one-step archlinux postgres installation

Contents

1 Introduction

let's try to summarize the essential steps to make sure postgres is installed flawlessly on every linux system with more stability than archlinux

2 Installation

2.1 package installation

$$sudo \quad pacman - S \quad postgresql$$

with the previous command the user postgres got created

2.1.1 assigning a password to the postgres user

$$sudo \quad passwd \quad postgres$$

2.2 add postgres user to sudoers

Let us put the postgres user in the sudoers straight away

$$sudo \quad visudo$$

now give to postgres the same privilegs as root. Just copy whatever is stated for root and replace root with postgres and then you can save and exit the vi environment

2.3 Let's log in under the name of postgres

$$sudo \quad -iu \quad postgres$$

Under the name of postgres we issue the command *psql* but there is an error coming up. Look:

```
[postgres@daniele-virtualbox2 ~]$ psql
psql: error: could not connect to server: No such file or directory
Is the server running locally and accepting
connections on Unix domain socket "/run/postgresql/.s.PGSQL.5432"?
```

$$initdb \quad -D \quad /var/lib/postgres/data$$

```
The files belonging to this database system will be owned by user
    "postgres".
This user must also own the server process.
The database cluster will be initialized with locales
COLLATE: en_GB.utf8
CTYPE:   en_GB.utf8
MESSAGES: en_GB.utf8
MONETARY: en_GB.UTF-8
NUMERIC: en_GB.UTF-8
TIME:    en_GB.UTF-8
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".
Data page checksums are disabled.
fixing permissions on existing directory /var/lib/postgres/data
    ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... Europe/London
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok
initdb: warning: enabling "trust" authentication for local
    connections
You can change this by editing pg_hba.conf or using the option -A,
    or
```

```
--auth-local and --auth-host, the next time you run initdb.
Success. You can now start the database server using:
pg_ctl -D /var/lib/postgres/data -l logfile start
```

$$pg\_ctl \quad -D \quad /var/lib/postgres/data \quad -l \quad logfile \quad start$$

```
waiting for server to start..../bin/sh: line 1: logfile:
    Permission denied
stopped waiting
pg_ctl: could not start server
Examine the log output.
```

$$systemctl \quad restart \quad postgresql.service$$

$$systemctl \quad enable \quad postgresql.service$$

```
Created symlink
    /etc/systemd/system/multi-user.target.wants/postgresql.service
    /usr/lib/systemd/system/postgresql.service
```

let us now issue the final command

$$psql$$

and what you should get is hopefully this

```
psql (13.4)
Type "help" for help.
postgres=#
```

3 You're all set

If you got the output above that means you're in