

# PostgreSQL

Fuente: [PostgreSQL](#) y [pgAdmin](#)

Al igual que en [MariaDB](#), uno de los elementos importantes de un entorno son las bases de datos (BBDD). Es donde se almacenan los datos necesarios para que las aplicaciones puedan guardar y gestionar persistentemente en el disco.

Habitualmente, cuando creo un Docker de motor de BBDD, añado también un gestor gráfico o tipo web para agilizar la gestión de la misma. Así que en este caso he añadido [pgAdmin](#).

También prefiero centralizarlo todo en un contenedor y que el resto se conecte a este, para así no duplicar las instalaciones de motores de BBDD y ser más conservador en los recursos de memoria, centralizar copias de seguridad...

```
---
services:
  postgres:
    container_name: postgresql
    image: postgres:15
    restart: always
    environment:
      POSTGRES_PASSWORD: ${POSTGRESQL_PWD}
    volumes:
      - ./postgresql:/var/lib/postgresql/data
    networks:
      - postgresqlnet

  pgadmin:
    container_name: pgadmin
    image: dpage/pgadmin4
    restart: always
    environment:
      PGADMIN_DEFAULT_EMAIL: ${PGADMIN_USR}
      PGADMIN_DEFAULT_PASSWORD: ${PGADMIN_PWD}
    ports:
      - 8181:80
```

```
volumes:
  - ./pgadmin:/var/lib/pgadmin

networks:
  - postgresqlnet

depends_on:
  - postgres

networks:
  postgresqlnet:
    external: true
```

## Variables

- `POSTGRESQL_PWD`: Contraseña del usuario administrador de PostgreSQL.
- `PGADMIN_USR`: Usuario de pgAdmin.
- `PGADMIN_PWD`: Contraseña del usuario pgAdmin.

## Notas

- **Importante**: el directorio `./pgadmin` se le han de otorgar permisos para usuario y grupo `5050` (`sudo chown -R 5050:5050 ./pgadmin`), ya que de lo contrario el servicio `pgadmin` no arrancará el servicio web de inicio.
- `image`: Escogeremos una versión concreta como la `15`, `16`, `17`... ya que si le ponemos `latest` (o nada), la imagen se actualizará en algún momento a la última versión y tendremos problemas para levantar las BBDD. Actualmente no hay retro/post/compatibilidad entre versiones. Si quieres pasar de una versión a otra la única opción es hacer una copia de seguridad en una versión y restaurarla en la otra.
- En mi caso particular, tengo varias versiones en marcha, según las necesidades de cada aplicación. Pero procuro tener las menos posibles.
- Otra opción es tener sólo una versión y centralizar todas las aplicaciones ahí y en algún momento, migrar todas las BBDD a otra versión de PostgreSQL, apuntar las aplicaciones al nuevo y eliminar el más antiguo.
- `networks`: Todos los contenedores que creemos a partir de ahora que necesiten conectarse a MariaDB, se han de añadir a la red `postgresqlnet`.

## Copias de seguridad

Hacer copias de seguridad es una obligación si no quieres perder tus datos. En este caso se hace una copia de seguridad de todas las BBDD y objetos. Tengo en mente hacer como en la copia de seguridad de [MariaDB](#) que se haga por separado, pero de momento es esto lo que tengo.

```
#!/bin/bash

# Eliminamos los archivos más antiguos de 20 días
```

```
find -L ./postgresql_bck -mtime +20 -type f -delete
```

```
# Hacemos copia de todas las BBDD y objetos
```

```
docker exec postgresql /usr/bin/pg_dumpall -U postgres >
```

```
./postgresql_bck/postgresql_bck_$(date +%Y%m%d%H%M%S).sql
```

```
##
```

```
# Para restaurar todas las BBDD: cat ./postgresql_bck/postgresql_bck_XXXXXXXXX.sql | docker  
exec -i postgresql psql -U postgres
```

```
# Ojo con restaurarlo todo, ya que también afectaría a los esquemas de sistema y no es válido  
si es de una versión a otra.
```

```
# La restauración habría que hacerla BD por BD si queremos pasar de una a otra versión.
```

```
##
```

Para ejecutarlo hay que recordar hacerlo ejecutable (`chmod u+x backup.sh`) y muy recomendable añadirlo a [cron](#) para que se haga periódicamente.

---

Revision #4

Created 8 November 2024 12:25:09 by Lynze

Updated 28 February 2025 09:57:54 by Lynze