

DOCKER en el taller Iris

Nota: desde la consola (Ctrl+Alt+t)

1.- Gestión del “daemon” Docker.

Ver estado, inicialmente estará parado (punto negro):

```
systemctl status docker
```

Arrancarlo

```
systemctl start docker
```

```
Authenticating as: alumno... (alumno)
```

```
Password: "alumno"
```

Ver el estado de nuevo, estará en marcha (punto verde):

```
systemctl status docker
```

2.- Imágenes y contenedores.

Ver contenedores:

```
docker ps
```

```
docker ps -a
```

```
docker ps -a -q
```

Ver imágenes:

```
docker image ls -a
```

```
docker images -a
```

Borrar contenedores (por limpieza):

```
docker rm -f f8306b233fe1
```

```
docker rm -f $(docker ps -a -q)
```

Borrar imágenes (por limpieza):

```
docker rmi nombreImagen
```

```
docker rmi $(docker images -a -q)
```

3.- Docker y PostgreSQL.

Iniciar servicio (daemon).

```
systemctl start docker.service
```

Buscar e instalar

```
docker search postgres
```

```
docker pull postgres
```

Entorno de trabajo.

```
cd Documentos/
```

```
mkdir practica5
```

```
cd practica5/
```

```
mkdir data
```

Arrancar contenedor (es una linea).

```
docker run --name dk_p5 -e POSTGRES_USER=postgres -e
```

```
POSTGRES_PASSWORD=postgres -p 5432:5432 -v
```

```
/home/alumno/Documentos/practica5/data:/var/lib/postgresql/data -d postgres
```

Ahora tenemos un contenedor llamado “dk_p05” en ejecución

Opciones:

- e variables de entorno.
- p mapeo de puertos: host:contenedor
- v volúmenes, mapea un directorio local a un directorio dentro del contenedor: ruta_host:ruta_contenedor
- d contenedor a ejecutarse. Modo desasociado (detached mode): en segundo plano, sin bloquear la terminal.

Ver los contenedores

docker ps -a
docker ps

Ejecutar en el contenedor

docker exec -it dk_p5 bash
Cambia el prompt: root@69793242c851:/#

Opciones:

- i (interactive): mantiene la entrada estándar (stdin) abierta. Permite interactuar con el contenedor.
- t (tty): Asigna un terminal al contenedor. Es útil cuando quieres ejecutar comandos interactivos en el contenedor (como una terminal Bash).
- “bash”: Comando a ejecutar dentro del contenedor.

Otros ejemplos:

Ver paquetes instalados en un contenedor Ubuntu/Debian:
docker exec dk_p5 dpkg --get-selections

4.- Dentro del gestor de bases de datos postgres.

Nota: Tras ejecutar “***docker exec -it dk_p5 bash***” estamos dentro **root@69793242c851:/#**

Entramos en postgres como el usuario “postgres”:

psql -U postgres -p

Opciones:

<i>\l</i>	Listar bases de datos.
<i>\c</i>	Usar la BDs.
<i>\q</i>	Salir. Vuelve al contenedor.
<i>\dt</i>	Listar tablas.
<i>\du</i>	Listar usuarios.
<i>\? o \h</i>	Ayuda:

Crear la BDs.

create database mydb;

Usar una base de datos. Ejemplo:

\c mydb
Password:
You are now connected to database "mydb" as user "postgres".

El prompt cambia a “**mydb-#**”

Ahora los “create table” y los “insert”.

Nota: en futuras ocasiones se puede entrar directamente a la DBs.
“psql -h localhost -U myuser -d mydb”

```
CREATE TABLE cliente (  
    numero SERIAL PRIMARY KEY,  
    nombre VARCHAR(25)  
);  
INSERT INTO cliente (nombre) VALUES  
    ('Juan Pérez'),  
    ('María García'),;  
select * from cliente;
```

5.- Saliendo del contenedor.

Salir y volver al contenedor. Prompt ***“root@69793242c851:/#”***
\q

Salir del contenedor (cuidado):

exit

Ctrl + d

Ctrl + c *Interrumpe la ejecución de un proceso interactivo. Bloqueado

Salir con la secuencia ***“Ctrl + p”*** y ***“Ctrl + q”*** se "desacopla" el terminal del contenedor sin detenerlo. Para volver en este caso:

docker attach dk_p5 (puede perder el proceso interactivo)

docker exec -it dk_p5 sh (mejor)

Guardar los datos del contenedor antes de salir. Con el volumen no sería tan necesario, pero puede ser interesante hacer commit y crear una imagen:

docker commit <nombre_o_id_del_contenedor> <nombre_de_la_imagen>:<tag>

docker commit dk_p5 mi_imagen_postgres:v1

6.- Trucos.

Nota: ejecutar con saltos de línea:

***docker run --name dk_p5 ***

***-e POSTGRES_USER=postgres ***

***-e POSTGRES_PASSWORD=postgres ***

***-p 5432:5432 ***

***-v /home/alumno/Documentos/practica5/data:/var/lib/postgresql/data ***

-d postgres

Y aun mejor, con ***“pwd”*** en el volumen:

***docker run --name dk_p5 ***

***-e POSTGRES_USER=postgres ***

***-e POSTGRES_PASSWORD=postgres ***

***-p 5432:5432 ***

***-v \$(pwd)/data:/var/lib/postgresql/data ***

-d postgres