

How to Install PostgreSQL on Ubuntu 22.04

Learn how to install PostgreSQL on Ubuntu 22.04 with our step-by-step guide. Includes setup instructions, configuration tips, and common troubleshooting solutions.

Contents

01	Introduction	3
02	Prerequisites	3
03	Install PostgreSQL	3
04	Secure the PostgreSQL Database Server	5
05	Access the PostgreSQL Database Server	6
06	Conclusion	8

Introduction

PostgreSQL is an open-source, advanced RDBMS designed for managing a wide range of data tasks. It supports SQL to handle both small and large applications, including analytics, GIS, healthcare, and dynamic web apps.

In this article, you are to install PostgreSQL on an Ubuntu 22.04 server, and finally secure it for production use on your server.

Prerequisites

Before you begin:

- Have an [Ubuntu 22.04 server](#).
- Access the server using SSH as a non-root user with sudo privileges.
- [Update the server](#).

Install PostgreSQL

PostgreSQL is available in Ubuntu's default APT repositories. Follow the steps below to install PostgreSQL and configure it to start automatically at boot.

1. Update the server package index.

CONSOLE

```
$ sudo apt update
```

2. Install the `postgresql-common` dependency package on your server.

CONSOLE

```
$ sudo apt install -y postgresql-common -y
```

3. Run the following command to execute the PostgreSQL APT repository script.

CONSOLE

```
$ sudo /usr/share/postgresql-common/pgdg/  
apt.postgresql.org.sh
```

Press Enter when prompted to add the new repository to your server sources.

This script will enable the PostgreSQL APT repository on apt.postgresql.org on your system. The distribution codename used will be noble-pgdg.

Press Enter to continue, or Ctrl-C to abort.

4. Install the `postgresql` database server package.

CONSOLE

```
$ sudo apt install -y postgresql
```

5. Start the PostgreSQL database server.

CONSOLE

```
$ sudo systemctl start postgresql
```

6. View the PostgreSQL system service status and verify that it's active.

CONSOLE

```
$ sudo systemctl status postgresql
```

Output:

- postgresql.service - PostgreSQL RDBMS
Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; preset:

```
enabled)
  Active: active (exited) since Mon 2024-05-27 16:09:21 UTC; 35s ago
  Process: 5601 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
Main PID: 5601 (code=exited, status=0/SUCCESS)
  CPU: 3ms
```

Secure the PostgreSQL Database Server

PostgreSQL uses the default `postgres` user account. Follow the steps below to enable password authentication and secure the server by granting access only to authorized users.

1. Check your installed PostgreSQL version.

CONSOLE

```
$ psql --version
```

Output:

```
psql (PostgreSQL) 17.4 (Ubuntu 17.4-1.pgdg22.04+2)
```

2. Log in to the PostgreSQL database server using the `postgres` user account.

CONSOLE

```
$ sudo -u postgres psql
```

3. Modify the default `postgres` user with a new strong password.

SQL

```
postgres=# ALTER USER postgres WITH ENCRYPTED PASSWORD
'strong_password';
```

4. Create a new user `db_manager` with a new strong password.

```
SQL
```

```
postgres=# CREATE USER db_manager ENCRYPTED PASSWORD  
'strong_password';
```

5. Exit the PostgreSQL console.

```
SQL
```

```
postgres=# quit;
```

6. Run the following command to change the default `peer` value in the `scram-sha-256` field in the main PostgreSQL configuration file `pg_hba.conf` to enable password authentication on the server.

```
CONSOLE
```

```
$ sudo sed -i '/^local/s/peer/scram-sha-256/' /etc/  
postgresql/17/main/pg_hba.conf
```

Replace `17` with your installed PostgreSQL version number if it's different.

7. Restart the PostgreSQL server to apply the new configuration changes.

```
CONSOLE
```

```
$ sudo systemctl restart postgresql
```

Access the PostgreSQL Database Server

Access the PostgreSQL database console using the pre-installed `psql` utility or graphical tools for a direct connection. Follow the steps below to create a new sample database for use with a non-privileged user.

1. Create a new sample PostgreSQL database `hospital` and grant the `db_manager` user ownership privileges to the database.

CONSOLE

```
$ sudo -u postgres createdb hospital -O db_manager
```

When prompted, enter the Postgres user password you created earlier.

2. Log in to the PostgreSQL database as the user `db_manager` to test access to the `hospital` database.

CONSOLE

```
$ sudo -u postgres psql -U db_manager -d hospital
```

Enter the database user password when prompted and press Enter to access the database.

3. Create a new sample `doctors` table.

SQL

```
hospital=> CREATE TABLE doctors (  
    doctor_id SERIAL PRIMARY KEY,  
    first_name VARCHAR(50),  
    last_name VARCHAR(50),  
    appointment_date DATE  
);
```

The above SQL statement creates a `doctors` table in the `hospital` database with the following columns:

- `doctor_id` as a `PRIMARY KEY` to uniquely identify each doctor.
- `first_name` and `last_name` for storing doctor names.
- `appointment_date` for storing the doctor's appointment date.
- `SERIAL` generates a unique `doctor_id` for each record.

4. Insert sample data into the `doctors` table.

SQL

```
hospital=> INSERT INTO doctors
  ( first_name, last_name, appointment_date)
VALUES
  ( 'Ben', 'Joe', '2024-11-15' ),
  ( 'Carson', 'Smith', '2024-02-28' ),
  ( 'Donald', 'James', '2024-04-10' );
```

5. Query the `doctors` table to view all available records.

```
SQL
```

```
hospital=> SELECT * FROM doctors;
```

Output:

```
SQL
```

```
doctor_id | first_name | last_name | appointment_date
-----+-----+-----+-----
      1 | Ben        | Joe       | 2024-11-15
      2 | Carson     | Smith    | 2024-02-28
      3 | Donald    | James    | 2024-04-10
(3 rows)
```

6. Exit the PostgreSQL console.

```
SQL
```

```
hospital=> \q
```

Conclusion

In this article, you installed PostgreSQL on your Ubuntu 22.04 server and used the `psql` utility to create databases and tables. You can now integrate PostgreSQL with your applications for secure database management. For more details, visit the [PostgreSQL documentation](#).



VULTR

