

# Postgres

## Install

container xenial 16.04 (optional)

```
lxc image copy images:ubuntu/xenial/amd64 local: --alias xenial
lxc launch xenial postgres
lxc exec postgres bash
```

```
sudo apt install postgresql
```

eventually open network for remote connections

</etc/postgresql/9.5/main/postgresql.conf>

```
listen_addresses = '*'
```

## Basic operations

```
# list databases
\l

# connect to database
\c giano

# list schema
\dn

# list tables in schema
\dt auth.*

# list of installed extensions
\dx

# enable expanded display
\x
```

set search path for database

```
psql <database>
ALTER DATABASE database_name SET search_path TO schema_name,public;
```

drop role gianoauth

```
REASSIGN OWNED BY gianoauth TO postgres;  
DROP OWNED BY gianoauth;  
DROP USER IF EXISTS gianoauth;
```

create role gianoauth

```
CREATE USER gianoauth WITH  
  ENCRYPTED PASSWORD 'xxx'  
  LOGIN  
  NOSUPERUSER  
  INHERIT  
  NOCREATEDB  
  NOCREATEROLE  
  NOREPLICATION;
```

... create database and schemas ...

grant operations on schema auth to role gianoauth

```
GRANT ALL PRIVILEGES ON SCHEMA auth TO gianoauth;  
ALTER DEFAULT PRIVILEGES FOR USER postgres IN SCHEMA auth GRANT SELECT ON  
TABLES TO gianoauth;  
ALTER DEFAULT PRIVILEGES FOR USER postgres IN SCHEMA auth GRANT SELECT,  
USAGE ON SEQUENCES TO gianoauth;
```

on database giano set search path on user gianoauth to search first in schema auth

```
ALTER USER gianoauth SET search_path TO auth;  
ALTER ROLE gianoauth IN DATABASE giano SET search_path TO auth;
```

create superuser <admin> with password

```
createuser -P -s -e admin
```

create user and database for sentry

```
CREATE ROLE sentry WITH PASSWORD 'blabla' LOGIN;  
CREATE DATABASE sentry ENCODING 'UTF8' OWNER sentry;
```

## developer containers

- pgadmin:
  - user: admin@admin
  - pass: admin@admin

set static IP on containers

```
/var/lib/lxd/networks/lxdbr0/dnsmasq.hosts  
├─ pgmodeler
```

```
└─ postgres
```

```
cat > /var/lib/xd/networks/xdbr0/dnsmasq.hosts/pgmodeler <<EOF
00:16:3e:9d:ad:d2,10.8.120.205,pgmodeler
EOF
```

```
cat > /var/lib/xd/networks/xdbr0/dnsmasq.hosts/postgres <<EOF
00:16:3e:ce:41:53,10.8.120.228,postgres
EOF
```

## pgadmin

```
apt install wget python3-pip
pip3 install -U pip
cd /tmp
wget
https://ftp.postgresql.org/pub/pgadmin/pgadmin4/v3.5/pip/pgadmin4-3.5-py2.py
3-none-any.whl
pip install ./pgadmin4-3.5-py2.py3-none-any.whl

# define ip binding
CONFIG=/usr/local/lib/python3.5/dist-packages/pgadmin4/config.py
sed -i -e "s/DEFAULT_SERVER\s*=\s*/DEFAULT_SERVER = '0.0.0.0'/" $CONFIG

# define pgadmin admin user
python3 /usr/local/lib/python3.5/dist-packages/pgadmin4/pgAdmin4.py
```

[/etc/systemd/system/pgadmin.service](#)

```
[Unit]
Description=pgadmin Service
After=network.target

[Service]
User=root
ExecStart=/usr/bin/python3 /usr/local/lib/python3.5/dist-
packages/pgadmin4/pgAdmin4.py

[Install]
WantedBy=multi-user.target
```

```
systemctl daemon-reload
systemctl start pgadmin
systemctl enable pgadmin
```

define admin user in postgresql to use from pgadmin

```
su - postgres
```

```
createuser --pwprompt --superuser admin
```

## pgmodeler

- [reference](#)

```
PG_VERSION=0.9.1-beta1
```

```
# for building, can remove after install
```

```
sudo apt install -y make g++ qt5-qmake libxml2-dev libpq-dev pkg-config  
libqt5svg5-dev
```

```
# runtime dependencies
```

```
sudo apt install -y qt5-default libqt5svg5 postgresql-server-dev-all
```

```
cd /usr/local/src
```

```
sudo apt install -y wget
```

```
wget https://github.com/pgmodeler/pgmodeler/archive/v${PG_VERSION}.tar.gz
```

```
tar xvzf v${PG_VERSION}.tar.gz
```

```
cd pgmodeler-${PG_VERSION}/
```

```
qmake pgmodeler.pro
```

```
make -j 4
```

```
make install
```

```
sudo mkdir /usr/local/lib/pgmodeler/plugins
```

```
cd ..
```

```
rm *.tar.gz
```

```
apt clean
```

desktop launcher

```
mkdir -p ~/.local/share/applications/
```

```
cat > ~/.local/share/applications/pgmodeler.desktop <<EOF
```

```
#!/usr/bin/env xdg-open
```

```
[Desktop Entry]
```

```
Type=Application
```

```
Terminal=false
```

```
Exec=/usr/local/bin/pgmodeler
```

```
Name=pgmodeler
```

```
EOF
```

if in container enable remote connection on host

[/etc/gdm3/custom.conf](#)

```
[security]
```

```
DisallowTCP=false
```

```
[xdmcp]
```

```
Enable=true
```

run pgmodeler in container

```
xhost +  
lxc exec pgmodeler -- pgmodeler
```

From:

<https://wiki.csgalileo.org/> - **Galileo Labs**

Permanent link:

<https://wiki.csgalileo.org/tips/db/postgres?rev=1551091425>

Last update: **2019/02/25 11:43**

