

TALLERES MIA 1S 2020

- Docker Taller 1
- NodeJS -> JavaScript Taller 2
- Angular -> TypeScript Taller 3

Practicas Intermedias – Voluntariado

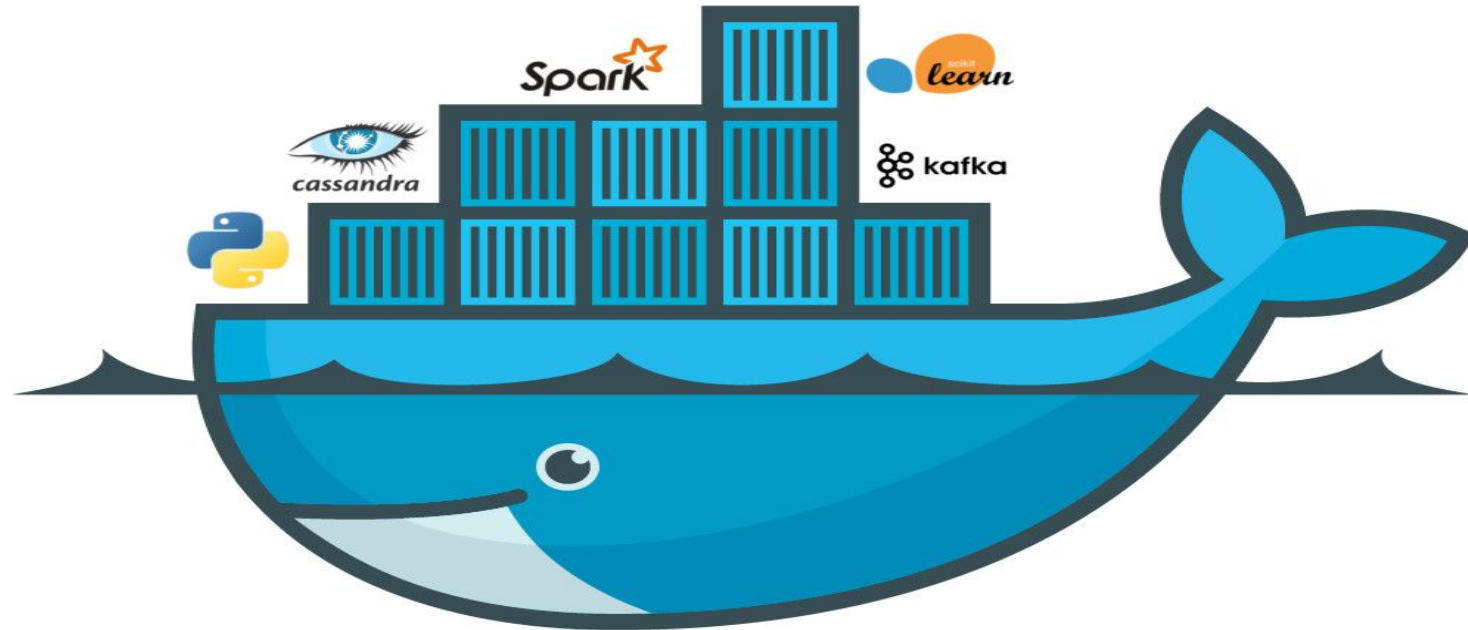


DOCKER

¿Que es?

¿Para que se utiliza?

¿Que beneficios ofrece?



¿Que es Docker?



¿ Que es Docker ?

Básicamente es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de **contenedores** de software.

¿Que es un contenedor?

Definición general de contenedor.

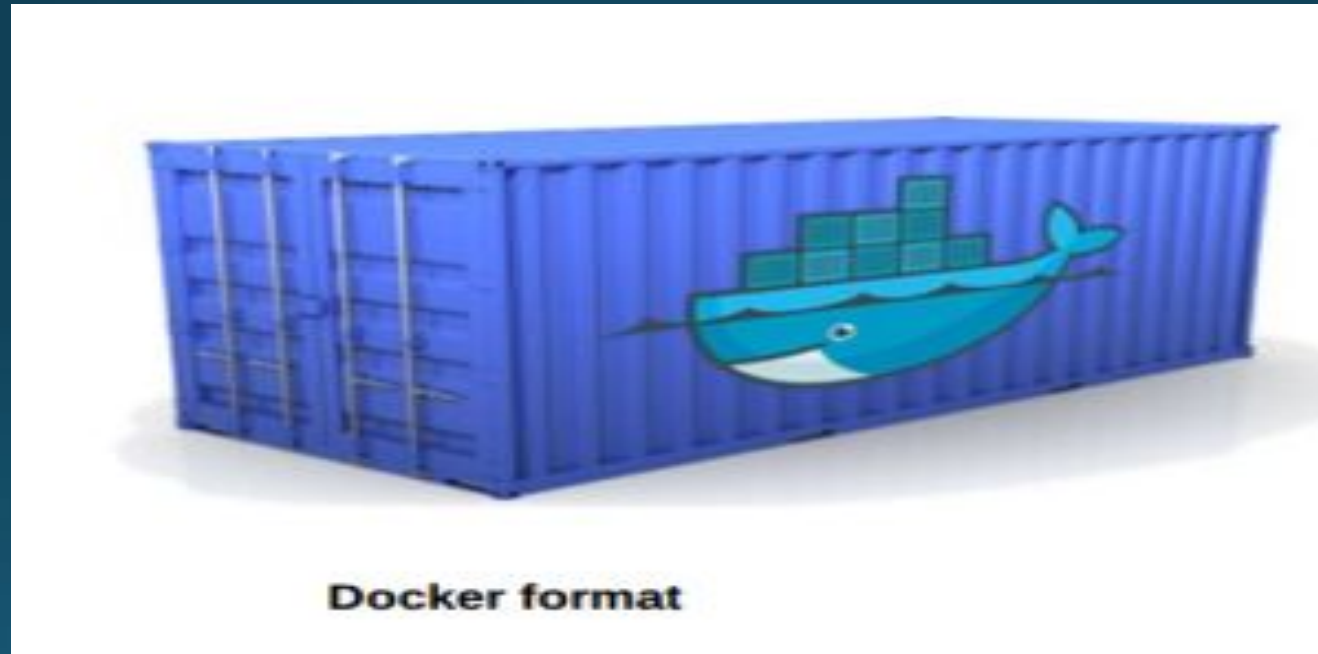
- Un contenedor es un recipiente de carga para el transporte marítimo o fluvial, transporte terrestre y transporte multimodal.
- Unidades estancas que protegen las mercancías de la climatología y que están fabricadas de acuerdo con la normativa ISO.

La idea del uso de Docker

La idea detrás de Docker es crear contenedores ligeros y portables para las aplicaciones de software que puedan ejecutarse en cualquier maquina con Docker instalado.

Su uso es Independiente del sistema operativo que la maquina tenga por debajo, facilitando así también los despliegues.

Es algo auto contenido en si, que se puede llevar de un lado a otro de forma independiente, ya que es portable.



Docker, me permite meter en un contenedor todas aquellas cosas que mi aplicación necesita para ser ejecutada (java, Maven, tomcat) y la propia aplicación.

Así yo me puedo llevar ese contenedor a cualquier máquina que tenga instalado Docker, sin preocuparme si tiene los elementos necesarios para que funcione mi aplicación o de si son compatibles.

Yo ejecutaré mi aplicación de software desde el contenedor de Docker, y dentro de él estarán todas las librerías y cosas que necesita dicha aplicación para funcionar correctamente.

¿Que beneficios da docker?

Docker es una herramienta diseñada tanto para beneficiar a desarrolladores, testers, administradores de sistemas. En relación a las maquinas y los entornos donde se ejecuten las aplicaciones de software.

En el caso de los desarrolladores, el uso de Docker hace que puedan centrarse en desarrollar su código sin preocuparse de si dicho código funcionará en la máquina en la que se ejecutará.

Escenarios donde utilizar docker

- Pepe tiene en su ordenador instalado Java 8, y está programando una funcionalidad específica de la aplicación con algo que solo está disponible en esa versión de Java.
- José tiene instalado en su máquina Java 7, porque está en otro proyecto trabajando sobre otro código, pero Pepe quiere que José ejecute el código de su aplicación en su máquina. O José se instala Java 8, o la aplicación en su máquina fallará.

Solución de escenario

- Este escenario desaparece con Docker. Para ejecutar la aplicación, Pepe se crea un contenedor de Docker con la aplicación, la versión 8 de Java y el resto de recursos necesarios, y se lo pasa a José.
- José, teniendo Docker instalado en su ordenador, puede ejecutar la aplicación a través del contenedor, sin tener que instalar nada más.

Que diferencia a docker con una maquina virtual

En concepto es algo similar, pero un contenedor no es lo mismo que una maquina virtual.

Un contener es mas ligero, ya que mientras una maquina virtual necesita instalarle un sistema operativo para funcionar, un contenedor de Docker funciona utilizando el sistema operativo que tiene la máquina en la que se ejecuta el contenedor.

Digamos que el contenedor de Docker toma los recursos más básicos, que no cambian de un ordenador a otro del sistema operativo de la máquina en la que se ejecuta. Y los aspectos más específicos del sistema que pueden dar más problemas a la hora de llevar el software de un lado a otro, se meten en el interior del contenedor.

Para que te hagas una idea, el concepto de portabilidad de un contenedor de Docker es algo similar a la máquina virtual de Java.

Que un contenedor Docker tome los aspectos básicos de funcionamiento del sistema operativo de la máquina en la que se ejecuta lo vuelve más ligero que una máquina virtual.

Encuestas

En la encuesta de desbordamiento de pila, los desarrolladores clasificaron a Docker.

- n.º 1 en "Plataforma más querida".
- n.º 2 en "Plataforma más buscada".
- n.º 3 en "Plataforma en uso".

Instalación de base de datos Oracle 18c en docker

Primero instalamos docker sobre Ubuntu 18.04

- Actualice su lista de paquetes existente:

```
sudo apt update
```

- A continuación, instalamos algunos paquetes de requisitos previos que le permiten a apt usar paquetes mediante HTTPS:

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

- Luego, agregue la clave GPG para el repositorio oficial de Docker a su sistema:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

- Agregue el repositorio de Docker a las fuentes de APT:

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu  
(lsb_release -cs) stable"
```

- Posteriormente, actualice la base de datos de paquetes usando los paquetes de Docker del repositorio que acaba de agregar:

```
sudo apt update
```

- Por último, instale Docker:

```
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

- Se puede verificar si la instalación de docker se realizó con éxito corriendo el siguiente comando:

```
sudo docker run hello-world
```


Segundo instalación de Oracle 18c sobre Docker en Ubuntu 18.04

- Primero extraemos la imagen del repositorio de docker:

```
sudo docker pull dockerhelp/docker-oracle-ee-18c
```

- Corremos la imagen de Oracle 18c y asignamos un puerto donde se va a comunicar:

```
sudo docker run -it dockerhelp/docker-oracle-ee-18c -p 1521:1521 bash
```

Alternativa

```
sudo docker run -p 8000:8000 -it dockerhelp/docker-oracle-ee-18c bash
```

- Corremos el archivo de instalacion sh

```
sh post_install.sh
```

Se procede a conectarse a la base de datos por medio del siguiente comando, siempre dentro del contenedor.

```
$ sqlplus
```

El usuario que nos crea por defecto es el siguiente.

```
user: sys as sysdba
```

```
pass: oracle
```

Ingresamos y creamos un usuario de prueba

```
SQL> alter session set "_ORACLE_SCRIPT"=true;
```

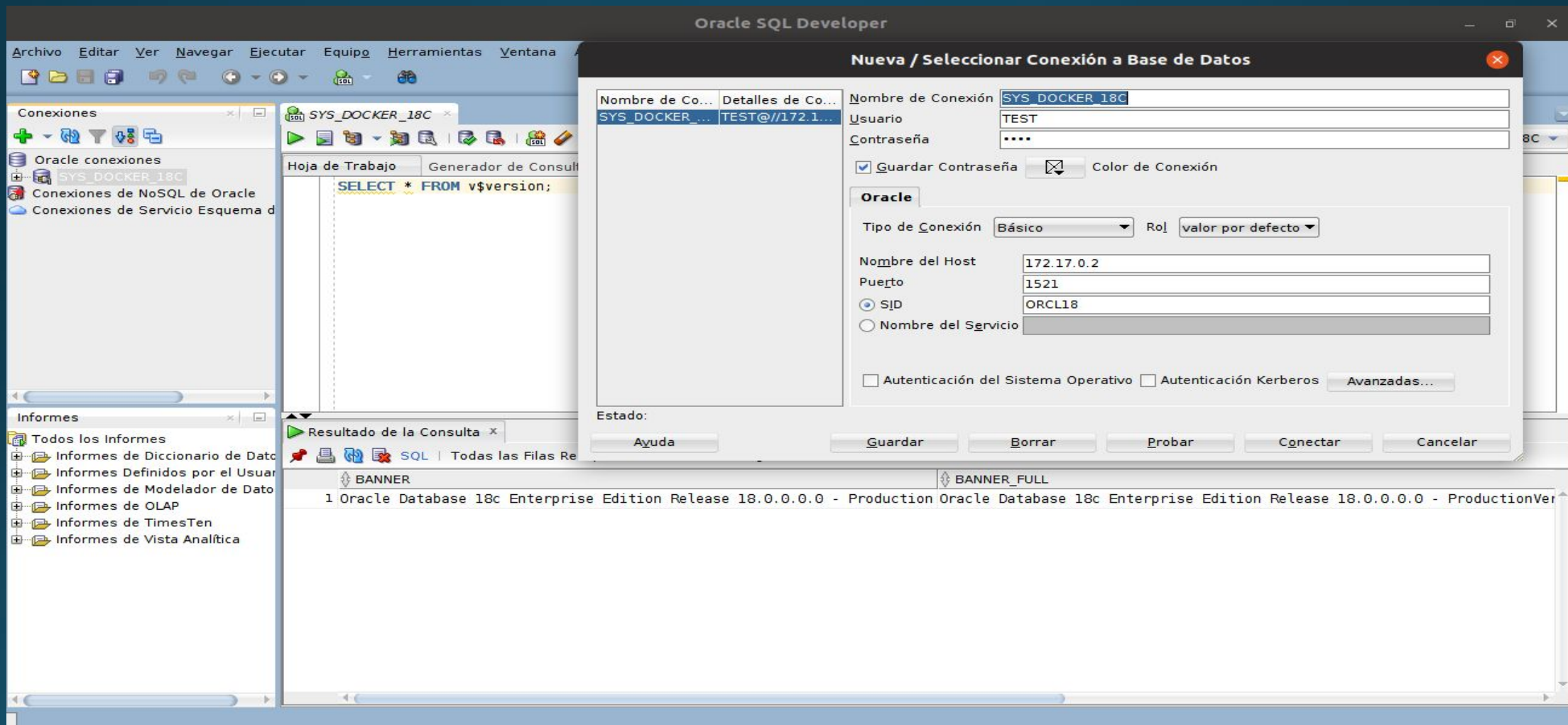
```
Session altered.
```

```
SQL> create user TEST identified by 1234;
```

```
User created.
```

```
SQL> grant dba to TEST;
```

Podemos conectarnos a nuestra base de datos por medio de SQL Developer instalado en el SO físico de la maquina por medio de la ip que nos genero el contenedor.



La IP del container se puede averiguar saliendo del container (sin apagarla) con la combinación de teclas CTRL + P y luego CTRL + Q y ejecutando los siguiente comando.

- Para mostrar todos los contenedores que se encuentran activos. A su vez nos mostrara el id del contenedor.

```
sudo docker ps
```

- El siguiente comando se utiliza para obtener la ip del contenedor

```
sudo docker inspect -f '{{range  
.NetworkSettings.Networks}} {{.IPAddress}} {{end}}' [CONTAINER_ID]
```

En la parte en rojo se sustituye el id generado en el paso anterior.

Iniciar un contenedor ya existente en docker

Se ejecutan los siguientes comandos para iniciar el servidor.

```
sudo docker start 862e461d5819
```

```
sudo docker exec -it 862e461d5819 bash
```

Lo marcado en rojo es el id del contenedor a iniciar.

En el caso de oracle 18c se ejecutan los siguientes comandos para iniciar sesión.

```
sh post_install.sh
```

```
sqlplus
```