

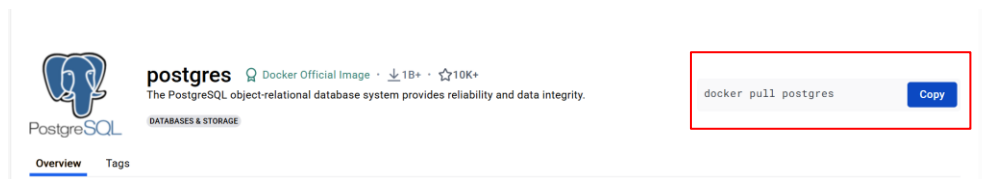
Abastecimiento de infraestructura

Natalia Urrego Rodriguez

1. Descargar imagen en Docker

Se utilizará en este caso la imagen de PostgreSQL donde se puede encontrar en la página de Docker hub.

<https://hub.docker.com/>



Para descargar la imagen se abre PowerShell y ejecuta el comando que esta dentro de un cuadro en la parte de arriba.

```
PS C:\Users\Usuario> docker pull postgres:latest
latest: Pulling from library/postgres
1c47da8f7970: Download complete
01f23f0b9c1b: Download complete
6ea2d101ccee: Download complete
8e0e9cc584ce: Download complete
04f0cbfc26b6: Download complete
6d821824937c: Download complete
565e7451acf: Download complete
f69cbdd712da: Download complete
8d0b32ee6c43: Download complete
f62596e0b037: Download complete
c29f5b76f736: Download complete
63b9b48ec2a9: Download complete
edeaa13d45b54: Download complete
84047ddafe46: Download complete
Digest: sha256:6e3358e46e34dae6c184f48fd06fe1b3dbf958ad5b83480031907e52b9ec2a7d
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
```

latest es una etiqueta que sirve para que descargue la última versión de la imagen, esto lo puedes hacer en cualquier imagen que desee descargar.

Para validar que la imagen se descargó sin ningún problema se pone el comando “Docker images”.

```
PS C:\Users\Usuario> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
postgres      latest    6e3358e46e34   2 days ago    690MB
```

2. Crear un contenedor

Para crear el contenedor en el caso de postgres es con el comando “docker run --name **Nombre del conenedor** -e POSTGRES_PASSWORD=**contraseña** -d -p 5432:5432 postgres:latest”. Se creo con los puertos 5432:5432 para indicar que estará en el localhost

Para validar si el contenedor está corriendo correctamente y verificar los puertos se puede poner el comando “docker ps” donde se mostrará

```
PS C:\Users\Usuario> docker run --name postgres-tadb-examen01 -e POSTGRES_PASSWORD=Examen01 -d -p 5432:5432 postgres:latest
d8bf8828fc3862aac1d7b6d77534523f05314744607661b1adba54f0a2e1b51
PS C:\Users\Usuario> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
d8bf8828fc3   postgres:late "docker-entrypoint.s..." 36 seconds ago Up 34 seconds  0.0.0.0:5432->5432/tcp postgres-tadb-examen01
```

También se puede validar en el Docker en la parte de contenedores que ya está creada y corriendo.

<input type="checkbox"/>	● postgres-tadb-examen01	d8bf8828fc3	postgres:la	5432:5432	0%			
--------------------------	---	-------------	-------------	-----------	----	--	--	--

3. Creación de la base de datos

- Se ingresa al contenedor que se creó y en la parte de exec se pone el comando “psql -U postgres” para que se pueda conectar.
- Adentro se crea la base de datos con el comando “create database examen01_db;”
- Para conectarse a ella es con el comando “\c examen01_db;”
- Se crea un usuario para tener privilegio para modificar o almacenar datos en la base de datos con el comando “create user examen01_usr with encrypted password 'Examen01';”
- Darle privilegios:
Se crea los privilegios con base a lo que se necesita en la base de datos, en este caso se dará privilegios a: establecer conexiones, crear tablas temporales, uso en el esquema, crear objetos, tablas existentes, secuencias existentes, funciones existentes, procedimientos existentes, futuras tablas y secuencias, futuras funciones y procedimientos, consulta sobre el esquema información.

4. Creación de conexión con la base de datos

Se tiene que tomar en cuenta cual fue el usuario, contraseña y los puertos del contenedor que se creó para ser efectiva la conexión.

Se abre primero Dbeaver en este caso y le tiene que colocar el nombre de la base de datos creada con el usuario y contraseña.

