

Creado por:

Isabel Maniega

PostgreSQL: Instalación con Dockers

<https://www.postgresql.org/> (<https://www.postgresql.org/>).

Para esta parte usaremos la imagen oficial de PostgreSQL de Docker HUB:

https://hub.docker.com/_/postgres (https://hub.docker.com/_/postgres).

Para ello descargamos la imagen del repositorio:

- `docker pull postgres`

```
isabel@isabel-SVE1512E1EW:~$ sudo docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
b4d181a07f80: Pull complete
46ca1d02c28c: Pull complete
a756866b5565: Pull complete
36c49e539e90: Pull complete
664019fbcaff: Pull complete
727ae9e9c480: Pull complete
796589e6b223: Pull complete
add1501eead6: Pull complete
fdad1da42790: Pull complete
8c60ea65a035: Pull complete
ccd5df5ee2b1: Pull complete
a3e1e8e2882e: Pull complete
a6032b436e45: Pull complete
Digest: sha256:2b87b5bb55589540f598df6ec5855e5c15dd13628230a689d46492c1d433c4df
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
```

Podemos ejecutar la imagen de dos formas mediante docker o mediante docker-compose:

docker run:

```
docker run -d --name postgresQL -e POSTGRES_PASSWORD=cursoPython -e POSTGRES_USER=IsaMan -p 5432:5432 postgres
```

donde podemos encontrar:

- `-e POSTGRES_PASSWORD=` es el password del usuario para poder conectarnos, es un campo obligatorio.
- `-e POSTGRES_USER=` es el nombre de usuario creado para conectarnos a la base de datos, campo obligatorio.
- `-p 5432:5432 -->` es el puerto al cual conectarnos, por defecto es el 5432.

- postgres es el nombre de la imagen

```
isabel@isabel-SVE1512E1EW:~/Documentos/BigData$ sudo docker run -d --name postgresQL -e POSTGRES_PASSWORD=cursoPython -e POSTGRES_USER=IsaMan -p 5432:5432 postgres
af766296a7f2c0792ce6a4a06b80ae27e1cfb1393ace607b18a7e63ded5ac602
```

Proceso activo:

```
isabel@isabel-SVE1512E1EW:~$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
01cf289faa57	postgres	"docker-entrypoint.s..."	37 seconds ago	Up 37 seconds	0.0.0.0:5432->5432/tcp, :::5432->5432/tcp	postgresQL

docker-compose:

En este caso la imagen indica que podemos ejecutarlo junto a otro servicio que corresponde el adminer, lanzaremos los dos ya que nos servirá luego para poder visualizar los datos:

Crearemos un script con nombre: docker-compose.yml

In []:

```
# Use postgres/example user/password credentials
version: '3.1'
```

services:

```
db:
  image: postgres
  restart: always
  ports:
    - 5432:5432
  environment:
    POSTGRES_PASSWORD: cursoPython
    POSTGRES_USER: IsaMan
```

```
adminer:
  image: adminer
  restart: always
  ports:
    - 8080:8080
```

```
isabel@isabel-SVE1512E1EW:~/atom$ sudo docker-compose up -d
Starting atom_adminer_1 ... done
Starting atom_db_1      ... done
isabel@isabel-SVE1512E1EW:~/atom$
```

```
isabel@isabel-SVE1512E1EW:~/atom$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
473837faeb01	adminer	"entrypoint.sh docke..."	8 minutes ago	Up 36 seconds	0.0.0.0:8080->8080/tcp, :::8080->8080/tcp	atom_adminer_1
9ce68f58dbe4	postgres	"docker-entrypoint.s..."	8 minutes ago	Up 36 seconds	0.0.0.0:5432->5432/tcp, :::5432->5432/tcp	atom_db_1

Creado por:

Isabel Maniega

