Content

Database Postgres Server Docker

Note: This Tutorial is a demonstration for a local server

Connect to your server

```
client@user:~$ ssh ubuntu@192.168.1.187
```

or

```
client@user:~$ ssh -i ssh-key-2022-12-17-oracle.key ubuntu@129.151.251.242
```

Install Docker

```
ubuntu@ubuntu:~$ sudo apt install docker.io
```

Help

```
ubuntu@ubuntu:~$ docker --help
```

Version

```
ubuntu@ubuntu:~$ docker --version
```

Download Posrgres into docker

Download the latest version:

```
ubuntu@ubuntu:~$ sudo docker pull postgres
```

Use alpine for to use an lower image size:

```
ubuntu@ubuntu:~$ sudo docker pull postgres:alpine
```

Result

```
Using default tag: latest
latest: Pulling from library/postgres
6064e7e5b6af: Pull complete
e8306d459bcf: Pull complete
396926fa389d: Pull complete
7c168cbb66ad: Pull complete
98f614904561: Pull complete
6147ac60a15b: Pull complete
ebe7e874f17a: Pull complete
e51620d95271: Pull complete
bf50e10a1ebb: Pull complete
85a0a9724933: Pull complete
311616407ef9: Pull complete
```

```
c41a09226d37: Pull complete
6ce846177c98: Pull complete
Digest:
sha256:10d6e725f9b2f5531617d93164f4fc85b1739e04cab62cbfbfb81ccd866513b8
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
```

Docker Images installed

```
ubuntu@ubuntu:~$ sudo docker images
REPOSITORY   TAG       IMAGE ID       CREATED      SIZE
postgres     latest    5eea76716a19   5 days ago   359MB
```

Note: This version has a size of **359MB**, if you use an alpine image ~50MB.

Run instance

```
ubuntu@ubuntu:~$ sudo docker run --name postgres-0 -e
POSTGRES_PASSWORD=password -d -p 5432:5432 postgres
```

`--name [instance name]`

`-e POSTGRES_PASSWORD=[password]`

`-p [port:port]`, default: 5432:5432

`-d [postgres version]`

Result: Container ID:

```
8fe59819205bb2bbdd6a620784726b6f71bc38199aef37605a39e0c2f3552811
```

Verify running container

```
ubuntu@ubuntu:~$ sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS
PORTS                                       NAMES
8fe59819205b   postgres   "docker-entrypoint.s…"   2 minutes ago   Up 2
minutes   0.0.0.0:5432->5432/tcp, :::5432->5432/tcp   postgres-0
```

Note: Our server and database are operational 😄

Operation on our database

This part is optional.

Move into our docker container

```
ubuntu@ubuntu:~$ sudo docker exec -it postgres-0 bash
root@8fe59819205b:/# ls
bin   dev                  etc   lib    mnt  proc  run   srv  tmp  var
boot  docker-entrypoint-initdb.d  home  media  opt  root  sbin  sys  usr
root@8fe59819205b:/#
```

Execute postgresql

```
root@8fe59819205b:/# psql -U postgres
psql (15.1 (Debian 15.1-1.pgdg110+1))
Type "help" for help.

postgres=#
```

Note: use `-U` for set username.

List users

```
postgres=# \du
                                   List of roles
 Role name |                         Attributes                         |
Member of
-----------+------------------------------------------------------------+--
---------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS |
{}

postgres=#
```

Note: postgres is super user.

Create database

```
postgres=# CREATE DATABASE test;
```

List databases

```
postgres=# \l
                                                List of databases
   Name    |  Owner   | Encoding |  Collate   |   Ctype    | ICU Locale |
Locale Provider |   Access privileges
-----------+----------+----------+------------+------------+------------+--
---------------+-----------------------
 postgres  | postgres | UTF8     | en_US.utf8 | en_US.utf8 |            |
libc            |
 template0 | postgres | UTF8     | en_US.utf8 | en_US.utf8 |            |
libc            | =c/postgres          +
           |          |          |            |            |            |
| postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.utf8 | en_US.utf8 |            |
libc            | =c/postgres          +
           |          |          |            |            |            |
| postgres=CTc/postgres
 test      | postgres | UTF8     | en_US.utf8 | en_US.utf8 |            |
libc            |
(4 rows)

postgres=#
```

Connect to database

```
postgres=# \c test
You are now connected to database "test" as user "postgres".
test=#
```

List all tables

Exemple with gogocar database.

```
gogocar=# \dt
           List of relations
 Schema |   Name    | Type  |  Owner
--------+-----------+-------+----------
 public | addresses | table | postgres
 public | carmodel  | table | postgres
 public | city      | table | postgres
 public | modules   | table | postgres
 public | users     | table | postgres
 public | vehicles  | table | postgres
(6 rows)

gogocar=#
```

Connect to our database from client

```
client@user:~$ sudo psql -h 192.168.1.187 -p 5432 -U postgres
[sudo] password for user:
Password for user postgres:
psql (14.5 (Ubuntu 14.5-0ubuntu0.22.04.1), server 15.1 (Debian 15.1-
1.pgdg110+1))
WARNING: psql major version 14, server major version 15.
         Some psql features might not work.
Type "help" for help.

postgres=#
```

`-h [host]`

`-p [port]`

`-U [user]`

Check database list

```
postgres=# \l
                                 List of databases
   Name    |  Owner   | Encoding |  Collate   |   Ctype    |   Access
privileges
-----------+----------+----------+------------+------------+---------------
--------
 postgres  | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
 template0 | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres
+
           |          |          |            |            |
postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres
+
           |          |          |            |            |
postgres=CTc/postgres
 test      | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
(4 rows)

postgres=#
```

Server reboot

Sometimes the server need to be reboot, then the database is no longer available.

To repair that:

Connect to our server in ssh

```
ssh ubuntu@192.168.1.187
ubuntu@192.168.1.187's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1021-raspi aarch64)
```

Verify our docker image

```
ubuntu@ubuntu:~$ sudo docker images
REPOSITORY   TAG       IMAGE ID       CREATED      SIZE
postgres     latest    5eea76716a19   7 days ago   359MB
```

Verify running container

```
ubuntu@ubuntu:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
ubuntu@ubuntu:~$
```

Restart container

```
ubuntu@ubuntu:~$ sudo docker restart postgres-0
postgres-0
ubuntu@ubuntu:~$ sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS
PORTS                                       NAMES
8fe59819205b   postgres   "docker-entrypoint.s…"   44 hours ago   Up 5
seconds   0.0.0.0:5432->5432/tcp, :::5432->5432/tcp   postgres-0
ubuntu@ubuntu:~$
```

Note: the database is now available.

Note: We can use `--restart always` (but not tested for the moment)

Install Extension

Crypto

```
psql -U postgres -d gogocar
```

```
CREATE EXTENSION pgcrypto;
```

Verification

```
gogocar=# \dx
                  List of installed extensions
   Name   | Version |   Schema   |         Description
----------+---------+------------+------------------------------
 pgcrypto | 1.3     | public     | cryptographic functions
 plpgsql  | 1.0     | pg_catalog | PL/pgSQL procedural language
(2 rows)
```

Postgis

Connect to your server

```
ubuntu@ubuntu:~$ sudo docker exec -it postgres-0 bash
```

```
root@8fe59819205b:/# apt install postgis
```

```
root@8fe59819205b:/# psql -U postgres -d gogocar
```