



Configuración de Servidores de Pruebas con Docker	
CODIFICACION:	DTI-JC-MU-09-2024-002
FECHA:	2024-09-06

Manual Técnico

Consortio de Gobiernos Autónomos Provinciales del Ecuador

Dirección de Tecnologías e Información

Configuración de Servidores de Pruebas con Docker

Dirección de Tecnologías e Información

Elaborado por:

Mgs. Javier Chillogallo

Versión 1.0

Fecha de emisión: 6 de septiembre de 2024

Quito – Ecuador



Configuración de Servidores de Pruebas con Docker	
CODIFICACION:	DTI-JC-MU-09-2024-002
FECHA:	2024-09-06

Contenido

1.	¿Qué es Docker?	3
1.1.	¿Para qué se utiliza Docker?	3
1.2.	Características clave de Docker	3
2.	Instalar Docker en Linux	4
3.	Desplegar una Máquina con Apache utilizando Docker	5
4.	Construir y desplegar el contenedor	5
5.	Creación del Docker para el back end	6
6.	Comandos útiles de Docker	7
7.	Recomendaciones	7
8.	Firmas	8

1. ¿Qué es Docker?

Docker es una plataforma de código abierto que permite automatizar el despliegue de aplicaciones dentro de contenedores ligeros. Estos contenedores agrupan todo lo que una aplicación necesita para ejecutarse, como el código, las bibliotecas y las dependencias, asegurando que funcione de manera consistente en diferentes entornos.

1.1. ¿Para qué se utiliza Docker?

- **Portabilidad:** Los contenedores Docker se pueden ejecutar en cualquier sistema que tenga Docker instalado, lo que facilita la migración entre entornos de desarrollo, prueba y producción.
- **Aislamiento:** Docker aísla las aplicaciones y sus dependencias, evitando conflictos con otras aplicaciones en el mismo sistema.
- **Eficiencia:** Los contenedores son ligeros, ya que comparten el kernel del sistema operativo, lo que reduce el uso de recursos en comparación con las máquinas virtuales.
- **Automatización y DevOps:** Facilita la integración y el despliegue continuo (CI/CD), permitiendo a los equipos de desarrollo y operaciones colaborar de manera más eficiente.

1.2. Características clave de Docker

Algunas características clave de Docker son:

1. **Contenedores ligeros:** Docker utiliza contenedores que son más ligeros y rápidos que las máquinas virtuales, ya que comparten el kernel del sistema operativo, lo que ahorra recursos y espacio.
2. **Portabilidad:** Los contenedores Docker pueden ejecutarse en cualquier sistema que tenga Docker instalado, lo que permite trasladar aplicaciones fácilmente entre entornos de desarrollo, prueba y producción.
3. **Aislamiento de aplicaciones:** Cada contenedor se ejecuta de manera aislada, lo que evita conflictos entre aplicaciones y sus dependencias, proporcionando entornos consistentes y controlados.
4. **Facilidad de despliegue:** Docker simplifica el proceso de empaquetado, distribución y despliegue de aplicaciones, ya que todo lo necesario para ejecutarlas está dentro del contenedor.
5. **Gestión de versiones:** Permite versionar imágenes de contenedores, lo que facilita el control de versiones y actualizaciones de las aplicaciones.

6. **Integración con DevOps:** Docker es compatible con herramientas de integración y entrega continua (CI/CD), lo que ayuda a automatizar los procesos de desarrollo y despliegue de software.
7. **Docker Hub:** Un repositorio público donde los desarrolladores pueden compartir y acceder a imágenes de contenedores preconfiguradas, facilitando la reutilización y la colaboración.
8. **Redes y almacenamiento flexibles:** Docker permite crear redes entre contenedores y asignar volúmenes de almacenamiento persistente, mejorando la flexibilidad para arquitecturas más complejas.

Estas características hacen que Docker sea ideal para el desarrollo ágil y el despliegue eficiente de aplicaciones.

2. Instalar Docker en Linux

A continuación se muestra el despliegue de una maquina en Linux(Ubuntu), los comandos varían dependiendo de la distribución de Linux pero es el mismo concepto.

- **Actualizar los paquetes existentes**

Abrir una terminal y ejecutar los siguientes comandos para asegurarse de que el sistema esté actualizado.

```
sudo apt update  
sudo apt upgrade
```

- **Instalar Docker**

Ejecutar los siguientes comandos para instalar Docker

```
sudo apt install apt-transport-https ca-certificates curl software-properties-common  
  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg  
  
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]  
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
  
sudo apt update  
  
sudo apt install docker-ce
```

- **Verificar la instalación de Docker**

Una vez instalado, verificar que Docker esté correctamente ejecutando el comando

```
sudo docker --version
```

3. Desplegar una Máquina con Apache utilizando Docker

- **Crear el directorio del proyecto**

Crear un directorio donde se va a trabajar

```
mkdir my-apache-server  
cd my-apache-server
```

- **Crear el archivo Dockerfile**

Dentro del directorio, crear un archivo llamado Dockerfile con el siguiente contenido:

```
FROM httpd:2.4  
# Esta línea ayuda a desplegar el código que este en el Carpeta CongopeSFG a la carpeta htdocs de apache  
COPY ./CongopeSFG/ /usr/local/apache2/htdocs/  
  
# Comando para iniciar Apache  
CMD ["httpd-foreground"]
```

4. Construir y desplegar el contenedor

- **Construir la imagen de Docker**

Ejecutar el siguiente comando en el directorio donde se encuentra el Dockerfile

```
## CONSTRUIR EL CONTENEDOR APACHE CON EL DOKERFILE APACHE  
docker build -t apache-front .
```

Este comando crea una imagen de Docker llamada apache-front.

- **Ejecutar el contenedor**

Para desplegar el servidor Apache usando la imagen recién creada, ejecutar el siguiente comando:

```
## EJECUTAR EL CONTENEDOR - EN ESTE CASO ADICIONALMENTE COMPARTIMOS UN FICHERO CON EL CODIGO FUENTE PARA RECARGAR LOS ARCHIVOS
```

```
docker run -d --name sfgprov -p 80:80 -v $(pwd)/CodigoFront:/usr/local/apache2/htdocs/ apache-front
```

- **Verificar la instalación**

Abrir un navegador y acceder a http://ip_equipo. Debería aparecer la aplicación que este dentro del código fuente.

5. Creación del Docker para el back end

Los pasos son los mismos que para el front end, sin embargo, acá se muestra los archivos de configuración para el proyecto SFGProv

A continuación, se muestra el código para la creación del Docker para el backEnd

```
# Utiliza la imagen oficial de .NET Core SDK como base
FROM mcr.microsoft.com/dotnet/aspnet:8.0 AS base
EXPOSE 8080

# Copiar los archivos de tu aplicación .NET Core
COPY ./CongopeBack/ /app/

# Instalar las dependencias necesarias para .NET
RUN apt-get update && \
    apt-get install -y wget apt-transport-https

# Añadir la llave GPG y el repositorio de paquetes de Microsoft
RUN wget https://packages.microsoft.com/config/debian/10/packages-microsoft-prod.deb -O packages-microsoft-prod.deb && \
    dpkg -i packages-microsoft-prod.deb && \
    rm packages-microsoft-prod.deb

# O si solo necesitas el runtime:

RUN apt-get update && \
    apt-get install -y aspnetcore-runtime-8.0

# Configurar el entorno de trabajo

WORKDIR /app

# Configurar la aplicación .NET Core para ejecutarse en Kestrel

CMD ["dotnet", "Congope.Empresas.dll"]
```

A continuación, se muestra el código para el despliegue del Back:

```
## CONSTRUIR EL CONTENEDOR APACHE CON EL DOKERFILE KESTRELL - DOTNET

docker build -t dotnet-back .

## EJECUTAR EL CONTENEDOR - EN ESTE CASO ADICIONALMENTE COMPARTIMOS UN FICHERO CON EL CODIGO FUENTE PARA RECARGAR LOS ARCHIVOS

docker run -d --name back-sfgprov -p 5000:8080 -v $(pwd)/CongopeBack:/app/ dotnet-back
```

Tomando en consideración que el despliegue del Docker se lo hace en la misma máquina del front se considera desplegar los servicios en el puerto 8080

- **Verificar la instalación**

Abrir un navegador y acceder a http://ip_equipo:5000. Debería aparecer la aplicación que muestra los servicios.

6. Comandos útiles de Docker

A continuación, se muestra algunos comandos que nos pueden ayudar

```
## COMANDOS ADICIONALES
## ELIMINAR DOCKERS
docker rm $(docker ps -a -f "status=exited" -q)

## EJECUTAR CONSOLA
docker exec -it back-sfgprov /bin/bash

## VER DOCKER EN EJECUCION
docker ps -a

## SI SE REQUIERA REINICIAR POR DEPENDENCIAS USAR EL SIGUIENTE COMANDOS
docker restart back-sfgprov
```

7. Recomendaciones

- La configuración mostrada en este manual proporciona una base inicial para el uso de Docker en el despliegue de aplicaciones simples, como un servidor Apache. Sin embargo, es fundamental que el equipo de trabajo realice una parametrización más exhaustiva en aspectos como seguridad y gestión, con el fin de asegurar un entorno robusto y eficiente.
- Se recomienda configurar certificados SSL/TLS para las conexiones que involucren servidores web dentro de los contenedores.
- Se debe limitar los permisos de los contenedores y usuarios dentro del entorno Docker para evitar accesos innecesarios o privilegiados.

- Se debe mantener actualizados los componentes del sistema operativo en los contenedores y las herramientas utilizadas dentro de ellos.
- Se recomienda investigar otras tecnologías complementarias, como Kubernetes, que permiten una administración más eficiente de contenedores a gran escala, y la adopción de soluciones avanzadas para la seguridad, orquestación y optimización del ciclo de vida de los contenedores en entornos de producción.

8. Firmas

	CARGO	FIRMA	FECHA
ELABORADO POR	Mgs. Javier Chillogallo Profesional Contratado		2024-09-06