

Reporte Instalación de Docker y del SMBD, PostgreSQL

Sistema operativo y versión

Linux kali 6.10.9-amd64

Distribución

Kali

Instalación de Docker

Se instalo la versión 26.1.5+dfsg1, build a72d7cd

Para instalar docker seguí los pasos enunciados en la documentación oficial de Kali. Estos lo puede encontrar en la siguiente liga:

<https://www.kali.org/docs/containers/installing-docker-on-kali/>

1. Al inicio seguí los pasos propuestos por el ayudante, por lo que actualice los paquetes y dependencias de mi equipo. Esto lo hice mediante el siguiente comando. Este no tardo tanto tiempo. Yo estimaría a lo máximo dos minutos.

```
(loren@kali)~[~]  
$ sudo apt update  
Get:2 https://packages.microsoft.com/repos/code stable InRelease [3590 B]  
Get:3 http://download.virtualbox.org/virtualbox/debian noble InRelease [4428 B]  
Get:4 https://dl.google.com/linux/chrome/deb stable InRelease [1825 B]  
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]  
Get:5 https://download.virtualbox.org/virtualbox/debian bullseye InRelease [7738 B]  
Get:6 https://packages.microsoft.com/repos/code stable/main armhf Packages [16.8 kB]  
Get:7 https://packages.microsoft.com/repos/code stable/main amd64 Packages [16.6 kB]  
Get:8 https://packages.microsoft.com/repos/code stable/main arm64 Packages [16.7 kB]  
Get:9 http://download.virtualbox.org/virtualbox/debian noble/contrib amd64 Packages [1544 B]  
Get:10 https://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1208 B]  
Get:11 http://download.virtualbox.org/virtualbox/debian noble/contrib amd64 Contents (deb) [3980 B]  
Get:12 http://kali.download/kali kali-rolling/main amd64 Packages [20.2 MB]  
Get:13 https://download.virtualbox.org/virtualbox/debian bullseye/contrib amd64 Packages [1933 B]
```

2. Para instalar docker en kali, según la documentación, basta con usar el siguiente comando.

NOTA: Kali tiene otra paquete con el nombre de docker, el cual es distinto a lo que queremos. Por lo que el paquete a instalar tiene el nombre de docker.io.

```
(loren@kali)-[~]
$ sudo apt install -y docker.io
The following packages were automatically installed and are no longer required:
aspnetcore-runtime-6.0      libwinpr2-2t64
aspnetcore-targeting-pack-6.0  libxcb-damage0
dnsmap                      libxerces-c3.2t64
dotnet-apphost-pack-6.0      medusa
dotnet-host                 netstandard-targeting-pack-2.1
dotnet-hostfxr-6.0          proj-bin
dotnet-runtime-6.0          proj-data
dotnet-runtime-deps-6.0      python-odf-doc
dotnet-sdk-6.0              python-odf-tools
dotnet-targeting-pack-6.0    python-tables-data
figlet                      python3-ajpy
finger                     python3-altgraph
firebird3.0-common          python3-bottleneck
firebird3.0-common-doc      python3-docopt
freedp2-x11                 python3-donut
gdal-data                   python3-dropbox
gdal-plugins                python3-git
```

3. Se inicia docker, tal que no necesite hacer `sudo systemctl start docker` cada vez que quiera usarlo.

```
(loren@kali)-[~]
$ sudo systemctl enable docker --now
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
```

4. Finalmente probé que mi instalación de docker funcionara mediante el comando `docker run hello world`. A continuación el resultado del comando.

```
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Todo parece funcionar correctamente.

Luego me di cuenta que en la guía dada por el ayudante estaban descargando `docker-ce`. Por suerte, la documentación igual incluye pasos para descargarlo.

1. `Docker-ce` esta disponible desde el repositorio de docker, el cual no tiene soporte directo de kali pero si de Debian. Kali esta basada en Debian, por lo cual se puede instalar y ejecutar paquetes de debian sin mayor complicación.

```
(loren@kali)-[~]
└─$ echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/debian bookworm stable" | \
    sudo tee /etc/apt/sources.list.d/docker.list
[sudo] password for loren:
deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/debian bookworm stable
```

2. Con el siguiente comando importe la llave gpg.

```
(loren@kali)-[~]
└─$ curl -fsSL https://download.docker.com/linux/debian/gpg | \
    sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

3. Finalmente instale docker-ce.

```
(loren@kali)-[~]
└─$ sudo apt install -y docker-ce docker-ce-cli containerd.io
```

Instalación del SMBD PostgreSQL en contenedor de docker

Se instalo la versión 17.2 (Debian 17.2-1.pgdg120+1)

Para instalar PostgreSQL seguí los pasos dados por el ayudante.

1. Primero instale la imagen de postgresQL para docker. Este proceso fue bastante rápido.

```
(loren@kali)-[~]
└─$ sudo docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
c29f5b76f736: Pull complete
f0a9d8a20b55: Pull complete
9a21b165f71a: Pull complete
3fc477a3cd04: Pull complete
cfafd8bb3e72: Pull complete
c6d3cd6e7746: Pull complete
eb5275734f43: Pull complete
0307008f950a: Pull complete
a0d2dc1de520: Pull complete
32d6a46b2b6d: Pull complete
e12abaa312b3: Pull complete
a97bf08b3792: Pull complete
17a0070c8a8a: Pull complete
375b183b135f: Pull complete
Digest: sha256:3267c505060a0052e5aa6e5175a7b41ab6b04da2f8c4540fc6e98a37210aa2d3
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
```

2. Siguiendo las instrucciones del pdf cree el contenedor de postgres.

```
(loren@kali)-[~]
└─$ sudo docker run -d --name postgres -e POSTGRES_PASSWORD=319331622FDB -p 5432:5432 postgres
b6e2a657c49ba3787a306e4e78518344b200114f0e10e2c791fb656faedbd4c7
```

3. Finalmente inicie postgres y pude observar que aparecía la terminal de este.

```
(loren@kali)-[~]  
$ sudo docker run -it --link postgres:postgres postgres psql -h postgres -U po  
stgres  
Password for user postgres:  
psql (17.2 (Debian 17.2-1.pgdg120+1))  
Type "help" for help.  
  
postgres=#
```

Conexión a DBeaver

Se instalo la versión DBeaver community versión 24.3.4.202502021521

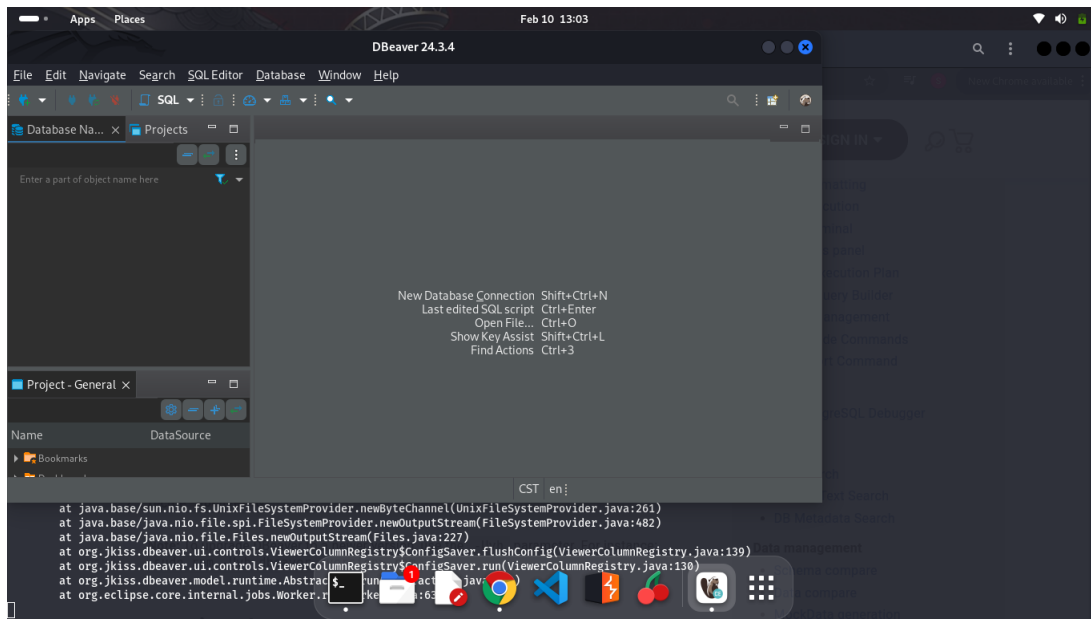
Para instalar DBeaver seguí lo indicado en la documentación de Kali: <https://www.kali.org/tools/dbeaver/> Esto consiste de un único paso.

```
$ sudo apt install dbeaver  
[sudo] password for loren:  
The following packages were automatically installed and are no longer required:  
aspnetcore-runtime-6.0 libwinpr2-2t64  
aspnetcore-targeting-pack-6.0 libxcb-damage0  
dnsmap libxerces-c3.2t64  
dotnet-apphost-pack-6.0 medusa  
dotnet-host netstandard-targeting-pack-2.1
```

Este paso fue bastante sencillo y una vez terminado probé que la instalación fuera correcta. Para esto ejecute el siguiente comando:

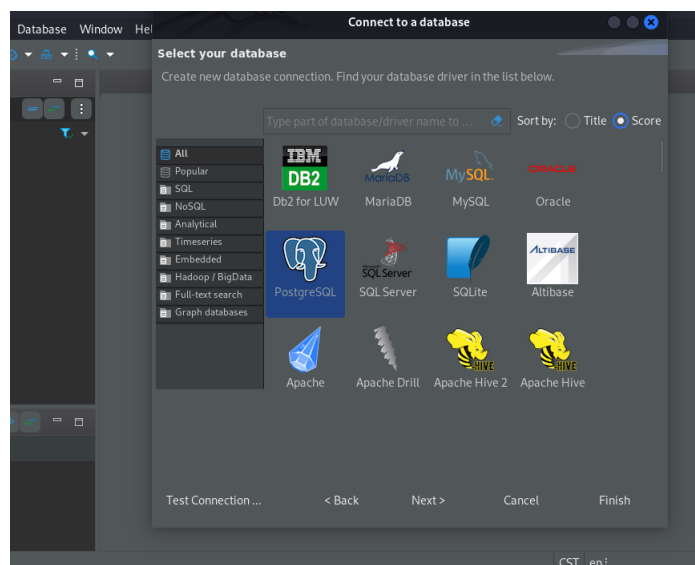
```
(loren@kali)-[~]  
$ dbeaver 5  
[1] 16180  
  
(loren@kali)-[~]  
$ Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true  
> Start Eclipse Jobs Mechanism [org.eclipse.core.jobs 3.15.400.v20240619-0602]  
> Start Equinox Java Authentication and Authorization Service (JAAS) [org.eclipse.equinox.security 1.4.400.v20240702-1702]  
> Start DBeaver Application Standalone [org.jkiss.dbeaver.ui.app.standalone 24.3.4.202502021521]  
> Start Eclipse IDE UI Application [org.eclipse.ui.ide.application 1.5.600.v20241021-1257]  
> Start Eclipse IDE UI [org.eclipse.ui.ide 3.22.400.v20241031-2136]  
> Start DBeaver Model Registry [org.jkiss.dbeaver.registry 1.0.142.202502021521]  
> Start jna [com.sun.jna 5.15.0.v20240915-2000]  
> Start DBeaver UI [org.jkiss.dbeaver.ui 5.1.163.202502021521]  
> Start Expression Language [org.eclipse.core.expressions 3.9.400.v20240413-1529]  
> Start DBeaver Usage Statistics [org.jkiss.dbeaver.ui.statistics 1.0.44.202502021521]  
> Start DBeaver Desktop Application Core [org.jkiss.dbeaver.core 24.3.4.202502021521]  
2025-02-10 13:03:19.900 - DBeaver 24.3.4.202502021521 is starting  
2025-02-10 13:03:19.901 - OS: Linux 6.10.9-amd64 (amd64)  
2025-02-10 13:03:19.901 - Java version: 21.0.3 by Oracle Corporation (64bit)  
2025-02-10 13:03:19.902 - Install path: '/usr/lib/dbeaver'  
2025-02-10 13:03:19.902 - Instance path: 'file:/home/loren/.local/share/DBeaverData/workspace6/'  
2025-02-10 13:03:19.906 - Memory available 117Mb/1024Mb  
2025-02-10 13:03:19.906 - Platform not initialized: metadata folder may be not set
```

El cual me dio como resultado la siguiente interfaz de usuario:



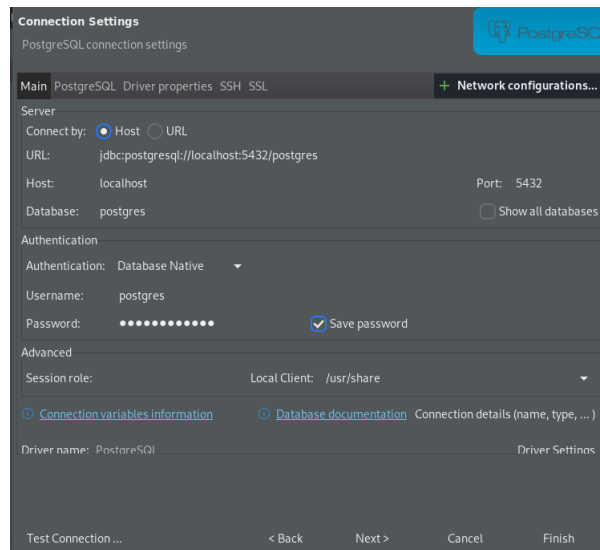
Ahora para la conexión seguí los pasos sugeridos en la documentación de DBeaver: <https://dbeaver.com/docs/dbeaver/Database-driver-PostgreSQL/>

1. Una vez abierta la interfaz, hice click en el menú dropdown con el nombre de databases. Ahí seleccione la opción de "New connection". Esta abrió la ventana que aparece a continuación.

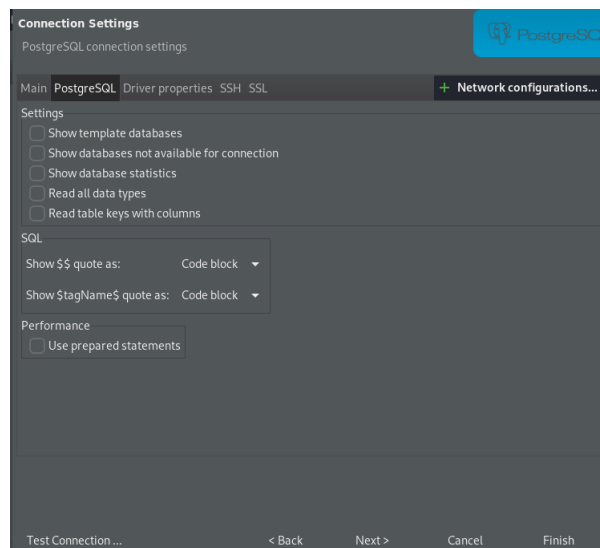


En la documentación de DBeaver se mencionan 3 diferentes versiones de postgresSQL. En mi caso, al tener una de las versiones más modernas, seleccione la versión default de postgresSQL.

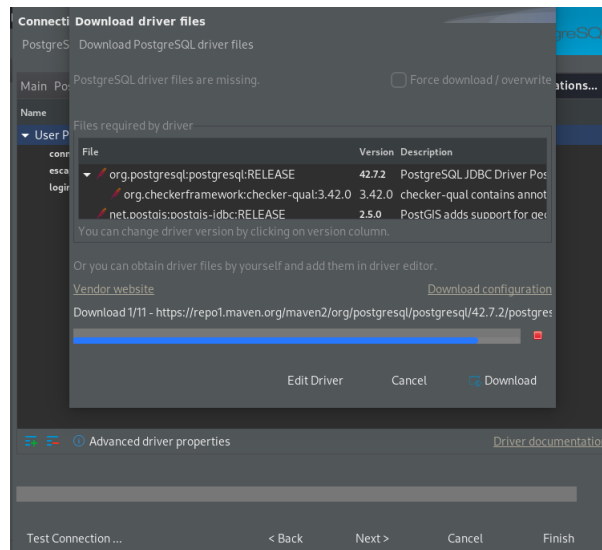
2. Esto me llevó a la siguiente ventana. En este caso lo deje como estaba por default, unicamente añadiendo la contraseña de mi usuario de postgres.



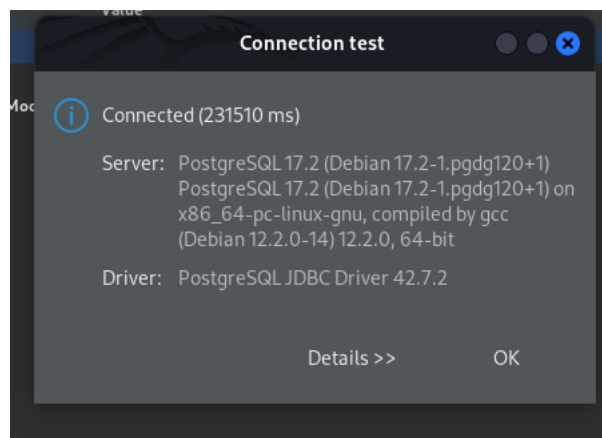
3. En el siguiente apartado se incluyen más configuraciones. Una vez, más lo deje como estaba pasándome al siguiente apartado.



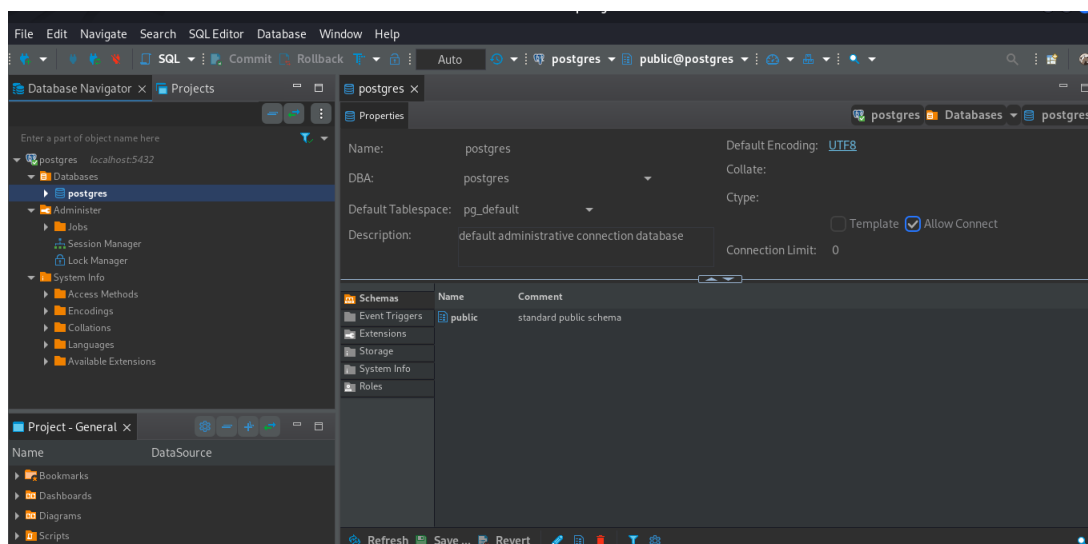
4. Al ser la primera vez que se realiza la conexión, se instalaron automáticamente los drivers necesarios. No hice mucho en esta pestaña mas que esperar a que se terminaran de descargar.



5. Finalmente apareció en pantalla que la conexión fue un éxito, mostrando mi versión de postgresSQL.



Así fue como quedó la pantalla al finalizar la conexión.



Tiempo requerido Comentarios y problemas de la instalación

En total creo que me tarde unos 30 minutos, siendo lo más tardado el descargar todo lo necesario.

En particular no tuve algún problema al realizar las diversas instalaciones. La documentación de los paquetes fue bastante clara y concisa, de igual manera es importante recalcar que uno de los principales motivos por el cual no tuve mayor dificultad es gracias a que todo fue compatible con mi sistema operativo.

No tengo algún comentario respecto a la actividad.