

INSTALACION BASE DATOS ORACLE 19.3 EN DOCKER

INSTALACIÓN DE LA IMAGEN DE ORACLE 19C

1º.- Descargar el proyecto: <https://github.com/oracle/docker-images>

2.- Descomprimir la carpeta:

Yo la he descomprimido en /home/rocio/Documentos/dockerOracle

Ir a la carpeta

.../Documentos/dockerOracle/docker-images-main/OracleDatabase/SingleInstance/dockerfiles

3.- Introducir el archivo obtenido de la versión a instalar de la página oficial de oracle (LINUX.X64_193000_db_home.zip) en la carpeta correspondiente, en nuestro caso, la correspondiente a la version 19.3.0 en:

/home/rocio/Documentos/dockerOracle/docker-images-main/OracleDatabase/SingleInstance/dockerfiles/19.3.0

4.- Ejecutar el script "./buildContainerImage.sh" con los parámetros necesarios:

`./buildContainerImage.sh -v 19.3.0 -e`

Nota: Al final sale el mensaje con el nombre de la imagen a ejecutar:

Oracle Database container image for 'ee' version 19.3.0 is ready to be extended:

--> oracle/database:19.3.0-ee

Build completed in 1132 seconds.

5.- Comprobar que se ha creado la imagen de la base de datos oracle:

`~$ docker images`

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
oracle/database	19.3.0-ee	77f1667ab90a	About an hour ago	6.54GB (<i>imagen bd oracle</i>)

CORRER EL CONTENEDOR DE ORACLE

```
$ docker run -d --name oracledb19_3 -p 1521:1521 -p 5500:5500 -e ORACLE_SID=TALLER1SID -e ORACLE_PDB=TALLER1PDB -e ORACLE_PWD=Oracle19 -e ORACLE_EDITION=standard oracle/database:19.3.0-ee
```

```
$ docker logs -f oracledb19_3 --> para que salgan los log del contenedor que estoy creando
```

Explicación de parámetros usados

-name el nombre que le damos al contenedor

-d arranca el contenedor en segundo plano (conocido tambien como background)

-p mapeamos los puertos del host al contenedor

-e hacer referencias a las variables de entorno dentro del contenedor, que en nuestro caso son:

ORACLE_SID: Nombre de la instancia

ORACLE_PDB: Nombre de la Base de datos Plugable

ORACLE_PWD: Clave de los usuarios Administrador SYS y SYSTEM

ORACLE_EDITION= Edición que vamos a instalar:

ANEXO:

Gestionando contenedores:

- Ver contenedores activos:

docker ps o docker container ls

Nota: Observar el campo status

status: (health: starting) --> iniciandose

status: (healthy) --> activo

- ver todos los contenedores parados

docker ps -a

- Iniciar el contenedor una vez parado:

\$ docker start CONTAINER ID

Ver redes que crea docker

\$ docker network ls

\$ docker network inspect bridge --> información sobre las interfaces dentro del docker