# Postgres: Troubleshooting Remote Access

Here's steps to work through, to ensure remote access to a PostgreSQL instance.

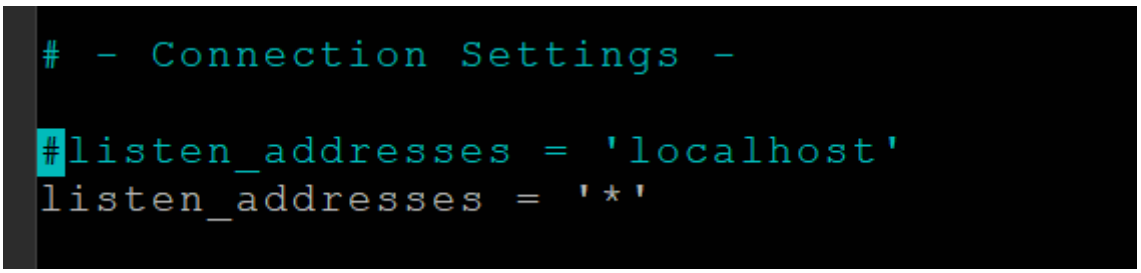
## Check PostgreSQL is Listening on All Interfaces

From a terminal, open the postgresql.conf file, with this:

```
sudo nano /etc/postgresql/<version>/main/postgresql.conf
```

> NOTE: Replace <version> with the version you installed. Will be '16' as of this writing.

Locate the line with 'listen_addresses' and ensure it is set to listen on all adapters.
Or, to listen on a specific one for the host.




It is commented out, by default.
So, you will need to open up the listener to at least one adapter, or, all, like above.

Save and close the config file, and restart the postgres instance with this:

```
sudo systemctl restart postgresql
```

## Allowed Source Subnets

Open the pg_hba.conf file, to verify allowed source subnets and protocols, with this:

```
sudo nano /etc/postgresql/<version>/main/pg_hba.conf
```

>

> NOTE: Same as before, replace <version> with the installed version. '16' as of this writing.

Make sure the allowed sources includes the subnet of your source hosts.

For this, you may need to add lines for each allowed subnet, like this:

```
host    all             all             192.168.1.0/24          md5
```

```
# DO NOT DISABLE!
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local   all             postgres                                peer

# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             127.0.0.1/32            scram-sha-256
# IPv6 local connections:
host    all             all             ::1/128                 scram-sha-256
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     peer
host    replication     all             127.0.0.1/32            scram-sha-256
host    replication     all             ::1/128                 scram-sha-256

host    all             all             192.168.1.0/24          md5
host    all             all             192.168.60.0/24         md5
host    all             all             192.168.70.0/24         md5
host    all             all             192.168.110.0/24        md5
host    all             all             192.168.120.0/24        md5
host    all             all             192.168.150.0/24        md5
host    all             all             192.168.160.0/24        md5
```

The above example allows access from several subnets.

Save and close the config.

And, reload it, with this:

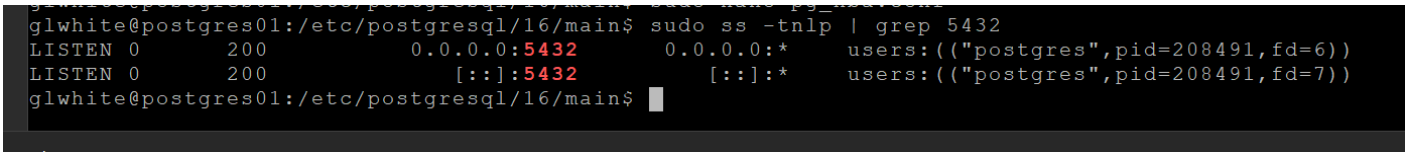
```
sudo systemctl reload postgresql
```

# Ensure Correct Listening Port

Run this on the Postgres host, to see what ports the database engine is listening on:

```
sudo ss -tnlp | grep 5432
```

You will see something like this for a default install:




> NOTE: By default, the engine listens on 5432.

## Firewall Ingress Rule

Make sure the host firewall allows incoming connections, with this:

```
sudo ufw allow 5432/tcp
```

## Test from Remote Client

Now, verify connectivity from your remote client.

---

Revision #1
Created 29 July 2025 14:59:23 by glwhite
Updated 29 July 2025 15:13:40 by glwhite