

## Create a Postgres docker container

```
root@ip-172-31-34-172:~# docker run --name some-postgres -e POSTGRES_PASSWORD=mysecretpassword -d postgres
46340b5220291479b846badac1f1bca86babbe83dd916b474e0fa6469842ae10
root@ip-172-31-34-172:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
46340b522029   postgres  "docker-entrypoint.s..." 32 seconds ago Up 32 seconds  5432/tcp       some-postgres
root@ip-172-31-34-172:~#
```

## Get inside container and create a table

```
postgres=# show databases
postgres=# \list
          List of databases
  Name      | Owner   | Encoding | Locale Provider | Collate | Ctype   | Locale | ICU Rules | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----+-----
postgres   | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |         |           |
template0  | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |         |           | =c/postgres +
template1  | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |         |           | postgres=Ctc/postgres +
(3 rows)
```

```
postgres=# CREATE TABLE cars (
  brand VARCHAR(255),
  model VARCHAR(255),
  year INT
);
CREATE TABLE
postgres=# \dt
          List of relations
 Schema | Name  | Type  | Owner
-----+-----+-----+-----
 public | cars  | table | postgres
(1 row)

postgres=# \d cars
          Table "public.cars"
  Column      |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
 brand       | character varying(255) |           |          |
 model       | character varying(255) |           |          |
 year        | integer                |           |          |
```

## Inside Ubuntu container install

```
# apt update
```

```
# apt install -y postgresql-common
```

```
# apt install -y postgresql-client
```

And try connecting to existing postgres container in the same host

```
# psql -h 172.17.0.3 -p 5432 -d postgres -U
```

```
Selecting previously unselected package libjson-xs-perl.
Preparing to unpack .../38-libjson-xs-perl_4.030-2build3_amd64.deb ...
Unpacking libjson-xs-perl (4.030-2build3) ...
Setting up media-types (10.1.0) ...
Setting up libgdcm64:amd64 (1.23-5.1build1) ...
Setting up libgdcm-compat4t64:amd64 (1.23-5.1build1) ...
Setting up xdg-user-dirs (0.18-1build1) ...
Setting up libsqlite3-0:amd64 (3.45.1-1ubuntu2.1) ...
Setting up cron (3.0pl1-184ubuntu2) ...
invoke-rc.d: could not determine current runlevel
invoke-rc.d: policy-rc.d denied execution of start.
Created symlink /etc/systemd/system/multi-user.target.wants/cron.service → /usr/lib/systemd/system/cron.service.
Setting up tzdata (2024b-0ubuntu0.24.04.1) ...
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 79.)
debconf: falling back to frontend: Readline
Configuring tzdata

Please select the geographic area in which you live. Subsequent configuration questions will narrow this down by presenting a list of cities, representing the time
zones in which they are located.

 1. Africa 2. America 3. Antarctica 4. Arctic 5. Asia 6. Atlantic 7. Australia 8. Europe 9. Indian 10. Pacific 11. Etc 12. Legacy
Geographic area: [1] [#####.....]
```