**HowTo Safely Open a PostgreSQL Port for Remote Access**

<http://www.project-open.com/en/howto-postgresql-port-secure-remote-access>

This document describes how to open a PostgreSQL database port for remote access and includes security considerations for both Linux and Windows.

**Why do you Need to Open a Port**

You need to open a remote port if you want to access data in the database from a client running on a different computer in the network. The same is true, if you want to use [pgAdmin](https://www.pgadmin.org/) for or similar tools for SQL development.

**Check for Port IP**

Please check if PostgreSQL is listening on a public port:

Linux:

# netstat -nlp | grep 5432

tcp 0 0 127.0.0.1:5432 0.0.0.0:\* LISTEN 1272/postgres

tcp6 0 0 ::1:5432 :::\* LISTEN 1272/postgres

Windows:

C:\>netstat -a | grep 5432

TCP 127.0.0.1:5432 tarraco:0 LISTENING

TCP [::1]:5432 tarraco:0 LISTENING

The results above (including "127.0.0.1:5432") shows that PostgreSQL is listening only for connects originating from the local computer, so we will have to edit the "postgresql.conf" configuration file. A result including "0.0.0.0:5432" indicates that PostgreSQL is already listening for remote connections.

You can usually ignore the line with "::1", as it refers to the IP v6 protocol, which is rarely used.

**Edit postgresql.conf**

This file is usually located in /var/lib/pgsql/data/ on Linux or C:\PostgreSQL\data\ on Windows or similar.

In this file we will edit the "listen\_address" and "port" parameters, so that they look like below:

#------------------------------------------------------------------------------

# CONNECTIONS AND AUTHENTICATION

#------------------------------------------------------------------------------

# - Connection Settings -

listen\_addresses = '0.0.0.0' # what IP address(es) to listen on;

# comma-separated list of addresses;

# defaults to 'localhost'; use '\*' for all

# (change requires restart)

port = 5432 # (change requires restart)

**Restart PostgreSQL**

After that we need to restart PG to activate the changes.

On recent Linux system you have to enter as user root:

systemctl restart postgresql.service

On Windows you can use Control Panel -> Administrative Tools -> Services and restart the PostgreSQL service. For ]po[, the PostgreSQL service is called "]po[ PostgreSQL".

Repeating the "Check for Port IP" step above, you should now see that the port IP is "0.0.0.0", meaning that it will accept connections from any remote computer.

**Authentication Configuration using pg\_hba.conf**

pg\_hba.conf is located in the same directly as postgresql.conf.

Please  add the following two lines at the end of the file:

host all all 0.0.0.0/0 md5

host all all ::/0 md5

This means that remote access is allowed using IP v4 and IP v6 to all databases and all users using the "md5" authentication protocol.

Please "restart postgresql" again.

**Open Linux Firewall Port**

Does your PostgreSQL database run on a Linux server with the firewall enabled (like the [[https://sourceforge.net/projects/project-open/files/project-open/V5.0/|CentOS 7 ]project-open[ virtual appliance]])?

In this case you will have to poke a hole (as root):

# firewall-cmd --zone=public --add-port=5432/tcp --permanent

success

# firewall-cmd --reload

success

**Open Windows Firewall Port**

Does your PostgreSQL database run on a Windows server with firewall enabled?

In this case you can just turn off the firewall for a first test in Control Panel -> Systems and Security -> Windows Firewall -> Turn Windows Firewall on or off.

As an alternative you can go to Control Panel -> Systems and Security ->  Windows Firewall -> Allow a program or feature through Windows Firewall -> Advanced Settings -> New Rule:

* Rule Type: Port
* TCP or UDP: TCP
* Specific local ports: 5432
* Action: Allow the connection
* When does this rule apply: Domain, Private and Public (all three checked)
* Name: "PostgreSQL Incoming"

**Setting PostgreSQL Passwords**

In order to set the PostgreSQL password for the user "projop" (or whatever user...) you need to connect locally to the database using an account with administration rights.

In the ]project-open[ Centos 7 virtual appliance just enter "psql" in a local terminal window running as user "projop".

Then please issue the following SQL command:

alter user projop with password 'secret';