

SEPTIEMBRE 2020



Proceso de Instalación de Base de datos

USO DE CONTENEDORES DOCKER

Prepared by: Jade Bañuelos

OBJETIVO Y ALCANCE

El objetivo de este documento es explicar las diferencias entre los diferentes modos de instalación de base de datos, así como presentar de manera clara el proceso de instalación de una base de datos mediante un contenedor de docker y un gestor de Base de Datos, explicando sus usos y componentes.

El alcance al cual se llego es en la instalación de un gestor de base de datos de MariaDB, mediante un contenedor docker.



PROCESO

ARCHIVO DOCKER-COMPOSE.YML

Version: "3.3"

Los archivos docker-compose.yml son versionados, lo que significa que es muy importante indicar la versión de las instrucciones que queremos darle. En este caso se indica que la versión que se utilizará es la "3.3"

Services:

Se describen los servicios que se ejecutarán en el archivo docker-compose.yml

db:

Es el nombre del servicio que se va a ejecutar, su contenido describe cómo debe iniciar los contenedores para este servicio.

Image:

Se define la imagen a utilizar que en este caso es MariaDB.

Ports:

Se mapean los puertos locales, aquí se define en qué puerto estará la base de datos.

Environment:

Aquí se declaran las variables de entorno.

- MYSQL_ROOT_PASSWORD=123456 // Contraseña del root
- MYSQL_DATABASE=myfirstdb // Nombre de nuestra base de datos
- MYSQL_USER=myfirstuser //Declaración de usuario de nuestra bd
- MYSQL_PASSWORD=myfirstpassword //Contraseña de nuestra bd

Volumes:

Es la dirección de dónde se pondrán los datos y de dónde se sacarán estos mismos datos.

- "./config:/etc/mysql/conf.d"
- "./files:/var/lib/mysql"
- "./log:/var/log/mysql"

véase imagen 1.0

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"
```

Imagen 1.0. archivo docker-compose.yml

PROCESO DE CONFIGURACIÓN DEL GESTOR.

- Descargamos la imagen de mariadb.
- Redactamos el archivo docker-compose.yml (este tendrá los parametros que vimos antes en "Proceso de archivo docker-compos.yml")
- Configuramos el archivo my.cnf para que se configure de manera publica
- Nos ubicamos en la carpeta donde se encuentra nuestro archivo docker compose (lo ejecutamos en una ventana de consola CMD)
- Para verificar que todo esta funcionando correctamente se ejecuta el comando docker-compose up -d
- Finalmente utilizamos una aplicación para conectarnos a nuestra base de datos (HeidiSQL)

para mas información:

https://hub.docker.com/_/mariadb

PROCESO DE CONFIGURACIÓN DE LOS LOGS

Para tener acceso a los registros de la base de datos se debe agregar la siguiente linea a nuestro archivo docker-compose.yml en el apartado volumes:

- "./log:/var/log/mysql"

véase imagen 2.0

```
13      volumes:  
14        - "./config:/etc/mysql/conf.d"  
15        - "./files:/var/lib/mysql"  
16        - "./log:/var/log/mysql"  
17
```

Imagen 2.0. archivo docker-compose.yml
logs

Ademas es necesario des comentar las siguientes líneas en el archivo my.cnf, para que lo logs funcionen de manera correcta:

- 53
- 60
- 62
- 72

DIFERENCIAS EN LOS DISTINTOS PROCESOS DE INSTALACIÓN DE BD

CONTENEDOR

Docker Engine se encarga de lanzar y gestionar los contenedores con nuestras aplicaciones, pero en lugar de exponer los diferentes recursos de hardware de la máquina de manera discreta (1 procesador y "x" GB de RAM... para cada aplicación), lo que hace es compartirlos entre todos los contenedores **optimizando su uso y eliminando la necesidad de tener sistemas operativos separados para conseguir el aislamiento.**

Docker funciona a partir de imágenes que se pueden reutilizar entre varias aplicaciones.

NUBE

Un gestor de base de datos en la nube es una colección de contenido, estructurado o no estructurado,

que se encuentra en una plataforma en la nube privada, pública o híbrida. Estas ofrecen una alta disponibilidad, seguridad mejorada y rendimiento flexible y escalable que ayuda a reducir significativamente los riesgos y costos.

HOST

Un gestor de base de datos con instalación en el host o de manera tradicional es cuando todos los servicios y aplicaciones que se necesitan se instalan en el dispositivo a utilizas y solo se crea una conexión de base de datos, por lo que esto limita en muchos aspectos al momento de la realización de proyectos grande o extensos.



CONCLUSIÓN

Me parece que dentro de todas las maneras o procesos de instalación de bases de datos que hemos visto a lo largo de la carrera, la mejor opción es mediante la nube o contenedores.

Ya que estos dos nos garantizan mayor seguridad y mejor administración de tareas "pesadas", para así enfocarnos en nuestra aplicación, ademas de que es mas practico tener una sola aplicación o servicio a tener 10 en un mismo ordenador.

Creo que es una manera muy sencilla de trabajar y cómoda.

