**INSTALLATION OF POSTGRESQL DB on EC2 INSTANCE & PGADMIN CLIENT**

This Blog aims to get up and running a fully functional platform with PostgreSQL installed on UBUNTU ec2 machine + pgAdmin working on your local machine. Let’s briefly describe these elements.

PostgreSQL is available in all Ubuntu versions by default. However, Ubuntu "snapshots" a specific version of PostgreSQL that is then supported throughout the lifetime of that Ubuntu version. Other versions of PostgreSQL are available through the PostgreSQL apt repository.

**PostgreSQL Apt Repository**

If the version included in your version of Ubuntu is not the one you want, you can use the [**PostgreSQL Apt Repository**](https://apt.postgresql.org/). This repository will integrate with your normal systems and patch management, and provide automatic updates for all supported versions of PostgreSQL throughout the support [**lifetime**](https://www.postgresql.org/support/versioning/) of PostgreSQL.

**Environment**

This step by step tutorial is written based on the following requirements:

* **Operating System**: UBUNTU OS AMI AWS EC2

The PostgreSQL Apt Repository supports the current versions of Ubuntu:

* jammy (22.04, LTS)

on the following architectures:

* arm64 (18.04 and newer; LTS releases only)

**Objectives**

This Blog aims to get up and running a fully functional platform with PostgreSQL + pgAdmin working on your local machine with the help of docker. Let’s briefly describe these elements.

**PostgreSQL:**

PostgreSQL, often referred to as "Postgres," is an open-source relational database management system (RDBMS) with a primary objective of providing a powerful, reliable, and extensible platform for data storage and management. Its core objectives include data integrity, high performance, scalability, and support for advanced SQL features. PostgreSQL aims to maintain data consistency through the implementation of the ACID properties (Atomicity, Consistency, Isolation, Durability) and offers support for complex data types, indexing, and foreign key constraints. Its commitment to open-source principles means it is available for free, making it accessible to a wide range of users, from small-scale projects to large enterprises. The PostgreSQL community actively maintains and enhances the system, ensuring ongoing development and feature improvements.

**pgAdmin:**

pgAdmin is an open-source administration and management tool designed to assist users in effectively interacting with PostgreSQL databases. Its primary objectives revolve around simplifying database administration, improving developer productivity, and enhancing the management of PostgreSQL instances. With a user-friendly graphical interface, pgAdmin enables users to perform tasks like creating, modifying, and querying databases, monitoring server performance, and managing user access with ease. Its support for SQL queries, visual query builders, and database object exploration simplifies database development and maintenance. Additionally, it promotes efficient collaboration among database administrators and developers through features such as version control integration and the ability to manage multiple database connections. Overall, pgAdmin aims to streamline PostgreSQL database management and make it accessible to users at all levels of expertise.

Installation of Postgresql Database

<https://www.postgresql.org/download/linux/ubuntu/>

a. Create the file repository configuration:

# sudo sh -c 'echo "deb https://apt.postgresql.org/pub/repos/apt $(lsb\_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'

b. Import the repository signing key:

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

c. Update the package lists:

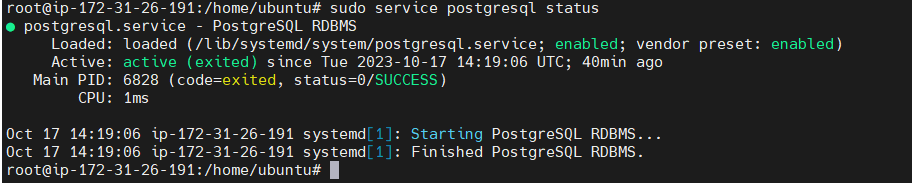
# sudo apt-get update

d. Install the latest version of PostgreSQL.

e. If you want a specific version, use 'postgresql-12' or similar instead of 'postgresql':

# sudo apt-get -y install postgresql-15

# sudo service postgresql status



Restart your postgresql service with below command

# sudo service postgresql restart

**Start using postgreSQL in Terminal**

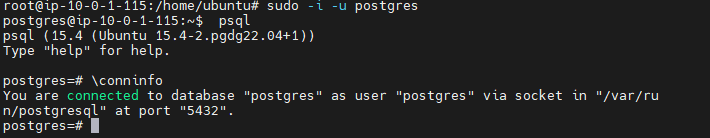
Enter postgre cli by changing user

# sudo -i -u postgres

# psql

Check Postgres sql connection

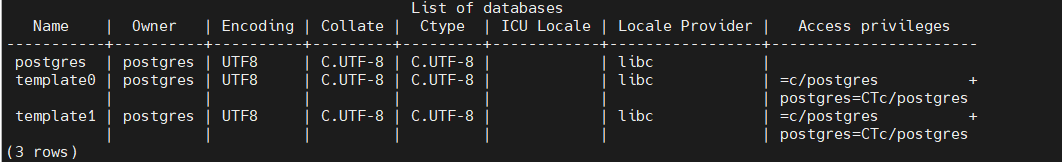
# \conninfo



Check List of Roles & DB Access

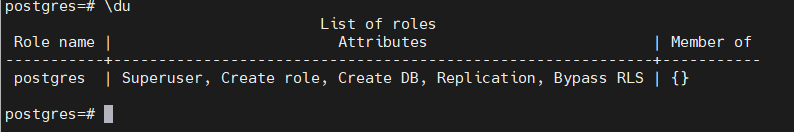
1. Check List **of** users of postgres

# \l



2. Check role(user) and Permission.

# \du



3. use \q to exit from postgres



**Follow below Steps to configure postgresql remote access**

**Creating user & password for Postgresql**

“postgres” admin user don’t have password been assigned, so we need to set a new password to postgres user by ourself.

Setup password access to postgres user

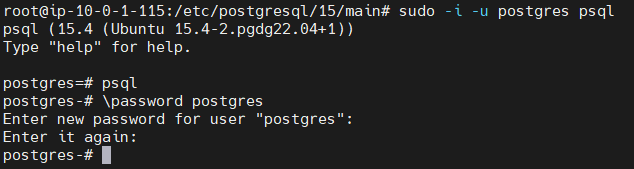
Enter below command, to enter psql cli to interact with postgres database engine.

Switch to postgres user

# sudo -i -u postgres

# psql

Create a password for the user & reenter password



**Restart PostgreSQL:** After making these changes, restart PostgreSQL to apply the new settings.

# sudo systemctl restart postgresql

# psql -c "ALTER USER postgres WITH PASSWORD 'postgres';"

If trouble to create user & password use this below command as well, The command **psql -c "ALTER USER postgres WITH PASSWORD 'postgres';"** is used to change the password of the PostgreSQL user named "postgres" to "postgres."

**Configuration files of Postgresql**

1. **Edit postgresql.conf file & change listen address**

When postgresql is been installed, psql allows listening to localhost connections only, & thus block all the connection request from remote TCP/IP connection.

Change directory .

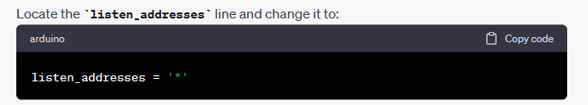




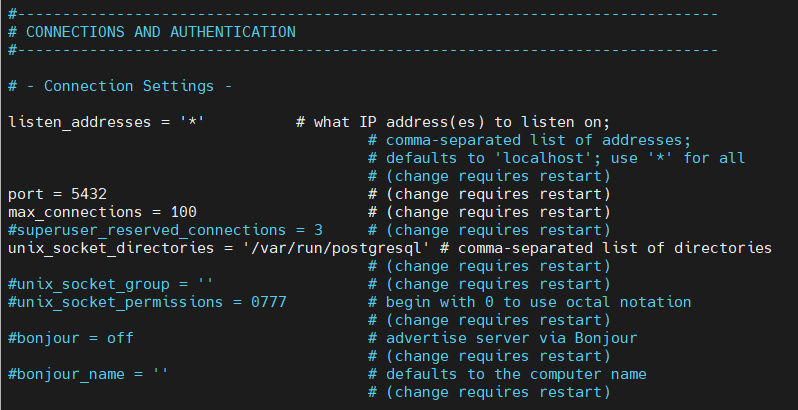
**Open postgresql.conf file**

# nano /etc/postgresql/15/main/postgresql.conf

Note \* Uncomment the # and Replace local host with \*



Exit with save



1. **Edit postgres pg\_hba.conf for remote host access for all IP**

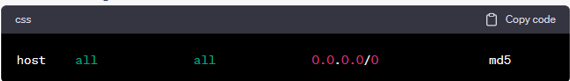
By default, psql connection is only accepted for localhost & it refuses all the request from TCP/IP remote connection. Therefore, we need to allow a user to login from any IP address.



Open **pg\_hba.conf** file & add this 2 lines at bottom of the file.

Next, edit the **pg\_hba.conf** file to allow connections from all IP addresses:

# sudo nano /etc/postgresql/15/ main/pg\_hba.conf



**Edit below IPV4 & IPV6 host connections & method with md5 auth**



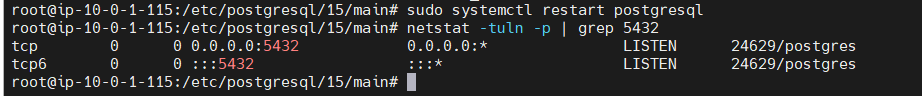
**Restart PostgreSQL:** After making these changes, restart PostgreSQL to apply the new settings.

# sudo systemctl restart postgresql

Check postgresql listening post number

# netstat -tuln -p **|** grep 5432

This should give response as running on global 5432, i.e. 0.0.0.0:5432. This means that psql is now enabled for remote access.



**Important Note:**Make sure you have defined **InBound rules port range 5432** on your cloud service instance security for remote access postgres



**Access *pgAdmin***

**CLIENT APP connect to remote connection access locally**

pgAdmin is an open-source administration and management tool designed to assist users in effectively interacting with PostgreSQL databases. Its primary objectives revolve around simplifying database administration, improving developer productivity, and enhancing the management of PostgreSQL instances. With a user-friendly graphical interface, pgAdmin enables users to perform tasks like creating, modifying, and querying databases, monitoring server performance, and managing user access with ease.

Accessing pgAdmin typically involves a few steps. Here's a basic guide on how to access pgAdmin:

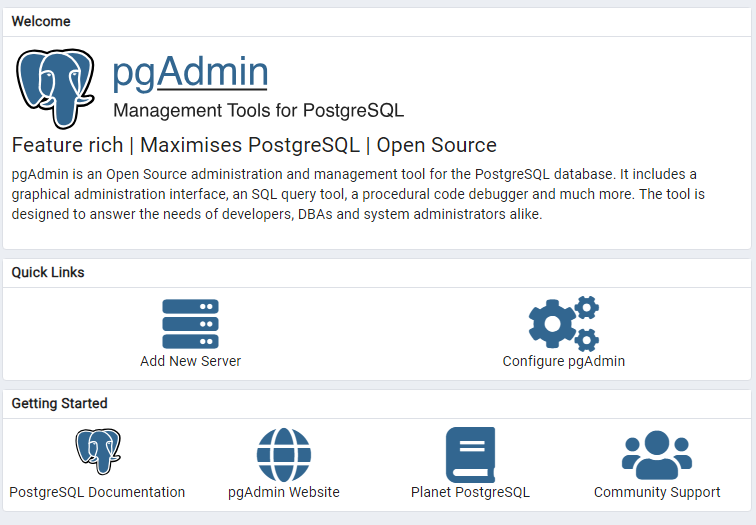
**Installation**:

* + Ensure that you have pgAdmin installed on your computer. If it's not installed, you can download it from the official website and follow the installation instructions for your specific operating system.

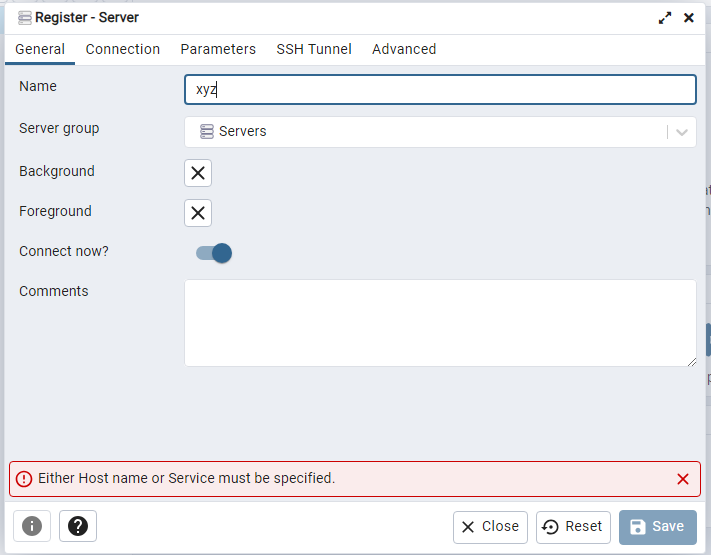
**Launch pgAdmin**:

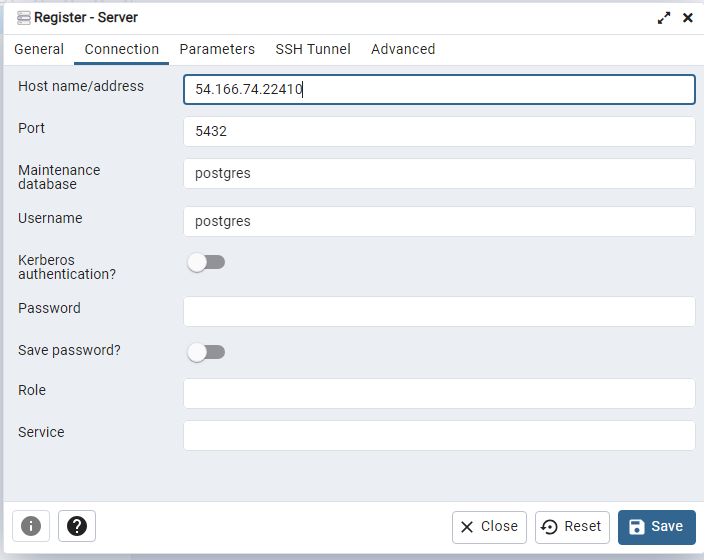
* + Once installed, open the pgAdmin application. You can usually find it in your system's applications or programs menu, or you may have created a shortcut on your desktop during installation.

**Connect to a Server**:



* + In pgAdmin, you'll need to connect to a PostgreSQL server. Here's how:
    - Click the "Add New Server" icon (usually a green plus sign) or select "Object" > "Create" > "Server."
    - In the "General" tab of the "Create - Server" dialog, give your server a name in the "Name" field.





* + - In the "Connection" tab, provide the connection details for your PostgreSQL server . Note to create elastic ip & associate with this server.
    - Click "Save" or "OK" to connect to the server.

**Accessing Databases**:

* + Once you've successfully connected to the server, you'll see it listed in the Servers panel on the left. Expand the server to see its databases.
  + You can access databases by expanding the server, then the "Databases" node, and selecting the database you want to work with.

**Using pgAdmin**:

* + With the database selected, you can perform various tasks, such as creating tables, running SQL queries, managing users, and more, using the pgAdmin interface.

**Conclusion**

In this post we have demonstrated how easy can be to get your *postgreSQL* + *pgAdmin* platform up and running using AWS ec2 Ubuntu os .

References

* <https://www.postgresql.org/download/linux/ubuntu/>
* <https://github.com/Aseemakram19/Postgressql-Pgadmin-on-EC2-server>
* [(1) cloud devops with aseem - YouTube](https://www.youtube.com/@clouddevopswithaseem)