Boundless Server » Working with your data » Getting started with PostGIS »

Connecting to PostgreSQL on Linux for the first time



Next 🔾

Connecting to PostgreSQL on Linux for the first time

Note

This section uses the command line utility psql and optionally the graphical utility pgAdmin.

psql is included with the Boundless Server PostgreSQL package. pgAdmin is provided as part of Boundless Desktop.

on Linux, both on Ubuntu and Red Hat-based systems, the default PostgreSQL configuration has connections turned off for the postgres user by default.

So after install of Boundless Server, if you try to connect to PostgreSQL via the **psql** command-line utility or through **pgAdmin**, you will get the following connection error:

```
psql: FATAL: peer authentication failed for user "postgres"
```

There are two steps to allow connections to PostgreSQL:

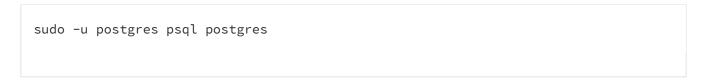
- Set a password for the postgres user
- Allow local connections to PostgreSQL

For more information, please see the Ubuntu documentation on PostgreSQL.

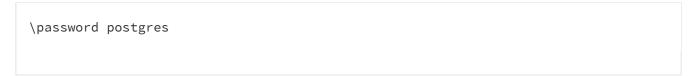
Setting a password for the postgres user

On Linux systems, there is no default password set.

1. Run the **psql** command from the **postgres** user account:



2. Set the password:



- 3. Enter a password.
- 4. Close **psql**.

\q

Allowing local connections

The file pg_hba.conf governs the basic constraints underlying connection to PostgreSQL. By default, these settings are very conservative. Specifically, local connections are not allowed for the postgres user.

To allow this:

- 1. As a super user, open /etc/postgresql/9.6/main/pg_hba.conf (Ubuntu) or /var/lib/pgsql/9.6/data/pg_hba.conf (Red Hat) in a text editor.
- 2. Scroll down to the line that describes local socket connections. It may look like this:



3. Change the peer method to md5.

Note

For more information on the various options, please see the PostgreSQL documentation on pg_hba.conf.

5/10/204.8 To allow connections using **pgAdmin**, Alhan Hennieum fer that the strict strict of the st

host	all	all	::1/128	ident

- 5. Change the ident method to md5.
- 6. Save and close the file.
- 7. Restart PostgreSQL:
 - Ubuntu:

```
sudo service postgresql restart
```

• Red Hat:

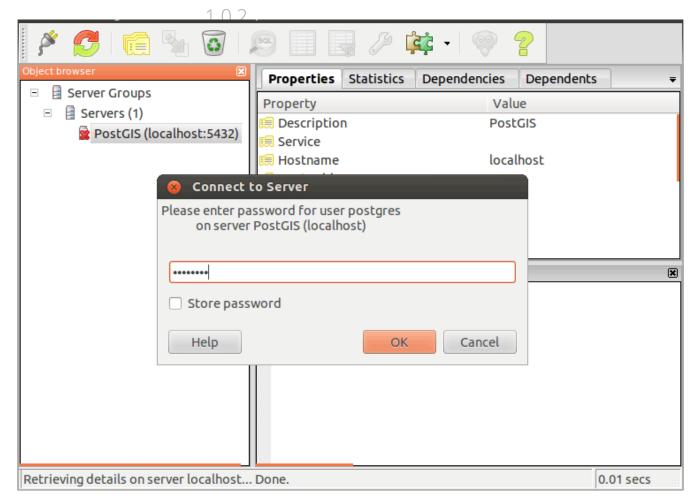
```
sudo service postgresql-9.6 restart
```

8. To test your connection using $\boldsymbol{psql},$ run the following command:

```
psql -U postgres -W
```

and enter your password when prompted. You should be able to access the **psql** console.

9. To test your connection using **pgAdmin**, connect to the database at localhost:5432 using the user name postgres and the password supplied.



Testing the connection in pgAdmin

If you encounter errors, make sure that the postgres password is set correctly, and that the correct line was edited in pg_hba.conf, as many look alike.

Allowing remote connections

Often the system running psql will be different from the system running the database. This is especially true if you want to run **pgAdmin** from your system.

In order to allow connections from remote systems, some slightly different configuration will be necessary.

The details are similar to that of allowing local connections, with some slight differences.

- 1. As a super user, open /etc/postgresql/9.6/main/pg_hba.conf (Ubuntu) or /var/lib/pgsql/9.6/data/pg_hba.conf (Red Hat) in a text editor.
- 2. Scroll down to the line that describes local socket connections. It may look like this:

3. Change to:

host all 0.0.0.0/0 trust

Warning

This is a potential security risk, and you may wish to customize this further. For more information on the various options, please see the PostgreSQL documentation on pg_hba.conf.

- 4. Save and close the file.
- 5. In the same directory, open postgresql.conf.
- 6. Under the section on **Connection Settings**), add or replace the line that starts with listen_addresses to respond to all requests:

```
listen_addresses = '*'
```

Note

Make sure the line is uncommented.

- 7. Save and close the file.
- 8. Restart PostgreSQL:
 - Ubuntu:

```
sudo service postgresql restart
```

Red Hat:

```
sudo service postgresql-9.6 restart
```

9. To test your connection using **pgAdmin**, connect to the database at the IP address or host name of the system that hosts the database. Enter the user name postgres and the password supplied.

• Note

Make sure that port 5432 is open on this system.