Install ppostgress in Debian and Ubuntu Linux

On Linux, after a fresh install (`apt install postgresql`, `dnf install postgresql-server`, etc.):

- **Default user:** `postgres` — both as a Linux system user and as the PostgreSQL superuser role.
- **Default password:** none. There is no preset password.

By default the `postgres` role uses **peer authentication** for local connections, which means PostgreSQL trusts the OS user. So you log in like this:

bash

```
sudo -u postgres psql
```

No password prompt — it works because you're acting as the Linux `postgres` user, and PostgreSQL matches that to the `postgres` role.

If you want to use the script I just made (which connects over TCP with a password), you need to **set a password** first:

bash

```
sudo -u postgres psql
```

Then inside `psql`:

sql

```
ALTER USER postgres WITH PASSWORD 'your_password_here';
\q
```

You may also need to allow password logins over TCP. Edit `pg_hba.conf` (usually at `/etc/postgresql/<version>/main/pg_hba.conf` on Debian/Ubuntu, or `/var/lib/pgsql/data/pg_hba.conf` on RHEL/Fedora) and find lines like:

```
local   all   postgres   peer
host    all   all        127.0.0.1/32   ident
```

Change the method to `scram-sha-256` (modern) or `md5` (older):

```
local   all   postgres   scram-sha-256
host    all   all        127.0.0.1/32   scram-sha-256
```

Then reload:

bash

```
sudo systemctl reload postgresql
```

After that, `psql -U postgres -h localhost` will prompt for the password you set, and the Python script will work with `host=localhost`, `user=postgres`, and your chosen password.

One caveat: some distros (notably certain Docker images or the official PostgreSQL apt repo) behave a little differently. The Docker `postgres` image, for instance, requires you to set `POSTGRES_PASSWORD` as an env var at container creation. But for a plain Linux package install, the answer is: user `postgres`, no password, peer auth.

Revision #1
Created 19 May 2026 16:19:03 by Admin
Updated 19 May 2026 16:20:01 by Admin