

How to Install and Setup PostgreSQL server on Ubuntu 20.04

by Mantas Levinas

October 8th, 2021

LINUX

Tutorial

How to Install and Setup PostgreSQL Server on Ubuntu 20.04



Introduction

PostgreSQL is a fully featured database management system (DBMS) with SQL compliance. It is backed by 20 years of open-source development (relational) and JSON (non-relational) querying.



Welcome aboard! :)Real people are available right here 24/7. How can I help you today?



POPULAR TOPICS ▾



You may want to install PostgreSQL from an official repository, since it is updated more frequently than official Ubuntu sources.

First, you should install prerequisite software packages that will be used to download and install software certificates for a secure SSL connection.

```
sudo apt install wget ca-certificates
```

Then, get the certificate, add it to apt-key management utility and create a new configuration file with an official PostgreSQL repository address inside.

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key  
add -
```

```
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ $(lsb_release -cs)-  
pgdg main" >> /etc/apt/sources.list.d/pgdg.list'
```

Install PostgreSQL

It is always a good idea to download information about all packages available for installation from your configured sources before the actual installation.

```
sudo apt update
```

Now is the time to do the actual PostgreSQL installation. This will install PostgreSQL with the newest extensions and additions that are not yet officially part of the distribution.

```
apt install postgresql postgresql-contrib
```

Check PostgreSQL status

✕
Welcome aboard! :)Real people
are available right here 24/7.
How can I help you today?

```
root@ubuntu-sandbox:~ > service postgresql status
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Tue 2022-03-15 09:56:35 EET; 46s ago
     Main PID: 3046735 (code=exited, status=0/SUCCESS)
       Tasks: 0 (limit: 19175)
      Memory: 0B
      CGroup: /system.slice/postgresql.service

Mar 15 09:56:35 ubuntu-sandbox systemd[1]: Starting PostgreSQL RDBMS...
Mar 15 09:56:35 ubuntu-sandbox systemd[1]: Finished PostgreSQL RDBMS.
```

Start Using PostgreSQL Command Line Tool

When you install PostgreSQL a default admin user “postgres” is created by the default. You must use it to log-in to your PostgreSQL database for the first time.

A “psql” command-line client tool is used to interact with the database engine. You should invoke it as a “postgres” user to start an interactive session with your local database.

```
sudo -u postgres psql
```

In addition to creating a postgres admin user for you, PostgreSQL installation also creates a default database named “postgres” and connects you to it automatically when you first launch psql.

After first launching psql, you may check the details of your connection by typing `\conninfo` into the interpreter.

```
postgres=# \conninfo
You are connected to database "postgres" as user "postgres" via socket in
```

You are now connected to database “postgres” as user “postgres”.

If you want to see a list of all the databases that are available on a server, use `\l` command.

✕
Welcome aboard! :)Real people
are available right here 24/7.
How can I help you today?

```
template1 | postgres | UTF8 | C.UTF-8 | C.UTF-8 | =c/postgres +
          |          |          |          |          | postgres=Ct/postgres
(3 rows)
```

And to see a list of all the users with their privileges use \du command.

```
postgres=# \du
               List of roles
Role name | Attributes | Member of
-----+-----+-----
postgres | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
```

Since the default “postgres” user does not have a password, you should set it yourself.

```
\password postgres
```

Create and Populate a New Database

You are now connected to your database server through psql command line tool with full access rights, so it's time to create a new database.

```
CREATE DATABASE test_erp;
```

After the new “test_erp” database is created, connect to it.

```
\c test_erp
```

Now you are ready to start creating tables where your data will be stored. We will create a table with a primary key, and three client attributes.

```
CREATE TABLE clients (id SERIAL PRIMARY KEY, first_name VARCHAR, last_name VARCHAR, role VARCHAR);
```



Welcome aboard! :) Real people are available right here 24/7. How can I help you today?

```
public | clients | table | postgres
(1 row)
```

Let's now insert the first row into your newly created "clients" table.

```
INSERT INTO clients (first_name, last_name, role) VALUES ('John', 'Smith', 'CEO');
```

And query the table to get all its rows.

```
SELECT * FROM clients;
```

```
test_erp=# SELECT * FROM clients;
 id | first_name | last_name | role
----+-----+-----+----
  1 | John      | Smith    | CEO
(1 row)
```

As you can see, John Smith has been successfully added to the "clients" table of the "test_erp" database.

Setup PostgreSQL server

It's fun to play with the database locally, but eventually you will need to connect to the server.

When you install a PostgreSQL server, it is only accessible locally through the host machine. However, you may change this setting in the PostgreSQL configuration file.

Let's now exit the interactive psql session by typing exit, and access the `postgresql.conf` configuration file of PostgreSQL version 14 by using vim text editor.



Welcome aboard! :)Real people are available right here 24/7. How can I help you today?



POPULAR TOPICS ▾

`listen_addresses =`

Now edit the PostgreSQL access policy configuration file.

```
vim /etc/postgresql/14/main/pg_hba.conf
```

Append a new connection policy (a pattern stands for [CONNECTION_TYPE] [DATABASE] [USER] [ADDRESS] [METHOD]) in the bottom of the file.

```
host all all 0.0.0.0/0 md5
```

We are allowing TCP/IP connections (host) to all databases (all) for all users (all) with any IPv4 address (0.0.0.0/0) using an MD5 encrypted password for authentication (md5).

It is now time to restart your PostgreSQL service to load your configuration changes.

```
systemctl restart postgresql
```

And make sure your system is listening to the 5432 port that is reserved for PostgreSQL.

```
ss -nlt | grep 5432
```

```
root@ubuntu-sandbox:~ > ss -nlt | grep 5432
LISTEN 0      244          0.0.0.0:5432
LISTEN 0      244          [::]:5432
```

Welcome aboard! :)Real people are available right here 24/7. How can I help you today?

If everything is OK, you should see this output.

Connect to PostgreSQL database through a remote host



POPULAR TOPICS ▾



remote machine yet, follow the steps 1 – 3 for a full PostgreSQL installation or install a command line tool only by using `sudo apt install postgresql-client` command.

You may now connect to a remote database by using the following command pattern:

```
psql -h [ip address] -p [port] -d [database] -U [username]
```

Let's now connect to a remote PostgreSQL database that we have hosted on one of the [Cherry Servers](#) machines.

```
psql -h 5.199.162.56 -p 5432 -d test_erp -U postgres
```

To double check your connection details use the `\conninfo` command.

```
test_erp=# \conninfo
You are connected to database "test_erp" as user "postgres" on host "5.199.162.56" at port "5432".
```

Now you can start writing SQL queries to retrieve data from your database tables.

```
SELECT * FROM clients;
```

We can see that our previously created entry is safely stored in the “clients” table.

```
test_erp=# SELECT * FROM clients;
 id | first_name | last_name | role
----+-----+-----+----
  1 | John      | Smith    | CEO
(1 row)
```



Welcome aboard! :)Real people are available right here 24/7. How can I help you today?

If you are using Python, a standard PostgreSQL driver is psycopg2. Let's install this library using pip package manager.

```
pip install psycopg2-binary
```

You can now import psycopg2 into your code and start using PostgreSQL natively.

```
import psycopg2

# Connect to your PostgreSQL database on a remote server
conn = psycopg2.connect(host="5.199.162.56", port="5432", dbname="test_erp", user="postgres", password="123456")

# Open a cursor to perform database operations
cur = conn.cursor()

# Execute a test query
cur.execute("SELECT * FROM clients")

# Retrieve query results
records = cur.fetchall()

# Finally, you may print the output to the console or use it anyway you like
print(records)
```

You will get the following output when using an ipython3 interpreter:



Welcome aboard! :)Real people
are available right here 24/7.
How can I help you today?


```
In [4]: cur.execute("SELECT * FROM clients")

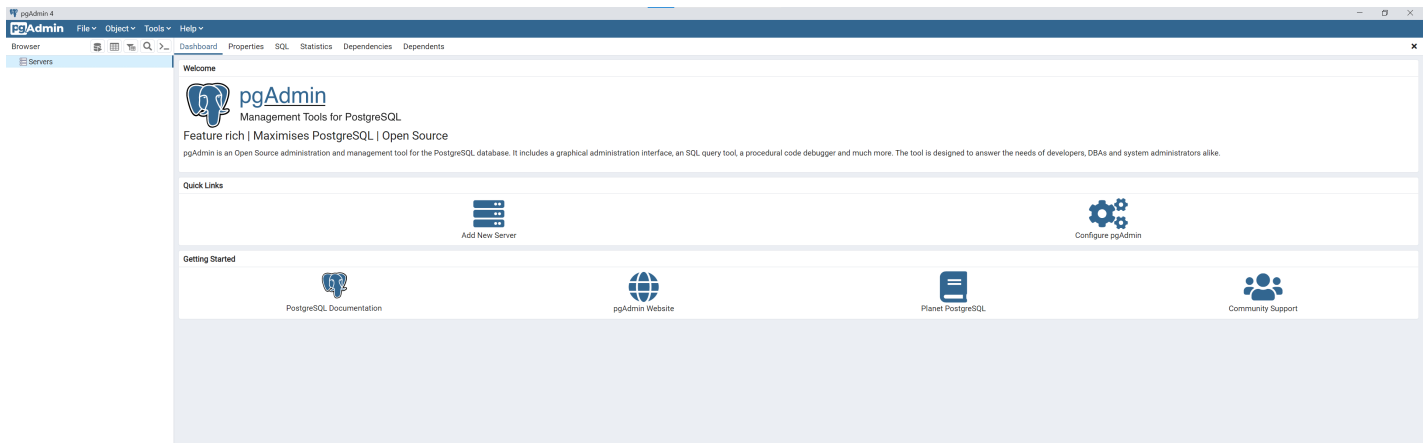
In [5]: records = cur.fetchall()

In [6]: print(records)
[(1, 'John', 'Smith', 'CEO')]
```

Connect via GUI Client

Although there are many GUI clients that can help you connect to a database and manage it, pgAdmin is probably the most popular option for PostgreSQL, and we highly recommend using it.

After [installing pgAdmin 4](#) and running it you will get to a standard pgAdmin 4 dashboard.



Press Add New Server button and enter the information of your remote server.

Welcome aboard! :)Real people
are available right here 24/7.
How can I help you today?

name/address

Port

5432

Maintenance database

postgres

Username

postgres

Kerberos authentication?

False

Password

.....

Save password?

☐

Role

Service

i

?

✕ Cancel

↺ Reset

Save

After saving your credentials you will be automatically connected to a re

Welcome aboard! :)Real people are available right here 24/7. How can I help you today?



POPULAR TOPICS ▾



Sessions	Locks	Prepared Transactions							
PID	User	Application	Client	Backend start	State	Wait event	Blocking PIDs		
3050746	postgres		5.199.160.9	2022-03-15 10:42:50 EET	idle in transaction	Client: ClientRead			
3051380	postgres	pgAdmin 4 - DB: test_erp	5.199.160.9	2022-03-15 11:02:09 EET	active				

You may now open a Query Tool for your selected database test_erp and start writing your queries.

The screenshot shows the pgAdmin 4 web interface. The left sidebar shows the 'Servers' tree with 'Test ERP database' expanded, showing 'Databases (2)' including 'postgres' and 'test_erp'. The 'test_erp' database is selected. The main panel shows the 'Query Editor' with the following SQL query:

```
1 SELECT *
2 FROM clients;
```

The 'Data Output' tab is active, showing the results of the query:

id	first_name	last_name	role
[PK] integer	character varying	character varying	character varying
1	John	Smith	CEO

Conclusion

Congrats! You have successfully installed a PostgreSQL database, set-up a database server and started interacting with it through a remote machine. It is now time to dive deeper into the [official PostgreSQL documentation](#) to build your application.

Welcome aboard! :) Real people are available right here 24/7. How can I help you today?

Share this article



Join Cherry Servers Community

Get monthly practical guides about building more secure, efficient and easier to scale systems on an open cloud ecosystem.

[Subscribe](#)☐ I agree to receive marketing communication as per privacy policy

Related Articles

LINUX



Welcome aboard! :)Real people are available right here 24/7.
How can I help you today?

How To Back Up A Linux Server

Jan 21, 2020 by Marius Rimkus

Short tutorial how to backup and restore your Linux server using TAR method and Bera backup tools.

LINUX

openSUSE Leap Review: What's New

May 17, 2021 by Mantas Levinas

openSUSE Leap 15.2 is one of the most versatile stable distributions available today, offering powerful tools for system developers and home users alike.



Welcome aboard! :)Real people are available right here 24/7.
How can I help you today?

AlmaLinux Review: a CentOS Clone Supported by CloudLinux

Jun 7, 2021 by Mantas Levinas

AlmaLinux is an open-source Linux distribution focused on long-term stability, that is a 1:1 binary compatible fork of Red Hat Enterprise Linux (RHEL)

Ready to get started?

Create an Account



Welcome aboard! :)Real people are available right here 24/7. How can I help you today?

[POPULAR TOPICS](#) Sales: sales@cherryservers.comSupport: support@cherryservers.com

GET THE LATEST NEWS FIRST

Subscribe to our newsletter

[Subscribe](#)☐ I agree to receive marketing communication as per privacy policy

PRICING

[Dedicated Servers](#)[Virtual Servers](#)[Custom Server Builder](#)

PRODUCTS

[Dedicated Servers](#)[Virtual Servers](#)[Spot Servers](#)[Cloud Platform](#)

RESOURCES

[API & Integrations](#)[Case Studies](#)[Knowledgebase](#)

Welcome aboard! :)Real people
are available right here 24/7.
How can I help you today?

POPULAR TOPICS [Core Values](#)[EU Projects](#)

LEGAL

[Terms of Service](#)[Privacy Policy](#)[Refund Policy](#)

© Cherry Servers 2001-2022



Welcome aboard! :)Real people
are available right here 24/7.
How can I help you today?

