

PostgreSQL Tutorial

Installation

A better way to install postgresQL is by [PostgreSQL.org](https://www.postgresql.org) , and instructed by documentation, select your system and the version of postgresQL that you want to install in your machine, and then install it by yum repository with following command,

```
yum install url_for_postgreSQL
```

then, install the client packages,

```
yum install postgresql12
```

Note: it is assumed the version of postgresQL is 12.1

Optionally install the server packages,

```
yum install postgresql12-server
```

Optionally initialize the database and enable automatic start,

```
service postgresql-12 initdb  
chkconfig postgresql-12 on  
service postgresql-12 start
```

Due to policies for Red Hat family distributions, the PostgreSQL installation will not be enabled for automatic start or have the database initialized automatically. To make your database installation complete, we need to perform the following steps for CentOS 6.X:

```
service postgresql initdb  
chkconfig postgresql on
```

Now, we have finished installation on our machine.

Management

Change your role into postgres, and execute `psql`, for the default authentication method is peer and the name of default database is postgres,

```
sudo su - postgres  
psql
```

Then we are in the database postgres with default administrator postgres, now we are able to configure our database. There are some syntax whereby we can manage users and database of PostgreSQL.

Create User

To create user foo1 with password foo111,

```
CREATE USER foo1 WITH PASSWORD 'foo111';
```

Note: every command must be ended with semicolon

Set/Alter Password For User

To alter the password of user foo1 into 'foo112',

```
ALTER USER foo1 PASSWORD 'foo112';
```

For user foo1 to change its own password,

```
# Enter database first, and then input  
$ \password
```

Create Database

To create database foo1db for user foo1,

```
CREATE DATABASE foo1db OWNER foo1;
```

and grant all privileges of foo1db to foo1,

```
GRANT ALL PRIVILEGES ON DATABASE foo1db TO foo1;
```

Drop Database

To delete database foo1db,

```
DROP DATABASE fooldb;
```

Note: users(except postgres) are allowed to delete database that they own.

Drop Users

To delete user foo1,

```
DROPUSER foo1;
```

These are some basic commands to manage database.

Remote Connection Configuration

In this section, we will show how to set postgresQL so that we can connect postgresQL server from other machines.

Setting for File pg_hba.conf

In default directory, which is in `/var/lib/pgsql/12/data` for postgresql-12, there are some configuration files. we need to make some changes to *pg_hba.conf*.

	TYPE	DATABASE	USER	ADDRESS	METHOD
-	local	all	all	0.0.0.0/0	peer
+	host	all	all	0.0.0.0/0	md5

And then we are able to login in database by password which is encrypted in md5.

Note: we can also set specific user from specific address login into specific database, more information are shown in pg_hba.conf.

Setting for File postgresql.conf

Again in default directory, some changes need to be made for *postgresql.conf*.

```
+ listen_addresses = '*'
```

And we can also set port in the file.

Setting for Firewall Rules

For whatever reason, the default port of postgresSQL, i.e. port=5432, could be stopped, and then we need to open that port as follows,

- find firewall configuration file at `/etc/sysconfig/iptables`
- modify firewall rules to allow port 5432 to be listened

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 5432 -j ACCEPT
```

- restart iptables service

```
service iptables restart
```

Other ports are set in same way.

PostgreSQL Syntax

There are many good websites to learn postgresSQL syntax, e.g. runoob.com.

Lucky for you []\~(¯▽¯)\~*