

Bitácora: Gallegos Diego Cristian Ricardo

1 Sistema operativo y versión.

Lo instale en dos sistemas operativos:

- Windows 10
- Linux Fedora 38

2 Distribución (solamente en el caso de Linux).

Fedora 38

3 Versión de la instalación.

Tanto para Windows como para Fedora, al ejecutar `docker inspect postgres`, me dio o arrojaron que tengo la versión 15.4-1 en ambos sistemas; me aparece algo como:

```
"Env": [ "POSTGRES_PASSWORD=318114723",  
"PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/lib/postgresql/15/bin",  
"GOSU_VERSION=1.16",  
"LANG=en_US.utf8",  
"PG_MAJOR=15",  
"PG_VERSION=15.4-1.pgdg120+1",  
"PGDATA=/var/lib/postgresql/data"]
```

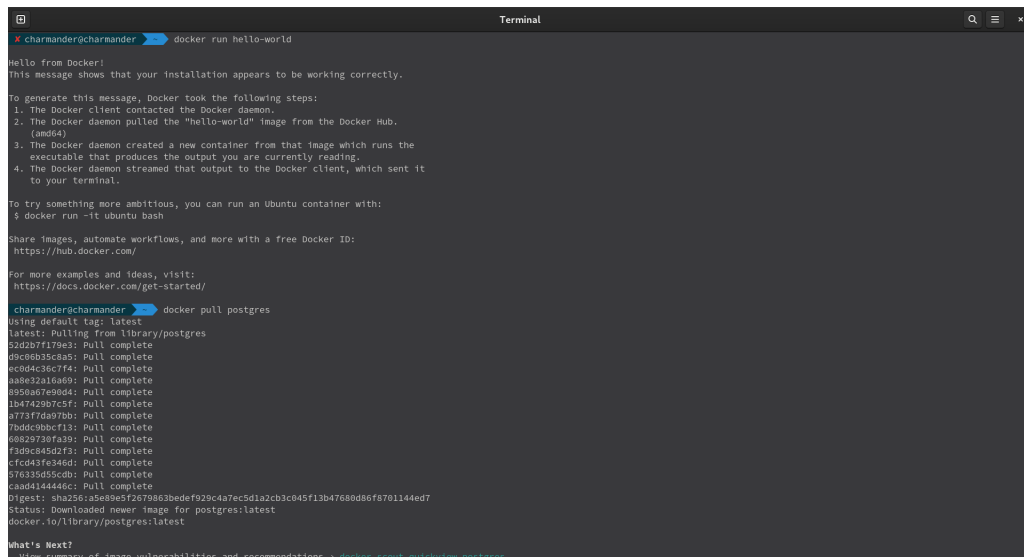
4 Tiempo requerido.

En Windows ya tenía Docker instalado y configurado con WSL, solo me tarde 5 minutos en instalar DBeaver. Para Fedora me tarde 1 - 2 horas, en cambiar los permisos, y agregar unos scripts o directorios que te piden en la página oficial de Docker, al igual que también descargue Docker Desktop, finalmente fue fácil instalar DBeaver.

5 Explicación del paso a paso que realizaste con sus respectivas capturas de pantallas. (Adicionalmente agrega las evidencias de los pasos que consideres esenciales de la instalación).

Fedora:

En la siguiente imagen se puede observar como ya quedo instalado Docker, y como llevamos a cabo la instalación de PostgreSQL



```
charmander@charmander ~$ docker run hello-world  
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/  
  
charmander@charmander ~$ docker pull postgres  
Using default tag: latest  
latest: pulling from library/postgres  
52d2b7f79e3: Pull complete  
d9c8b35c8a5: Pull complete  
ec8d4c36c7fa: Pull complete  
a88e32a1a6a9: Pull complete  
8950a67e90d4: Pull complete  
1b47429b7c5f: Pull complete  
b723fda97db: Pull complete  
7baddc9bdcf13: Pull complete  
66829730fa39: Pull complete  
73d9c845d2f3: Pull complete  
efcd43fe346d: Pull complete  
576335d55cdb: Pull complete  
caad4144446c: Pull complete  
Digest: sha256:a5e89e5f26788e2b0def929c4a7ecd1a2cb3c045f13b47680d86f9701144ed7  
Status: Downloaded newer image for postgres:latest  
docker.io/library/postgres:latest  
  
What's Next?  
View summary of image vulnerabilities and recommendations + docker scout quickview postgres
```

Despues ejecutamos el comando: `docker run -d --name postgres -e POSTGRES_PASSWORD=318114723 -p 5432:5432 postgres`
Donde 318114723 es la contraseña que yo escogi.

```

charmander@charmander:~$ docker --version
Docker version 24.0.5, build ced9996
charmander@charmander:~$ docker run -d --name postgres -e POSTGRES_PASSWORD=318114723 -p 5432:5432 postgres
44ce8469eb5233e5f1bcbd8ca11fd75946ac9e59d12d454f6253f432b4f6b7f4
charmander@charmander:~$

```

Después listamos todos los contenedores aunque no estén activos, y lo paramos, es decir hicimos un docker stop.

```

charmander@charmander ➤ docker --version
Docker version 24.0.5, build ced0996

charmander@charmander ➤ docker run -d --name postgres -e POSTGRES_PASSWORD=318114723 -p 5432:5432 postgres
44ce8469eb5233e5f1bcbdsca1f47594eac9e59d12d454f6253f432b4f6b7f4

charmander@charmander ➤ docker ps -a

```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS | NAMES |
|--------------|-------------|--------------------------|--------------------|---------------------------|---|-----------------|
| 44ce8469eb52 | postgres | "docker-entrypoint.s..." | About a minute ago | Up About a minute | 0.0.0.0:5432->5432/tcp, :::5432->5432/tcp | postgres |
| 2c33d6523d92 | hello-world | "/hello" | 26 minutes ago | Exited (0) 26 minutes ago | | elated_herschel |

```

charmander@charmander ➤ docker start 44ce8469eb52
44ce8469eb52

charmander@charmander ➤

```

Notemos que nos dice el pdf que paremos el contenedor, sin embargo hay que volverlo activar para se respete el comando: `docker run -it --link postgres:postgres postgres psql -h postgres -U postgres` y salga la consola de comandos de PostgreSQL.

```

charmander@charmander ~$ docker --version
Docker version 24.0.5, build ced9996

charmander@charmander ~$ docker run -d --name postgres -e POSTGRES_PASSWORD=318114723 -p 5432:5432 postgres
44ce8469eb52385f1bcbddca71f75946ace59d12d454f62d3f432b4f6b7f4

charmander@charmander ~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
44ce8469eb52   postgres  "docker-entrypoint.s..." About a minute ago Up About a minute    0.0.0.0:5432->5432/tcp, :::5432->5432/tcp    postgres
2c33d652ad92   hello-world  "/hello"                26 minutes ago Exited (0) 26 minutes ago                                elated_herschel

charmander@charmander ~$ docker start 44ce8469eb52
44ce8469eb52

charmander@charmander ~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
44ce8469eb52   postgres  "docker-entrypoint.s..." About a minute ago Up About a minute    0.0.0.0:5432->5432/tcp, :::5432->5432/tcp    postgres
2c33d652ad92   hello-world  "/hello"                27 minutes ago Exited (0) 27 minutes ago                                elated_herschel

charmander@charmander ~$ docker stop 44ce8469eb52
44ce8469eb52

charmander@charmander ~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
44ce8469eb52   postgres  "docker-entrypoint.s..." 2 minutes ago   Exited (0) 4 seconds ago    0.0.0.0:5432->5432/tcp, :::5432->5432/tcp    postgres
2c33d652ad92   hello-world  "/hello"                27 minutes ago Exited (0) 27 minutes ago                                elated_herschel

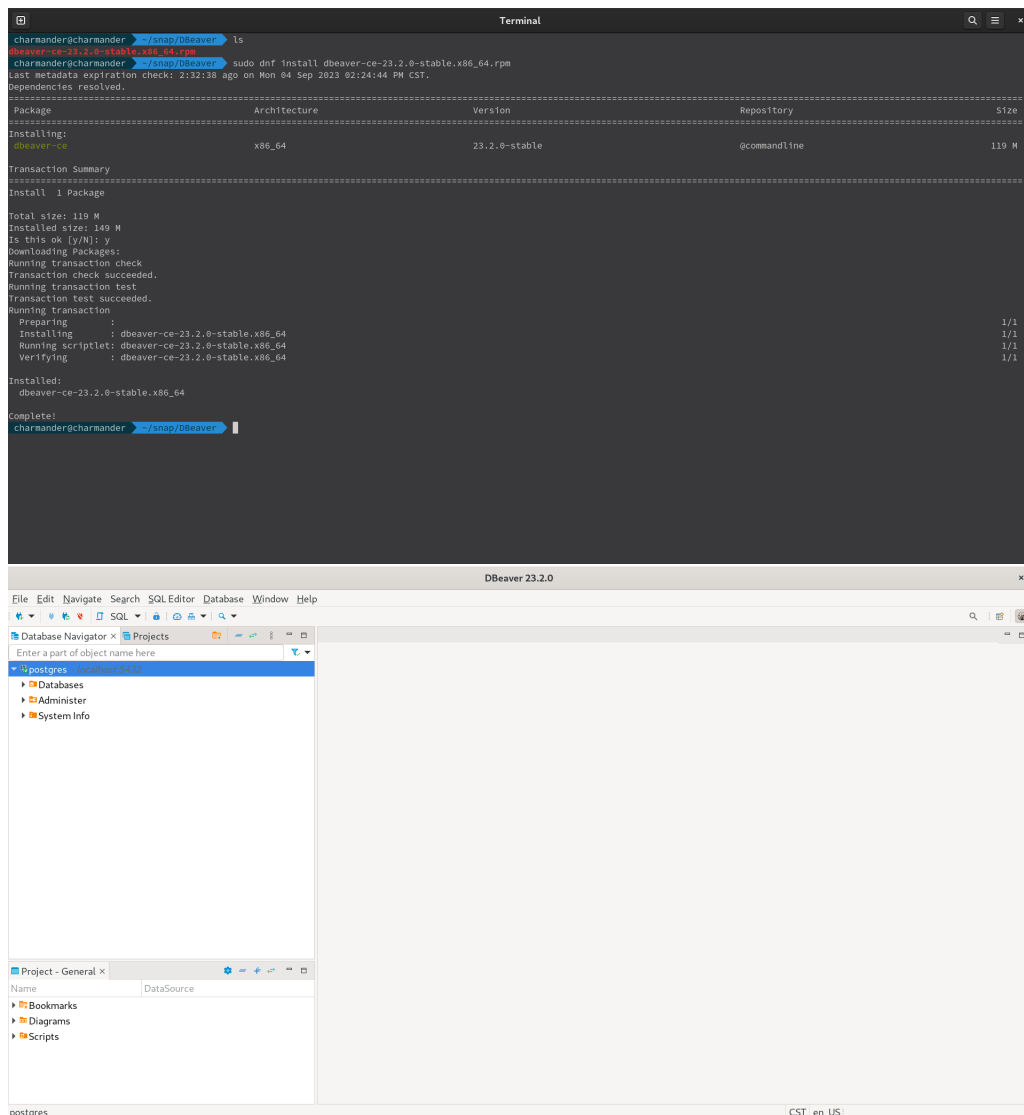
charmander@charmander ~$ docker run -it --link postgres:postgres postgres psql -h postgres -U postgres
docker: Error response from daemon: Cannot link to a non running container: /postgres AS /upbeat_nash/postgres.
[0001] error waiting for container:
charmander@charmander ~$ docker start 44ce8469eb52
44ce8469eb52

charmander@charmander ~$ docker run -it --link postgres:postgres postgres psql -h postgres -U postgres
Password for user postgres:
psql (15.4 (Debian 15.4-1.pgd128+1))
Type "help" for help.

postgres=#

```

Recuerdo que en la siguientes imágenes lleve la instalación de DBeaver, pero para eso le di en start al contenedor de PosgreSQL, y cuando abri DBeaver, me pidió la contraseña que hemos ingresado antes, la cual es 318114723. Y finalmente, asi lleve a cabo la instalación correcta de Docker y DBeaver en Fedora.



Windows:

Para Windows, creo que esta de más explicar lo mismo, además ya lo tenía instalado, a excepción de DBeaver, por lo cual solo agregare unas capturas de pantalla que pienso que son fundamentales. Aquí al igual que en Fedora, me pidió la contraseña, pero fue exactamente lo mismo.

```
MINGW64/C:/Users/cgali
$ signalR.MPTOP-KEE52EAG MINGW64 ~
$ docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
2c2b2f17f9e3: Pulling fs layer
d9c0b35c8a5: Pulling fs layer
ec0d436c7f4: Pulling fs layer
aa8e32a1da09: Pulling fs layer
8950ab7e90d4: Pulling fs layer
1b4742b0c5f: Pulling fs layer
6a73f7da97bb: Pulling fs layer
7bdc9bbcf13: Pulling fs layer
60829730fa39: Pulling fs layer
f3b9c845d2f3: Pulling fs layer
cfcd43fe346d: Pulling fs layer
576335d53cde: Pulling fs layer
caad4144446c: Pulling fs layer
aa8e32a1da09: Waiting
8950ab7e90d4: Waiting
1b4742b0c5f: Waiting
6a73f7da97bb: Waiting
7bdc9bbcf13: Waiting
60829730fa39: Waiting
f3b9c845d2f3: Waiting
cfcd43fe346d: Waiting
576335d53cde: Waiting
caad4144446c: Waiting
d9c0b35c8a5: Verifying Checksum
d9c0b35c8a5: Download complete
aa8e32a1da09: Verifying Checksum
aa8e32a1da09: Download complete
ec0d436c7f4: Verifying Checksum
ec0d436c7f4: Download complete
8950ab7e90d4: Verifying Checksum
8950ab7e90d4: Download complete
1b4742b0c5f: Verifying Checksum
1b4742b0c5f: Download complete
6a73f7da97bb: Verifying Checksum
6a73f7da97bb: Download complete
7bdc9bbcf13: Verifying Checksum
7bdc9bbcf13: Download complete
f3b9c845d2f3: Verifying Checksum
f3b9c845d2f3: Download complete
cfcd43fe346d: Verifying Checksum
cfcd43fe346d: Download complete
576335d53cde: Verifying Checksum
576335d53cde: Download complete
caad4144446c: Verifying Checksum
caad4144446c: Download complete
2c2b2f17f9e3: Verifying Checksum
2c2b2f17f9e3: Download complete
2c2b2f17f9e3: Pull complete
d9c0b35c8a5: Pull complete
ec0d436c7f4: Pull complete
aa8e32a1da09: Pull complete
8950ab7e90d4: Pull complete
1b4742b0c5f: Pull complete
6a73f7da97bb: Pull complete
8950ab7e90d4: Waiting
1b4742b0c5f: Waiting
6a73f7da97bb: Waiting
7bdc9bbcf13: Waiting
f3b9c845d2f3: Waiting
cfcd43fe346d: Waiting
576335d53cde: Waiting
caad4144446c: Waiting
d9c0b35c8a5: Verifying Checksum
d9c0b35c8a5: Download complete
aa8e32a1da09: Verifying Checksum
aa8e32a1da09: Download complete
ec0d436c7f4: Verifying Checksum
ec0d436c7f4: Download complete
8950ab7e90d4: Verifying Checksum
8950ab7e90d4: Download complete
1b4742b0c5f: Verifying Checksum
1b4742b0c5f: Download complete
6a73f7da97bb: Verifying Checksum
6a73f7da97bb: Download complete
7bdc9bbcf13: Verifying Checksum
7bdc9bbcf13: Download complete
f3b9c845d2f3: Verifying Checksum
f3b9c845d2f3: Download complete
cfcd43fe346d: Verifying Checksum
cfcd43fe346d: Download complete
576335d53cde: Verifying Checksum
576335d53cde: Download complete
caad4144446c: Verifying Checksum
caad4144446c: Download complete
2c2b2f17f9e3: Verifying Checksum
2c2b2f17f9e3: Download complete
2c2b2f17f9e3: Pull complete
d9c0b35c8a5: Pull complete
ec0d436c7f4: Pull complete
aa8e32a1da09: Pull complete
8950ab7e90d4: Pull complete
1b4742b0c5f: Pull complete
6a73f7da97bb: Pull complete
7bdc9bbcf13: Pull complete
60829730fa39: Download complete
f3b9c845d2f3: Pull complete
cfcd43fe346d: Pull complete
576335d53cde: Pull complete
caad4144446c: Pull complete
Digest: sha256:4e0b95f2b7983bde9f29c4e7ec5d1a23bc045f13b47680d86f701144ed7
Status: downloaded newer image for postgres:latest
docker.io/library/postgres:latest

What's Next?
View summary of image vulnerabilities and recommendations → docker scout quickview postgres

$ signalR.MPTOP-KEE52EAG MINGW64 ~
$ docker run -d --name postgres --e POSTGRES_PASSWORD=18114723 -p 5432:5432 postgres
430a7a28f817dddc6e275b8558f277a50bfdd9f0170475ef85a72b086d3f

$ signalR.MPTOP-KEE52EAG MINGW64 ~
$ docker run -d --name postgres:postgres postgres psql -h postgres -U postgres
The input device is not a TTY. If you are using minity, try prefixing the command with 'winpty'

$ signalR.MPTOP-KEE52EAG MINGW64 ~
$ winpty docker run -it --link postgres:postgres postgres psql -h postgres -U postgres
bash: winpty: command not found

$ signalR.MPTOP-KEE52EAG MINGW64 ~
$ winpty docker run -it --link postgres:postgres postgres psql -h postgres -U postgres
docker: Error response from daemon: Cannot link to a non running container: /postgres AS /fervent_mirzakhani/postgres.

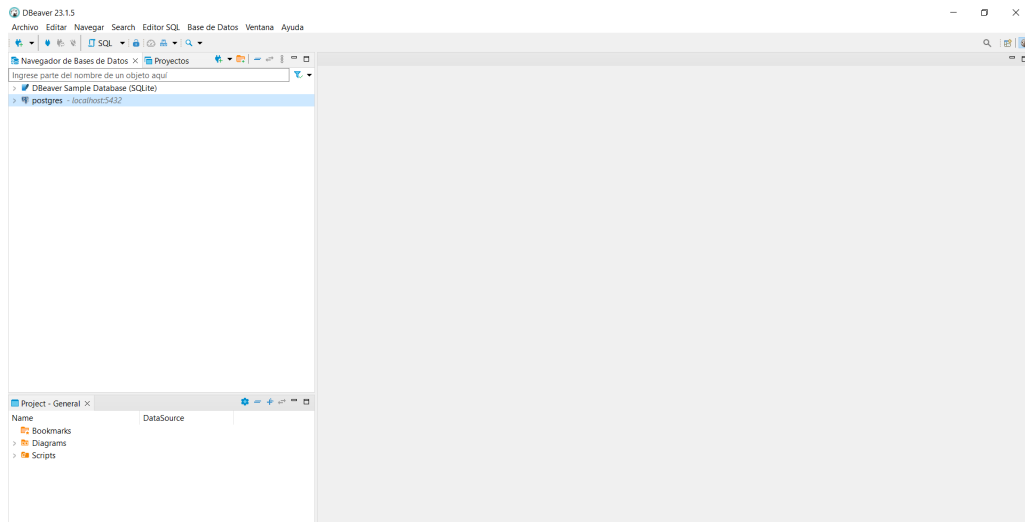
$ signalR.MPTOP-KEE52EAG MINGW64 ~
$ docker start postgres
postgres

$ signalR.MPTOP-KEE52EAG MINGW64 ~
$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
3c0482b29595       postgres           "docker-entrypoint.s..." 23 seconds ago      Created             5432/tcp            fervent_mirzakhani
d9c63da69922       hello-world        "/hello"            15 minutes ago      Exited (0) 15 minutes ago                                trusting_cori

$ signalR.MPTOP-KEE52EAG MINGW64 ~
$ winpty docker run -it --link postgres:postgres postgres psql -h postgres -U postgres
psql (13.4 (Debian 13.4-1.pgdg120+1))
Type "help" for help.

postgres=# \q
postgres=# \q
postgres=# exit
bye, us to quit.
postgres=# \q

$ signalR.MPTOP-KEE52EAG MINGW64 ~
$
```



6 Comentarios y los problemas a los que te enfrentaste en la instalación.

En Fedora con Docker tuve unos pequeños problemas, ya que primero instale Docker Engine y una vez que lo instale, también quise agregar Docker Desktop, pero ya cuando ahora si quería probar docker, no me aceptaba los comandos como `docker ps -a` o cualquier otro comando de docker, así que investigando, tenía que generar como una clave, e iniciar sesión.

Ya una vez que lo corrí, solo me pidió que cambiara unos permisos y ya solo es activarlo manualmente cada vez que vaya a ocuparlo. A diferencia que en Windows, que solo abres la app y ya puedes correr los comandos en terminal, en cambio en Fedora si abría primero la app, como que crasheaba y mejor lo tenía que iniciar desde terminal para después usar la app.

Decidí instalarlo en dos Sistemas Operativos, porque ya me toco en una practica de otra materia que los DockerFiles en Windows y en Linux se configuran distinto, creo que es por tema de permisos de Windows, ya que esa misma practica me daba errores de lecturas en otros archivos, mientras que en Linux a la primera hizo lo que tenía que mostrar.

Otro punto que también note, es que hay veces que cuando quieres iniciar el Docker, me parece que es activar el Demonio, hay veces que como no agarra bien y no se puede activar, para solucionarlo lo que vi es que solo era reiniciar la computadora.