Install and configure PostgreSQL

PostgreSQL <https://www.postgresql.org/> (commonly referred to as “Postgres”) is an object-relational database system that has all the features of traditional commercial database systems, but with enhancements to be found in next-generation database management systems (DBMS).

Install PostgreSQL

To install PostgreSQL, run the following command in the command prompt:

```
sudo apt install postgresql
```

The database service is automatically configured with viable defaults, but can be customised based on your specific needs.

Configure PostgreSQL

PostgreSQL supports multiple client authentication methods. In Ubuntu, `peer` is the default authentication method used for `local` connections, while `scram-sha-256` is the default for `host` connections (this used to be `md5` until Ubuntu 21.10). Please refer to the PostgreSQL Administrator’s Guide <http://www.postgresql.org/docs/current/static/admin.html> if you would like to configure alternatives like Kerberos.

The following discussion assumes that you wish to enable TCP/IP connections and use the MD5 method for client authentication. PostgreSQL configuration files are stored in the `/etc/postgresql/<version>/main` directory. For example, if you install PostgreSQL 14, the configuration files are stored in the `/etc/postgresql/14/main` directory.

> **Tip**:
> To configure *IDENT* authentication, add entries to the `/etc/postgresql/*/main/pg_ident.conf` file. There are detailed comments in the file to guide you.

By default, only connections from the local system are allowed. To enable all other computers to connect to your PostgreSQL server, edit the file `/etc/postgresql/*/main/postgresql.conf`. Locate the line: *#listen_addresses = ‘localhost’* and change it to `*`:

```
listen_addresses = '*'
```

> **Note**:
> ‘*’ will allow all available IP interfaces (IPv4 and IPv6), to only listen for IPv4 set `0.0.0.0` while ‘`::`’ allows listening for all IPv6 addresses.

For details on other parameters, refer to the configuration file or to the PostgreSQL documentation <https://www.postgresql.org/docs/> for information on how they can be edited.

Now that we can connect to our PostgreSQL server, the next step is to set a password for the `postgres` user. Run the following command at a terminal prompt to connect to the default PostgreSQL template database:

```
sudo -u postgres psql template1
```

The above command connects to PostgreSQL database `template1` as user `postgres`. Once you connect to the PostgreSQL server, you will be at an SQL prompt. You can run the following SQL command at the `psql` prompt to configure the password for the user `postgres`.

```
ALTER USER postgres with encrypted password 'your_password';
```

After configuring the password, edit the file `/etc/postgresql/*/main/pg_hba.conf` to use `scram-sha-256` authentication with the `postgres` user, allowed for the `template1` database, from any system in the local network (which in the example is `192.168.122.1/24`) :

```
hostssl template1       postgres        192.168.122.1/24        scram-sha-256
```

> **Note**:
> The config statement `hostssl` used here will reject TCP connections that would not use SSL. PostgreSQL in Ubuntu has the SSL feature built in and configured by default, so it works right away. On your PostgreSQL server this uses the certificate created by `ssl-cert` package which is great, but for production use you should consider updating that with a proper certificate from a recognised Certificate Authority (CA).

Finally, you should restart the PostgreSQL service to initialise the new configuration. From a terminal prompt enter the following to restart PostgreSQL:

```
sudo systemctl restart postgresql.service
```

> **Warning**:
> The above configuration is not complete by any means. Please refer to the **PostgreSQL Administrator's Guide <http://www.postgresql.org/docs/current/static/admin.html>** to configure more parameters.

You can test server connections from other machines by using the PostgreSQL client as follows, replacing the domain name with your actual server domain name or IP address:

```
sudo apt install postgresql-client
psql --host your-servers-dns-or-ip --username postgres --password --dbname template1
```

Streaming replication

PostgreSQL has a nice feature called **streaming replication** which provides the ability to continuously ship and apply the Write-Ahead Log **(WAL) XLOG <http://www.postgresql.org/docs/current/static/wal.html>** records to some number of standby servers to keep them current. Here is a simple way to replicate a PostgreSQL server (main) to a standby server.

First, create a replication user in the main server, to be used from the standby server:

```
sudo -u postgres createuser --replication -P -e replicator
```

Let's configure the main server to turn on the streaming replication. Open the file `/etc/postgresql/*/main/postgresql.conf` and make sure you have the following lines:

```
listen_addresses = '*'
wal_level = replica
```

Also edit the file `/etc/postgresql/*/main/pg_hba.conf` to add an extra line to allow the standby server connection for replication (that is a special keyword) using the `replicator` user:

```
host  replication   replicator   <IP address of the standby>      scram-sha-256
```

Restart the service to apply changes:

```
sudo systemctl restart postgresql
```

Now, in the standby server, let's stop the PostgreSQL service:

```
sudo systemctl stop postgresql
```

Edit the `/etc/postgresql/*/main/postgresql.conf` to set up hot standby:

```
hot_standby = on
```

Back up the current state of the main server (those commands are still issued on the standby system):

```
sudo su - postgres
# backup the current content of the standby server (update the version of your postgres accordingly)
cp -R /var/lib/postgresql/14/main /var/lib/postgresql/14/main_bak
# remove all the files in the data directory
rm -rf /var/lib/postgresql/14/main/*
pg_basebackup -h <IP address of the main server> -D /var/lib/postgresql/14/main -U replicator -P -v -R
```

After this, a full single pass will have been completed, copying the content of the main database onto the local system being the standby. In the `pg_basebackup` command the flags represent the following:

- `-h`: The hostname or IP address of the main server
- `-D`: The data directory
- `-U`: The user to be used in the operation
- `-P`: Turns on progress reporting
- `-v`: Enables verbose mode
- `-R`: Creates a `standby.signal` file and appends connection settings to `postgresql.auto.conf`

Finally, let's start the PostgreSQL service on standby server:

```
sudo systemctl start postgresql
```

To make sure it is working, go to the main server and run the following command:

```
sudo -u postgres psql -c "select * from pg_stat_replication;"
```

As mentioned, this is a very simple introduction, there are way more great details in the upstream documentation about the configuration of **replication <https://www.postgresql.org/docs/current/static/runtime-config-replication.html>** as well as further **High Availability, Load Balancing, and Replication <https://www.postgresql.org/docs/current/static/high-availability.html>** .

To test the replication you can now create a test database in the main server and check if it is replicated in the standby server:

```
sudo -u postgres createdb test # on the main server
sudo -u postgres psql -c "\l" # on the standby server
```

You need to be able to see the `test` database, that was created on the main server, in the standby server.

Backups

PostgreSQL databases should be backed up regularly. Refer to the **PostgreSQL Administrator's Guide <http://www.postgresql.org/docs/current/static/backup.html>** for different approaches.

Further reading

- As mentioned above, the **PostgreSQL Administrator's Guide <http://www.postgresql.org/docs/current/static/admin.html>** is an excellent resource. The guide is also available in the `postgresql-doc` package. Execute the following in a terminal to install the package:

```
sudo apt install postgresql-doc
```

This package provides further manpages on PostgreSQL “dblink” and “server programming interface” as well as the upstream HTML guide. To view the guide enter `xdg-open /usr/share/doc/postgresql-doc-*/html/index.html` or point your browser at it.

- For general SQL information see the O’Reilly books **Getting Started with SQL: A Hands-On Approach for Beginners <http://shop.oreilly.com/product/0636920044994.do>** by Thomas Nield as an entry point and **SQL in a Nutshell <http://shop.oreilly.com/product/9780596518851.do>** as a quick reference.

Previous MySQL Next Mail services: Install Postfix

This page was last modified 4 months ago. **Help improve this document in the forum <https://discourse.ubuntu.com/t/install-and-configure-postgresql/11516>** .