

01/10/2020

# Plus + Restaurant

Web Platform and App



Presentado por: Hanna Siddharththa Lizarraga Ceballos

# Plus + Restaurant

Web Platform and App



Plus + Restaurant

## Manual de infraestructura de los contenedores V.01

# Índice

Actual version .....	3
Tabla de contenido .....	3
Tabla de control y cambios .....	4
Objetivo .....	5
Alcance .....	6
Docker Compose .....	7
Configuración del gestor .....	9
Configuración de log .....	10
Diferencias .....	11
Conclusiones .....	12

## CONTROL DE DOCUMENTACIÓN

**Esta sección contiene el historial de versiones y cambios.**

### GENERAL DATA

<b>Author[s]</b>	Hanna S. Lizarraga Ceballos
<b>Authorizer[s]</b>	Alfredo Bolio
<b>Date</b>	01/Oct/2020

### VERSION HISTORY

VERSION	DATE	STATUS	COORDINATED BY:
1.0.0	01/10/2020	R	Hanna S. Lizarraga Ceballos

### CHANGES HISTORY

VERSION	DATE	CHANGES	COORDINATED BY:
1.0.0	01/10/2020	First Version	Alfredo Bolio

# Objetivo

Este documento tiene como objetivo explicar el uso de los contenedores a modo de gestores de base de datos y explicando la diferencia de estos con los gestores de base de datos de uso tradicional así como dar a conocer el proceso de manera detallada de los pasos a realizar para que pueda crearse un gestor de base de datos a modo contenedor.

# Alcance

La virtualización ha revolucionado la informática. La distribución de los recursos de una máquina física en varias máquinas virtuales ha entrado en el panorama de las tecnologías de la información en forma de virtualización de hardware. Este enfoque se basa en la emulación de componentes de hardware para poder utilizar varios servidores virtuales con su propio sistema operativo en un sistema de alojamiento común. Esto es lo que sucede a menudo en el ámbito del desarrollo de software, al tener que implementar entornos de prueba diferentes en un mismo ordenador. La virtualización también se encuentra en la base de productos de alojamiento web con servidor en la nube.

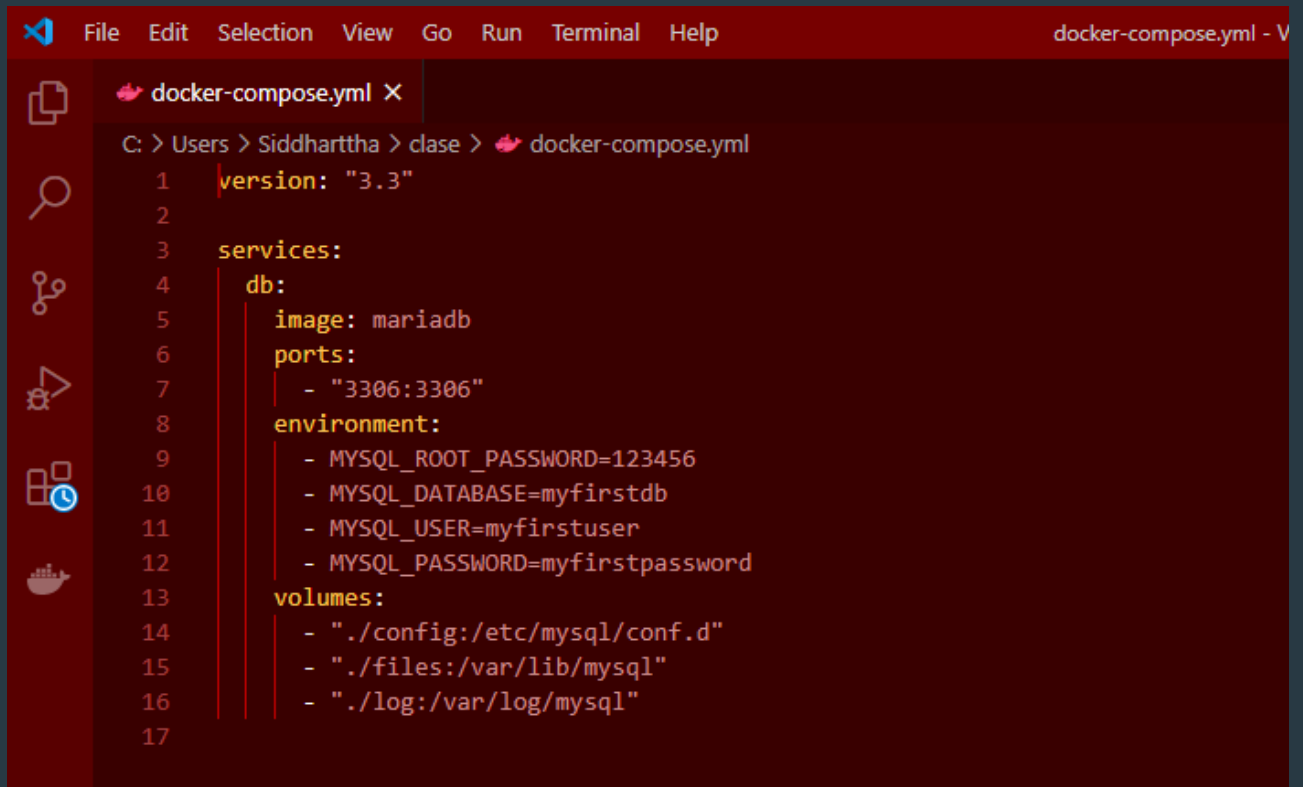
Una alternativa a la virtualización de hardware la constituye la virtualización del sistema operativo, por la cual diversas aplicaciones de un servidor se implementan en entornos virtuales aislados, los llamados contenedores, que funcionan en el mismo sistema operativo (virtualización basada en contenedores). Como las máquinas virtuales, cada una con su sistema operativo, los contenedores también brindan la posibilidad de usar en paralelo diversas aplicaciones, cada una con sus requisitos, en un mismo sistema físico. Como los contenedores no disponen de ningún sistema operativo propio, esta virtualización se diferencia por una instalación mucho más sencilla y un menor consumo (overhead), así como al fin y al cabo es un buen modelo de trabajo y no tiene un alto costo.

# Docker Compose

## ¿Qué es?

Docker Compose es una herramienta que permite simplificar el uso de Docker. A partir de archivos YAML es mas sencillo crear contenedores, conectarlos, habilitar puertos, volumenenes, etc.

## docker-compose.yml



```
File Edit Selection View Go Run Terminal Help
docker-compose.yml X
C: > Users > Siddhartha > clase > docker-compose.yml
1  version: "3.3"
2
3  services:
4    db:
5      image: mariadb
6      ports:
7        - "3306:3306"
8      environment:
9        - MYSQL_ROOT_PASSWORD=123456
10       - MYSQL_DATABASE=myfirstdb
11       - MYSQL_USER=myfirstuser
12       - MYSQL_PASSWORD=myfirstpassword
13     volumes:
14       - "./config:/etc/mysql/conf.d"
15       - "./files:/var/lib/mysql"
16       - "./log:/var/log/mysql"
17
```

# Explicación del código

Explicación paso por paso de cada línea que conforma el yml y con esto la configuración del gestor.

- **Línea 1:** En esta nos dice que la versión del archivo compose es el 3.3, lo que indica las instrucciones deben de respetar a esa versión en específico.
- **Línea 3:** Nos indica la configuración respecto a los servicios del contenedor.
- **Línea 4:** Nos da a indicar el nombre del servicio que se está configurando.
- **Línea 5:** Ahí se indica la imagen que se va a utilizar para iniciar el contenedor.
- **Línea 6:** Nos indica los puertos que se usarán para acceder al gestor.
- **Línea 7:** En este caso, el puerto que se va a utilizar es el 3306, este se utiliza usualmente para MySQL.
- **Línea 8:** En este apartado se indican las variables de entorno.
- **Línea 9:** Aquí se especifica la contraseña del root.
- **Línea 10:** Aquí se especifica el nombre de la base de datos que vamos a utilizar.
- **Línea 11:** En este apartado se especifica el nombre del usuario.
- **Línea 12:** En esta parte se escribe la contraseña del usuario.
- **Línea 13:** En este apartado se indican los volúmenes, estableciendo conexiones entre el host y el contenedor.
- **Línea 14:** Aquí se establece la conexión entre la carpeta "config" en la computadora y el archivo "conf.d" en el contenedor, esto es esencial.
- **Línea 15:** Aquí se establece la conexión entre la carpeta "files" del computador y los archivos de Mysql en el contenedor. Esto para que los archivos de la base de datos se puedan guardar afuera del contenedor.
- **Línea 16:** Aquí se hace conexión entre la carpeta "log" en el computador y los archivos de auditoría en el contenedor. Esto para que se visualice los "logs" afuera del contenedor.



# Explicación del proceso

Explicación del proceso para la configuración de auditoría y registros de logs.

Para poder tener acceso a los registros de todo lo que ocurre en la base de datos "**Registros de auditoría**", es necesario implementar ""**volumes**" dentro de la línea `./log:/var/log/mysql`", esto para que el contenido del log de MySQL del contenedor en Linux pueda ser obtenido en la carpeta "**log**" de nuestro computador.

Mas adelante, tienen que activar o "**descomentar**" unas partes del archivo "**my.cnf**" esto para que los logs puedan correr correctamente.

A continuación las líneas que se tienen que activar:

- Línea **53**: `general_log_file = /var/log/mysql/mysql.log`
- Línea **60**: `log_error = /var/log/mysql/error.log`
- Línea **62**: `slow_query_log_file = /var/log/mysql/mariadb-slow.log`
- Línea **72**: `log_bin = /var/log/mysql/mysql-bin.log`

Luego se debe de ubicar el archivo **50-server.cnf** en donde en la parte de configuración se modificara el bin adress a **0.0.0.0** para que se pueda permitir el uso de conexiones externas al contenedor.

Por ultimo se utiliza la aplicación de su preferencia para conectarse al gestor de base de datos.

# Diferencias entre un gestor de base de datos a modo contendor.

## GESTORES DE BASES DE DATOS:

Son aquellos productos/software desarrollados por empresas especializadas, que adaptan el estándar SQL como por ejemplo:

- MySQL
- ServerSQL
- ServerOracle.

Los anteriores, dependiendo de las características que tengan agregadas, son capaces de:

- Realizar transacciones
- Procedimientos almacenados
- Permite la conexión en modo cliente-servidor

## GESTORES DE BASES DE DATOS EN CONTENEDOR

Resulta práctico utilizar las bases de datos y que se ejecuten como contenedores ya que el hecho de que ya que no tiene ninguna dependencia externa y solo con ejecutar el comando **docker-compose** up ya se inicia toda la aplicación. Tener esas bases de datos como contenedores también es muy útil para las pruebas de integración, porque la base de datos se inicia en el contenedor y siempre se rellena con los mismos datos de ejemplo, por lo que las pruebas pueden ser más predecibles. Además de que los respaldos pueden ser automatizados y de muy fácil acceso.

# Instalación del host y servicio en la nube

## Ejemplo de Ubuntu

Instalación de los repositorios:

En primer lugar procederemos a instalar los repositorios así como también las dependencias necesarias

1. Actualizamos la lista de `paquetes`  
`apt-get update`

2. Aseguramos poder establecer conexión con repositorios a través del protocolo HTTPS

`apt-get install apt-transport-https ca-certificates curl software-properties-common.`

El comando anterior instalará las dependencias necesarias para poder añadir y trabajar correctamente con un repositorio a través de una conexión segura.

3. Descargamos y añadimos la clave pública del repositorio de Docker  
`curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`

Esta clave nos permitirá verificar la integridad de los archivos y la autenticidad del emisor, evitando de esta forma cualquier tipo de suplantación de paquetes / archivos. Esta es una medida de seguridad estándar y común.

4. Añadimos el repositorio y actualizamos la lista de paquetes

`add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"`

`apt-get update`

Finalmente, actualizamos la lista de paquetes, habilitando con ello el repositorio.

5. Instalamos Docker mediante el gestor de paquetes

`apt-get install docker-ce`

El nombre de paquete "`docker-ce`" hace referencia a la versión gratuita ("community edition") de Docker.

6. Probamos la instalación lanzando el contenedor "hello-world"

`docker container run hello-world`

# Conclusiones

Actualmente en un mundo repleto de tecnología la cual es muy cambiante de manera constante, Docker es una herramienta increíblemente útil para cualquier desarrollador. Es muy fácil de usar y es una tecnología moderna a la cual nos deberíamos de adaptar. Docker tiene la capacidad de probar, implementar y desarrollar aplicaciones sin problemas es una utilidad que puede acelerar el flujo de trabajo de manera exponencial.