

How to Access a PostgreSQL Database From Another Host?

written by sysadmin | 22 November 2025

I want to access a PostgreSQL database from another host.

Problem

How to access a PostgreSQL database from another host?

Solution

I have 2 servers, the PostgreSQL server IP is 192.168.56.101, and the Ubuntu server IP is 192.168.56.11. I want to access the PostgreSQL database from the Ubuntu server. By default, use the format below if you want to access PostgreSQL:

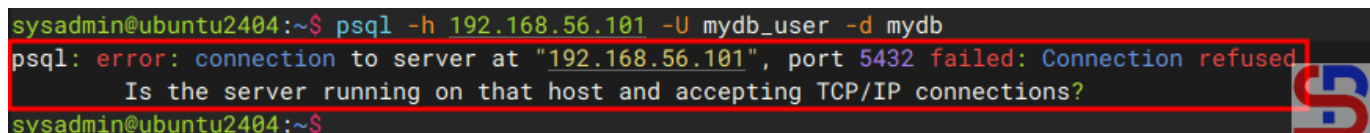
```
psql -h ip_db -u username -d db_name -p port_number
```

By default, the PostgreSQL database uses port 5432, so if you access your PostgreSQL using port 5432, you don't need to write the port. I accessed my PostgreSQL by writing the command:

```
psql -h 192.168.56.101 -u postgres -d mydb
```

But I got an error like the image below:

```
psql: error: connection to server at '192.168.56.101', port 5432 failed:
Connection refused
Is the server running on that host and accepting TCP/IP connections?
```

A terminal window screenshot showing a command being executed and the resulting error message. The command is 'psql -h 192.168.56.101 -U mydb_user -d mydb'. The error message is 'psql: error: connection to server at "192.168.56.101", port 5432 failed: Connection refused. Is the server running on that host and accepting TCP/IP connections?'. The terminal prompt is 'sysadmin@ubuntu2404:~\$'.

```
sysadmin@ubuntu2404:~$ psql -h 192.168.56.101 -U mydb_user -d mydb
psql: error: connection to server at "192.168.56.101", port 5432 failed: Connection refused
Is the server running on that host and accepting TCP/IP connections?
sysadmin@ubuntu2404:~$
```

Error when accessing the PostgreSQL database

The following are the steps to access a PostgreSQL database from another host:

1. Open port

Open port 5432 on each server. If you use RockyLinux/AlmaLinux/RHEL for your servers, use the command below:

```
firewall-cmd --add-port=5432/tcp --permanent  
firewall-cmd --reload
```

But if you use Ubuntu/Debian for your server, type the command below:

```
sudo ufw allow 5432
```

2. Config postgresql.conf

In the database server, go to **/etc/postgresql/18/main/postgresql.conf** file if you use PostgreSQL 18, but back up the file first:

```
sudo cp /etc/postgresql/18/main/postgresql.conf  
/etc/postgresql/18/main/postgresql.conf.ori
```

After that, change the script below in the file:

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

to

```
listen_addresses = '*'
```

3. Configure pg_hba.conf

After that, go to **/etc/postgresql/18/main/pg_hba.conf** if you use PostgreSQL 18, but backup the file first:

```
sudo cp /etc/postgresql/18/main/pg_hba.conf  
/etc/postgresql/18/main/pg_hba.conf.ori
```

Then, add your IP host (in this case, IP is 192.168.56.11) like the below script:

```
host    all             all             192.168.56.11/32      scram-sha-256
```

Warning

You must change the file `/etc/postgresql/18/main/postgresql.conf` if you want to use scram-sha-256 encryption as described in [this article](#).

4. Restart PostgreSQL

Restart postgresql service using the command below:

```
sudo systemctl restart postgresql
```

Now, try to access the PostgreSQL again, and you should be able to access the database like the command below:

```
sysadmin@docker:~$ psql -h localhost -U mydb_user -d mydb
Password for user mydb_user:
psql (18.0 (Ubuntu 18.0-1.pgdg24.04+3))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off, ALPN: postgresql)
Type "help" for help.

mydb=> INSERT INTO employee (name,age,city) VALUES ('queen',23,'Florida');
INSERT 0 1
mydb=> select * from employee;
 name | age | city
-----+-----+-----
 bob  |  21 | New York
 John |  22 | Chicago
 steve | 22  | Kansas
 trump |  20 | Orlando
 paul  |  20 | Texas
 queen |  23 | Florida
(6 rows)
```

Do CRUD on the database

Note

If you have an error when you want to display data, like the image below:

```
ERROR: permission denied for table employee
```

```
sysadmin@ubuntu2404:~$ psql -h 192.168.56.101 -U mydb_user -d mydb
Password for user mydb_user:
psql (18.0 (Ubuntu 18.0-1.pgdg24.04+3))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off, ALPN: postgresql)
Type "help" for help.

mydb=> SELECT * FROM employee;
ERROR: permission denied for table employee
mydb=>
```

Error when querying in the postgresql

Then you have to change the owner of the database and the tables in the database. You can follow how to change it via [this page](#).

References

dev.to

dba.stackexchange.com

prisma.io

tigerdata.com