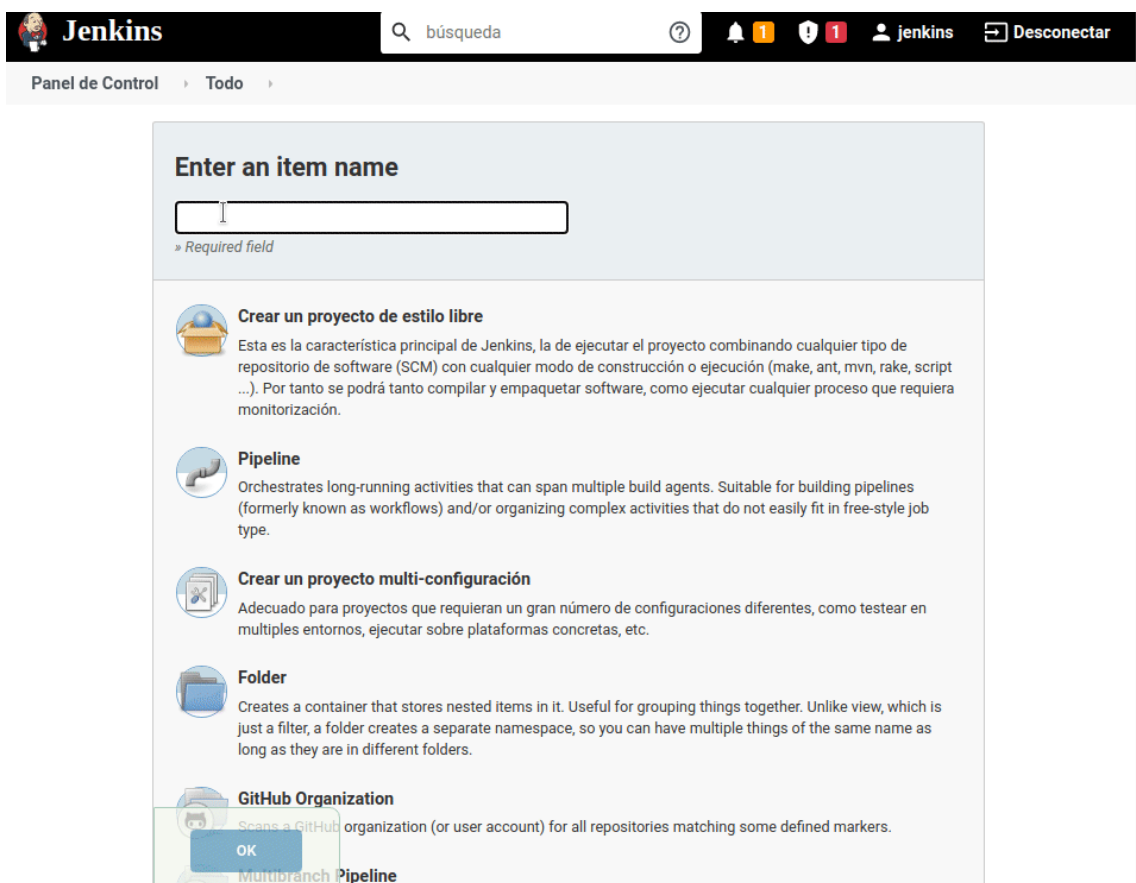
En la última línea se ve el socket montado en el contenedor.

Una vez finalizado, encontramos en nuestra carpeta estos dos archivos:



Ejecutamos el comando **docker-compose up** y terminamos de configurar nuestro Jenkins.

Podemos probar esta configuración generando un job y agregando un paso de build para correr el script **docker ps**, que deberá mostrar todos los contenedores que tenemos funcionando en nuestro host donde por supuesto estará el de Jenkins, como muestra la siguiente animación:



Conclusión

En esta guía se consigue hacer que Jenkins –corriendo en un contenedor servido por Docker– pueda controlar al mismo Docker.

Esta forma de levantar Jenkins no es segura para ambientes productivos, pero es muy sencilla y útil para debugear y trabajar en ambientes locales de desarrollo.