Install PostgreSQL 13 on Ubuntu 22.04|20.04|18.04

By **Josphat Mutai** - August 22, 2023

PostgreSQL is a very popular open source Object-Relational database management system (DBMS) created for reliability, efficiency and data integrity guarantee. The development work is now over 30 years and the project has earned it a strong reputation for reliability, feature robustness, and performance. In this tutorial we will explain how you can install and configure PostgreSQL 13 Database server on Ubuntu 22.04|20.04|18.04 Linux server.

PostgreSQL is used by thousands of companies to power payment transactions, huge website traffic, E-commerce platforms e.t.c. It also enables you to add custom functions developed using programming languages such as Java, Python, C/C++ and many others.

As of this article update the latest release of PostgreSQL is 13 which is intended for use to run workloads in Production environments. You can find information about all of the features and changes found in PostgreSQL 13 in the release notes.

Step 1: Update Ubuntu system

We always work on a latest release of OS to make sure there are no old dependency issues. Login to your Ubuntu server and run the following commands to update all the packages installed.

```
sudo apt update && sudo apt -y full-upgrade
[ -f /var/run/reboot-required ] && sudo reboot -f
```

Once the system has been updated, I recommend you perform a reboot to get the new kernel running incase it was updated.

```
sudo reboot
```

Step 2: Add PostgreSQL 13 repository

Install required dependency packages

```
sudo apt update
sudo apt install curl gpg gnupg2 software-properties-common apt-
transport-https lsb-release ca-certificates
```

Now that we have updated and rebooted our system, let's add the APT repository required to pull the packages form the PostgreSQL repository.

```
curl -fsSL https://www.postgresql.org/media/keys/ACCC4CF8.asc|sudo
gpg --dearmor -o /etc/apt/trusted.gpg.d/postgresql.gpg
```

After importing GPG key, add repository contents to your Ubuntu 22.04|20.04|18.04 system:

```
echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb_release -cs`-
pgdg main" |sudo tee  /etc/apt/sources.list.d/pgdg.list
```

The repository added contains many different packages including third party addons. They include:

- *postgresql-client*
- *postgresql*
- *libpq-dev*
- *postgresql-server-dev*
- *pgadmin* packages

Step 3: Install PostgreSQL 13

With the repository added we can install the PostgreSQL 13 packages on our Ubuntu 22.04|20.04|18.04 Linux server. But first update the package index for the version to be available at the OS level.

```
sudo apt update
```

The run the commands below to install PostgreSQL 13 on Ubuntu 22.04|20.04|18.04 Linux system.

```
sudo apt install postgresql-13 postgresql-client-13
```

The PostgreSQL service is started and set to come up after every system reboot.

```
$ systemctl status postgresql@13-main.service
● postgresql@13-main.service - PostgreSQL Cluster 13-main
     Loaded: loaded (/lib/systemd/system/postgresql@.service;
enabled-runtime; vendor preset: enabled)
     Active: active (running) since Wed 2022-05-18 15:50:22 EAT; 1min
55s ago
```

```
     Process: 4294 ExecStart=/usr/bin/pg_ctlcluster --skip-systemctl-
redirect 13-main start (code=exited, status=0/SUCCESS)
    Main PID: 4299 (postgres)
       Tasks: 7 (limit: 9460)
      Memory: 18.2M
         CPU: 150ms
      CGroup: /system.slice/system-postgresql.slice/postgresql@13-
main.service
             ├─4299 /usr/lib/postgresql/13/bin/postgres -D
/var/lib/postgresql/13/main -c
config_file=/etc/postgresql/13/main/postgresql.conf
             ├─4301 "postgres: 13/main: checkpointer " "" "" "" "" ""
"" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
"" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ">
             ├─4302 "postgres: 13/main: background writer " "" "" ""
"" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
"" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "">
             ├─4303 "postgres: 13/main: walwriter " "" "" "" "" "" ""
"" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
"" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ">
             ├─4304 "postgres: 13/main: autovacuum launcher " "" ""
"" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
"" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" >
             ├─4305 "postgres: 13/main: stats collector " "" "" "" ""
"" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
"" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ">
             └─4306 "postgres: 13/main: logical replication launcher
" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
"" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" >

Mei 18 15:50:19 ubuntu22 systemd[1]: Starting PostgreSQL Cluster 13-
main...
Mei 18 15:50:22 ubuntu22 systemd[1]: Started PostgreSQL Cluster 13-
main.
```

Step 4: Test PostgreSQL Connection

During installation, a postgres user is created automatically. This user has full **superadmin** access to your entire PostgreSQL instance. Before you switch to this account, your logged in system user should have sudo privileges.

```
sudo su - postgres
```

Let's reset this user password to a strong Password we can remember.

```
psql -c "alter user postgres with password 'StrongAdminP@ssw0rd'"
```

Start PostgreSQL prompt by using the command:

```
$ psql
```

Get connection details like below.

```
$ psql
psql (13.7 (Ubuntu 13.7-1.pgdg22.04+1))
Type "help" for help.

postgres=# \conninfo
You are connected to database "postgres" as user "postgres" via
socket in "/var/run/postgresql" at port "5432".
```

Let's create a test database and user to see if it's working.

```
postgres=# CREATE DATABASE mytestdb;
CREATE DATABASE
postgres=# CREATE USER mytestuser WITH ENCRYPTED PASSWORD
'MyStr0ngP@SS';
CREATE ROLE
postgres=# GRANT ALL PRIVILEGES ON DATABASE mytestdb to mytestuser;
GRANT
```

List created databases:

```
postgres=# \l
                                 List of databases
   Name    |  Owner   | Encoding | Collate |  Ctype  |    Access
privileges
-----------+----------+----------+---------+---------+---------------
----------
 mytestdb  | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =Tc/postgres
+
           |          |          |         |         |
postgres=CTc/postgres  +
           |          |          |         |         |
mytestuser=CTc/postgres
 postgres  | postgres | UTF8     | C.UTF-8 | C.UTF-8 |
```

```
 template0 | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres
+
           |          |          |         |         |
postgres=CTc/postgres
 template1 | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres
+
           |          |          |         |         |
postgres=CTc/postgres
(4 rows)
```

Connect to database:

```
postgres-# \c mytestdb
You are now connected to database "mytestdb" as user "postgres".
```

Other PostgreSQL utilities installed such as **createuser** and **createdb** can be used to create database and users.

```
postgres@ubuntu:~$ createuser myuser --password
Password:
postgres@ubuntu:~$ createdb mydb -O myuser
postgres@ubuntu:~$ psql -l
```

We can create and connect to a database on PostgreSQL server.

Step 5: Configure remote Connection (Optional)

Installation of PostgreSQL 13 on Ubuntu only accepts connections from localhost. In ideal production environments, you'll have a central database server and remote clients connecting to it – But of course within a **private network** (LAN).

To enable remote connections, edit PostgreSQL configuration file:

```
sudo nano /etc/postgresql/13/main/postgresql.conf
```

Uncomment line **59** and change the Listen address to accept connections within your networks.

```
# Listen on all interfaces
listen_addresses = '*'

# Listen on specified private IP address
listen_addresses = '192.168.10.11'
```

Also set PostgreSQL to accept remote connections from allowed hosts.

```
$ sudo nano /etc/postgresql/13/main/pg_hba.conf

# Accept from anywhere
host all all 0.0.0.0/0 md5

# Accept from trusted subnet
host all all 10.10.10.0/24 md5
```

After the change, restart postgresql service.

```
sudo systemctl restart postgresql
```

Confirm Listening addresses.

```
# netstat  -tunelp | grep 5432
tcp        0      0 0.0.0.0:5432            0.0.0.0:*
LISTEN      111        112837     11143/postgres
tcp6       0      0 :::5432                 :::*
LISTEN      111        112838     11143/postgres
```

Step 6: Install pgAdmin4 Management Tool

If you want to manage your PostgreSQL database server from a web interface, then install pgAdmin4.

- Install pgAdmin4 on Ubuntu

Enjoy using PostgreSQL 13 on Ubuntu 22.04|20.04|18.04. Other guides related to databases are shared in the list below.

- How to Install PostGIS on Ubuntu
- Install PostgreSQL 13 on CentOS 7
- Install PostgreSQL on CentOS 8 / RHEL 8

Tags:

- Install PostgreSQL 13 on Ubuntu 22.04|20.04|18.04
- Install PostgreSQL 13 on Ubuntu

YOU CAN SUPPORT OUR WORK WITH A CUP OF COFFEE

As we continue to grow, we would wish to reach and impact more people who visit and take advantage of the guides we have on our blog. This is a big task for us and we are so far extremely grateful for the kind people who have shown amazing support for our work over the time we have been online.

Thank You for your support as we work to give you the best of guides and articles. Click below to buy us a coffee.

Josphat Mutai

https://computingforgeeks.com/

Founder of Computingforgeeks. Expertise in Virtualization, Cloud, Linux/UNIX Administration, Automation,Storage Systems, Containers, Server Clustering e.t.c.