

INSTALACIÓN POSTGRES UBUNTU DOCKER

Base de Datos
CFGs DAW

Francisco Aldarias Raya
paco.aldarias@ceedcv.es

2019/2020

Fecha 13/03/20

Versión:200313.1114


Licencia



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 Importante

 Atención

 Interesante

código

Revisión:

10/3/2020. Página 4. Conectar como otro usuario al nombre de la base de datos.

ÍNDICE DE CONTENIDO

1.INTRODUCCIÓN.....	3
2.INSTALACIÓN.....	3
3.EJECUTAR SERVIDOR.....	3
4.EJECUTAR CLIENTE.....	3
5.BIBLIOGRAFÍA.....	5

UD04. INSTALACIÓN POSTGRES UBUNTU DOCKER

1. INTRODUCCIÓN

Muchas empresas están reemplazando los caros Oracle's por PostgreSQL's, reduciendo recursos, bajando el costo de licencias, simplificando la infraestructura.

La siguiente instalación es de posgres sobre ubuntu desde terminal.

2. INSTALACIÓN

Secuencia para ponerlo en funcionamiento en un contenedor. Creamos "postgresql" basándonos en la imagen postgres bajada.

```
# docker pull postgres
```

La imagen de PostgreSQL la podemos ver con:

```
$ docker images
```

```
paco@pacocced:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
postgres	latest	dd4fa36a9c2f	9 hours ago	395MB

3. EJECUTAR SERVIDOR

Ejecutamos el servidor:

```
docker run --name postgres1 -e POSTGRES_PASSWORD=testing -d postgres
```

Comprobamos su estado con:

```
$docker stats
```

```
paco@pacocced: ~
```

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
19cd97d56bc7	postgres1	0.02%	3.391MiB / 7.676GiB	0.04%	4.11kB / 0B	4.1kB / 41.6MB	7

4. EJECUTAR CLIENTE

Podemos crear otro contenedor para correr el cliente, lo llamamos "psql", lo linkeamos al

servidor y usamos el modificador “-rm” para que, cuando el contenedor se apague se borre automáticamente.

```
$ docker run -it --rm --name psql --link postgres1:postgres postgres psql -h postgres -U postgres
Password for user postgres:
psql (12.2 (Debian 12.2-2.pgdg100+1))
Type "help" for help.
```

Podemos conectar con el usuario user1 la base de datos bd1 de este modo:

```
$ docker run -it --rm --name psql --link postgres1:postgres postgres psql -h postgres -U user1 bd1
```

Se pueden consultar todos los comandos de psql desde la terminal con la orden:

```
$psql -help
```

Podemos ver las base de datos con: \l

```
postgres=# \l
```

List of databases					
Name	Owner	Encoding	Collate	Ctype	Access privileges
postgres	postgres	UTF8	en_US.utf8	en_US.utf8	
template0	postgres	UTF8	en_US.utf8	en_US.utf8	=c/postgres +
template1	postgres	UTF8	en_US.utf8	en_US.utf8	=c/postgres +
					postgres=CTc/postgres

```
(3 rows)
postgres=# \q
```

PD: Recordar que le hemos dado el password: **testing**

Ahora crea mos una base de datos nueva “test” y una tabla t1.

```
postgres=# create database test;
CREATE DATABASE
```

Nos conectamos con la base de datos creada:

```
postgres=# \c test
You are now connected to database "test" as user "postgres".
```

Creamos la tabla t1

```
test=# create table t1 (id int);
CREATE TABLE
```

Sacamos su contenido:

```
test=# select * from t1;
id
```

```
-----  
(0 rows)
```

Podemos ver los comandos con:

```
postgres=#\?
```

Podemos ver las tablas creadas con:

```
postgres=#\dt
```

5. BIBLIOGRAFÍA

- Corriendo PostgreSQL 10 en Docker.
<https://dockertips.com/postgresql>
- Trucos para Postgres y Docker que te harán la vida más fácil
<https://neodata.ai/trucos-para-postgres-y-docker-que-te-haran-la-vida-mas-facil/>
-