Install pgAdmin 4 for PostgreSQL Database Server on Ubuntu Linux

Author: Thomas Rakwach

Last Updated: Mon, Jun 28, 2021

PostgreSQL Server Apps System Admin Ubuntu

Using a Different System?

Introduction

pgAdmin is an open-source software project for administration and management of PostgreSQL database server. It includes a graphical administration interface, an SQL query tool, a procedural code debugger, and more other tools. This guide will take you through installing pgAdmin for PostgreSQL server on Ubuntu Linux.

Prerequisites

Deploy a fully updated Ubuntu Linux LTS server at Vultr with at least 2GB of RAM and 1 vCPU cores.

Create a non-root user with sudo access.

1. Install and Configure PostgreSQL Database Server

Import the repository PGP signing key for PostgreSQL.

```
$ wget --quiet -O - https://www.postgresql.org/medi
```

Add PostgreSQL APT repository.

```
$ sudo sh -c 'echo "deb http://apt.postgresql.org/p
```

Then, install PostgreSQL database server.

```
$ sudo apt install postgresql-12 postgresql-client-
```

Enable the database server to start automatically on a reboot.

```
$ sudo systemctl enable postgresql
```

Start the database server.

```
$ sudo systemctl start postgresql
```

Change the default PostgreSQL password.

```
$ sudo passwd postgres
```

Switch to the **postgres** user.

```
$ su - postgres
```

Create a new database user named **pgadmin**.

```
$ createuser pgadmin
```

Log in to the PostgreSQL instance.

```
$ psql
```

Set a secure password for the user **pgadmin** by changing the value of `secure_password` .

```
ALTER USER pgadmin WITH ENCRYPTED password 'secure_
```

Create a database named **testdb** and set the owner to **pgadmin**.

```
CREATE DATABASE testdb OWNER pgadmin;
```

Grant all the privileges on the **testdb** database to the user **pgadmin**.

```
GRANT ALL PRIVILEGES ON DATABASE testdb to pgadmin;
```

Exit PostgreSQL instance.

```
\q
```

Return to your non-root sudo user account.

```
$ exit
```

2. Change PostgreSQL Configurations

Default PostgreSQL config only allow connection to localhost (or 127.0.0.1) interface. We should edit the config files **postgresql.conf** and **pg_hba.conf** to allow for remote connection. These files are located in the `/etc/postgresql/*/main` directory.

Open the file **postgresql.conf**.

```
$ sudo nano /etc/postgresql/*/main/postgresql.conf
```

Find the following line.

```
#listen_addresses = 'localhost'
```

To listen to all IP addresses, change the line to the below code and save the file.

```
listen_addresses = '*'
```

Open the file **pg_hba.conf**.

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

At the end of the file add the following lines and save the file.

```
host    all             all             0.0.0.0/0
host    all             all             ::/0
```

Restart the PostgreSQL Database Server service to ensure the changes are saved.

```
$ sudo service postgresql restart
```

3. Install pgAdmin 4

Install all required dependencies for pgAdmin 4 installation.

```
$ sudo apt-get install curl gnupg2 -y
```

Import the repository PGP signing key for pgAdmin.

```
$ sudo curl https://www.pgadmin.org/static/packages
```

Add pgAdmin 4 APT repository.

```
$ sudo sh -c 'echo "deb https://ftp.postgresql.org/
```

Install the web-based version of pgAdmin 4 tool for your platform.

```
$ sudo apt install pgadmin4-web -y
```

Configure the web mode. You'll be required to enter an email address and a password for later login.

```
$ sudo /usr/pgadmin4/bin/setup-web.sh
```

For UFW firewall configured, allow http or https traffic depending on what you are using.

```
$ sudo ufw allow http
$ sudo ufw allow https
```

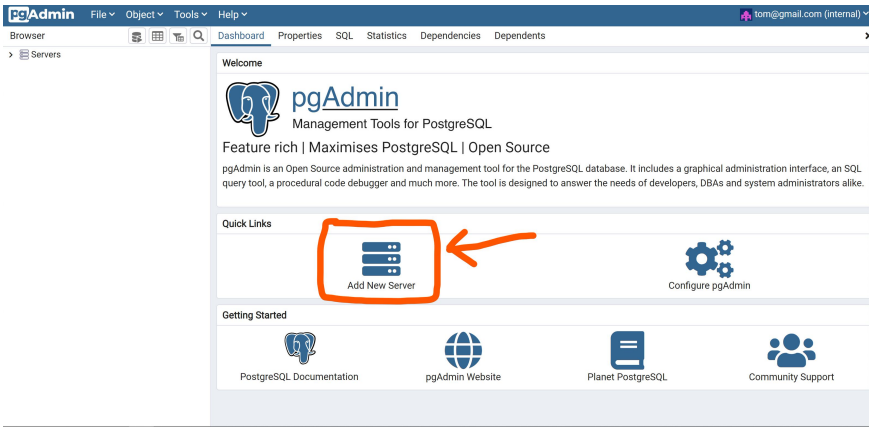
After installation, open your browser and go to `http://ServerIP/pgadmin4` to access the pgAdmin 4 user interface. For example:

```
http://192.0.2.48/pgadmin4
```

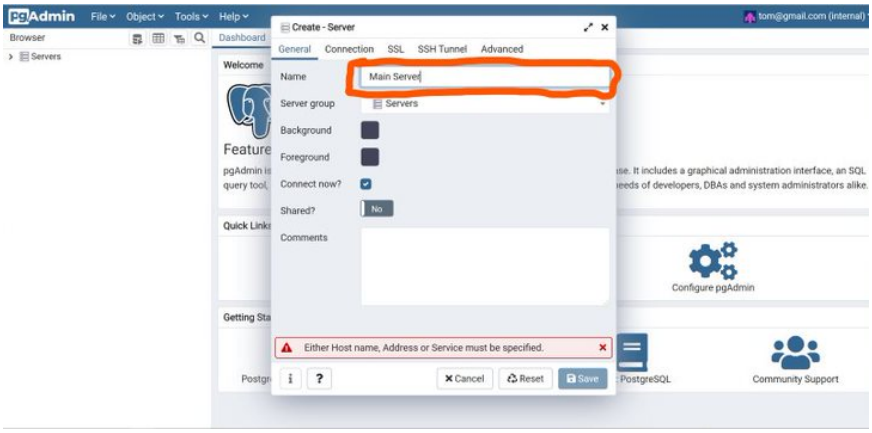
Login with the **email** address and **password** that you set during pgAdmin 4 web mode configuration.

4. Connect to Database Server on pgAdmin 4

On the web interface of your preinstalled pgAdmin 4, locate the dashboard and click, **Add New Server**, you'll get a dialogue box where you'll enter your preferred name for that server.



Enter the preferred name of the server in the **Name** input section.

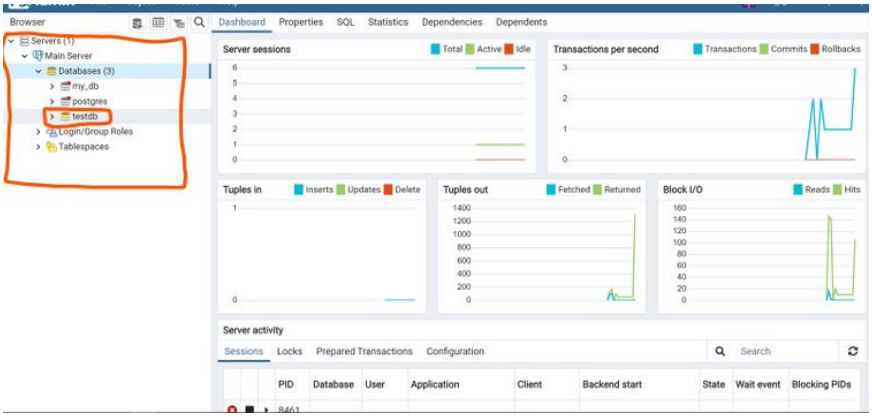


On the pop up dialogue box, switch to the **Connection** tab and enter the required fields with the appropriate data.

1. Enter your **IP address** under the **Host name/address**.
2. Leave **Port** value as **5432**.
3. Enter `testdb` under **Maintenance database**.
4. Enter `pgadmin` under **Username**.
5. Enter `secure_password` as password. Modify the `secure_password` value with the actual value used in Step 1.
6. Click **Save** to login to the server.

After the connection is successful, go to the upper left corner of the dashboard, and you will see your server.

browse through by expanding the arrows to access your database **testdb**.



Conclusion

You have now successfully browsed to the database. You'll be able to see the database server metrics on the main dashboard on graphs. You can then begin to fully manage the server and add more databases.