

# Containers

Sebastian I. Rodríguez

Instituto Politécnico Superior Gral. San Martín

Junio 2019

# Índice

- 1 **Introducción**
  - Aclaracion Preliminar
  - Definición de Container?
  - Qué es Docker?
  - Container vs Máquina Virtual
- 2 **Container Image**
  - Qué es una Imagen de Container?
  - Que es un DockerFile?
- 3 **Volumes**
  - Qué es un Volume?
  - Volume vs Bind Mounts
- 4 **Conexion entre Containers**
  - Introduccion
  - Cómo se conectan?
  - Red Bridge
  - Red Host
  - Red None
- 5 **Kubernetes**

# Seccion

- 1 **Introducción**
  - Aclaracion Preliminar
  - Definición de Container?
  - Qué es Docker?
  - Container vs Máquina Virtual
- 2 **Container Image**
  - Qué es una Imagen de Container?
  - Que es un DockerFile?
- 3 **Volumes**
  - Qué es un Volume?
  - Volume vs Bind Mounts
- 4 **Conexion entre Containers**
  - Introduccion
  - Cómo se conectan?
  - Red Bridge
  - Red Host
  - Red None
- 5 **Kubernetes**

# Aclaración

A lo largo de esta presentación haremos referencia a características específicas a Docker, porque este es el software más popular en cuanto a manipulación de Containers.

# Qué es un Container?

- Es una unidad de software que contiene un programa y todas las dependencias necesarias para su funcionamiento.
- Introduce una abstracción que garantiza el funcionamiento del programa independientemente de las características del ambiente en el que se encuentre el container.
- Aisla al programa del entorno en que se ejecuta.

# Qué es Docker?

Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de containers.

Utiliza características de aislamiento de recursos del kernel Linux, como `cgroups` y espacios de nombres (`namespaces`) para permitir que containers independientes se ejecuten en una sola instancia de Linux, evitando el coste de iniciar y mantener máquinas virtuales.

# Container vs Máquina Virtual

## Maquina Virtual

- Para aislar  $n$  procesos se necesitan  $n$  SO operativos virtualizados.
- Virtualizaciones controladas por Hypervisor.
- Gran consumo de recursos.
- SO virtualizado puede contener bibliotecas innecesarias.

## Container

- Cada container se ejecuta sobre un único Kernel.
- Ahorra recursos
- Containers controlados, en el caso de Docker, por Docker Engine.
- Cada containers contiene las dependencias minimas y necesarias

# Seccion

1

## Introducción

- Aclaracion Preliminar
- Definición de Container?
- Qué es Docker?
- Container vs Máquina Virtual

2

## Container Image

- Qué es una Imagen de Container?
- Que es un DockerFile?

3

## Volumes

- Qué es un Volume?
- Volume vs Bind Mounts

4

## Conexion entre Containers

- Introducción
- Cómo se conectan?
- Red Bridge
- Red Host
- Red None

5

## Kubernetes



# Qué es una Imagen de Container?

Básicamente un Container es la instancia de su Imagen. En la imagen del Container se almacena la configuración del mismo.

En esta configuración se establece el proceso a ejecutar y sus dependencias necesarias, debiéndose especificar las versiones de las mismas.

Esta configuración se almacena en una estructura de árbol. A partir de una Imagen se puede crear un Contenedor y viceversa, podemos crear y configurar un Contenedor manualmente y luego generar una Imagen.

Se denomina DockerFile al archivo en que se almacena la configuracion de una Imagen de Container en Docker.

# Seccion

- 1 **Introducción**
  - Aclaracion Preliminar
  - Definición de Container?
  - Qué es Docker?
  - Container vs Máquina Virtual
- 2 **Container Image**
  - Qué es una Imagen de Container?
  - Que es un DockerFile?
- 3 **Volumes**
  - Qué es un Volume?
  - Volume vs Bind Mounts
- 4 **Conexion entre Containers**
  - Introducción
  - Cómo se conectan?
  - Red Bridge
  - Red Host
  - Red None
- 5 **Kubernetes**

El Volume es el mecanismo más común dentro de Docker para persistir información generada y usada por los Containers.

## Volume



## Bind Mount

- Consiste en montar un archivo o directorio perteneciente al host, dentro de un container. Este archivo se referencia por su path.
- Buen rendimiento.
- Crea dependencias por fuera del Container y del Docker Engine, que no es lo ideal.
- No puede utilizarse desde la consola de Docker.

# Seccion

- 1 **Introducción**
  - Aclaracion Preliminar
  - Definición de Container?
  - Qué es Docker?
  - Container vs Máquina Virtual
- 2 **Container Image**
  - Qué es una Imagen de Container?
  - Que es un DockerFile?
- 3 **Volumes**
  - Qué es un Volume?
  - Volume vs Bind Mounts
- 4 **Conexion entre Containers**
  - Introducción
  - Cómo se conectan?
  - Red Bridge
  - Red Host
  - Red None
- 5 **Kubernetes**

# Introducción

Como ya sabemos, la idea fundamental de los Containers, es aislar software. Sin embargo, en muchas aplicaciones es necesaria una forma de transmitir información entre Containers, para esto, se crean mecanismos de conexión entre Containers.

Cabe aclarar que en las siguientes diapositivas se haga referencia a mecanicas específicas a Docker, al ser este el software de Containers en el que nos concentramos.

## Como se conectan los Containers?

En la comunicacion entre Containers lo que se hace básicamente es crear una red en la que se expone cada Container como un host. Esto implica que a cada container se le asigna su propia direccion IP y MAC, entre otros parametros.



# Redes: Bridge

Por defecto todos los containers se conectan a un bridge virtual de Ethernet, en `Docker` esta representado por la red `docker0`. Existen otros 2 tipos de redes por defecto en `Docker`, la red `host` y la red `none`

La red `host` añade el Container a la pila de protocolos del Host en el que se ejecuta. Esto significa que no existe ningun tipo de aislamiento entre el Container y el Host, en cuanto a redes se refiere.

Esto implica una correspondencia entre los puertos del Container y el Host.

Por ejemplo, cualquier servicio ejecutándose en un puerto X del Container, es accesible desde el puerto X del Host.

La red `none` agrega al Container a una pila de Protocolos específica a dicho Container. En este caso, el Container no tiene una interfaz de red.

# Seccion

- 1 **Introducción**
  - Aclaracion Preliminar
  - Definición de Container?
  - Qué es Docker?
  - Container vs Máquina Virtual
- 2 **Container Image**
  - Qué es una Imagen de Container?
  - Que es un DockerFile?
- 3 **Volumes**
  - Qué es un Volume?
  - Volume vs Bind Mounts
- 4 **Conexion entre Containers**
  - Introduccion
  - Cómo se conectan?
  - Red Bridge
  - Red Host
  - Red None
- 5 **Kubernetes**

# Summary

- The **first main message** of your talk in one or two lines.
- The **second main message** of your talk in one or two lines.
- Perhaps a **third message**, but not more than that.
- Outlook
  - Something you haven't solved.
  - Something else you haven't solved.