



**Universidad del Valle**

## Despliegue de la base de datos Postgres usando Docker

Juan Esteban Guerrero Camacho – 2040798

Faculty of Engineering, Universidad del Valle

750001 C: SISTEMAS OPERATIVOS

Prof. John Alexander Sanabria Ordoñez

23 de abril del 2024

Estas líneas de código son instrucciones para configurar y utilizar contenedores de Docker para PostgreSQL, así como para realizar operaciones básicas en la base de datos. Aquí tienes una breve descripción de cada una:

1. ``docker volume create pg_db``: Crea un volumen de Docker llamado "pg\_db" que se utilizará para almacenar los datos de PostgreSQL.
2. ``docker network create pg_network``: Crea una red de Docker llamada "pg\_network" que se utilizará para conectar los contenedores.
3. ``docker run --name pg_server --network pg_network -v pg_db:/var/lib/postgresql/data -e POSTGRES_PASSWORD=juan -d postgres:15-bookworm``: Ejecuta un contenedor de PostgreSQL llamado "pg\_server", lo conecta a la red "pg\_network", monta el volumen "pg\_db" en el directorio de datos de PostgreSQL, establece la contraseña del usuario "postgres" como "juan" y utiliza la imagen "postgres:15-bookworm".
4. ``docker exec -it pg_server psql -U postgres``: Se conecta al contenedor de PostgreSQL llamado "pg\_server" y abre el cliente de PostgreSQL.
5. ``CREATE DATABASE tarea_db;``: Crea una base de datos llamada "tarea\_db" dentro del servidor de PostgreSQL.
6. ``\c tarea_db``: Se conecta a la base de datos "tarea\_db".
7. ``CREATE TABLE pg_tabla ...``: Crea una tabla llamada "pg\_tabla" con dos columnas: "id" y "mensaje".
8. ``INSERT INTO pg_tabla ...``: Inserta un registro en la tabla "pg\_tabla" con el mensaje "hola mundo".
9. ``\q``: Sale del cliente de PostgreSQL.
10. ``docker run --name pg_client ...``: Ejecuta un contenedor cliente de PostgreSQL llamado "pg\_client" conectado a la red "pg\_network".
11. ``psql -h pg_server ...``: Se conecta al servidor PostgreSQL desde el cliente.
12. ``SELECT * FROM pg_tabla;``: Realiza una consulta para verificar que el registro se haya insertado correctamente en la tabla "pg\_tabla".
13. ``docker stop pg_server``: Detiene el contenedor del servidor PostgreSQL.
14. ``docker rm pg_server pg_client``: Elimina los contenedores del servidor y del cliente.
15. ``docker volume rm pg_db` y `docker network rm pg_network`: Elimina el volumen y la red creados anteriormente.`

Estas líneas de código forman un flujo básico para configurar y utilizar contenedores de Docker para PostgreSQL, realizar operaciones en la base de datos y limpiar los recursos utilizados una vez completadas las tareas.

### Líneas utilizadas para la realización de la tarea:

```
# Crear un volumen de Docker para almacenar datos de PostgreSQL
docker volume create pg_db
```

```
# Crear una red Docker para conectar los contenedores
docker network create pg_network
```

```
# Ejecutar el contenedor de PostgreSQL y conectarlo a la red pg_network
docker run --name pg_server --network pg_network -v pg_db:/var/lib/postgresql/data -e
POSTGRES_PASSWORD=juan -d postgres:15-bookworm
```

```
# Conectarse al contenedor de PostgreSQL
docker exec -it pg_server psql -U postgres
```

```
# Crear la base de datos llamada 'tarea_db'
CREATE DATABASE tarea_db;
```

```
# Conectarse a la base de datos 'tarea_db'
\c tarea_db
```

```
# Crear la tabla 'pg_tabla' con campos 'id' y 'mensaje'
CREATE TABLE pg_tabla (
  id SERIAL PRIMARY KEY,
  mensaje VARCHAR(255)
);
```

```
# Insertar un registro con el mensaje 'hola mundo'
INSERT INTO pg_tabla (mensaje) VALUES ('hola mundo');
```

```
# Salir de psql
\q
```

```
# Ejecutar el contenedor cliente conectado a pg_network
docker run --name pg_client --network pg_network -it --rm postgres:15-bookworm bash
```

```
# Conectarse a la base de datos 'tarea_db' desde el cliente
psql -h pg_server -U postgres -d tarea_db
```

# Verificar que el registro se haya insertado correctamente

```
SELECT * FROM pg_tabla;
```

# Se ve el registro con el mensaje 'hola mundo'

# Detener el contenedor del servidor PostgreSQL

```
docker stop pg_server
```

# Eliminar los contenedores del servidor y cliente

```
docker rm pg_server pg_client
```

# Eliminar el volumen y la red

```
docker volume rm pg_db
```

```
docker network rm pg_network
```

### **Link Youtube:**

▶ Despliegue de la base de datos Postgres usando Docker - Juan Esteban Guerrero 2020...

<https://www.youtube.com/watch?v=9MLcSRZTUjg>