Postgresql
Installation and Configuration

Postgres Installation and Configuration

Ø **Installation Steps (By TAR FILE):**

First Unzip the Postgres zip file then follow the following steps:

```
gunzip postgresql-7.4.2.tar.gz
tar xf postgresql-7.4.2.tar
```

This will create a directory postgresql-7.4.2 under the current directory with the PostgreSQL sources.

```
./configure
gmake or make
su
gmake install or make install
adduser postgres
mkdir /usr/local/pgsql/data
chown postgres /usr/local/pgsql/data
su - postgres
/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data
/usr/local/pgsql/bin/postmaster -D /usr/local/pgsql/data >logfile 2>&1 &
/usr/local/pgsql/bin/createdb test
/usr/local/pgsql/bin/psql test
```

Ø **Configuration Steps:**

1) Change pg_hba.conf file:

File path is /var/lib/pgsql/data/

Client authentication is controlled by the file pg_hba.conf in the data directory, The general format of the pg_hba.conf file is a set of records, one per line. Blank lines are ignored, as is any text after the "#" comment character. A record is made up of a number of fields which are separated by spaces and/or tabs. Fields can contain white space if the field value is quoted. Records cannot be continued across lines.

Each record specifies a connection type, a client IP address range (if relevant for the connection type), a database name, a user name, and the authentication method to be used for connections matching these parameters. The first record with a matching connection type, client address, requested database, and user name is used to perform authentication. There is no "fall-through" or "backup": if one record is chosen and the authentication fails, subsequent records are not considered. If no record matches, access is denied.

A record may have one of the three formats

local   *database* *user* *authentication-method* [*authentication-option*]
host   *database* *user* *IP-address* *IP-mask* *authentication-method*
[*authentication-option*]
hostssl *database* *user* *IP-address* *IP-mask* *authentication-method*
[*authentication-option*

The meaning of the fields is as follows:

local

This record matches connection attempts using Unix domain sockets. Without a record of this type, Unix-domain socket connections are disallowed

host

This record matches connection attempts using TCP/IP networks. Note that TCP/IP connections are disabled unless the server is started with the -i option or the tcpip_socket postgresql.conf configuration parameter is enabled.

hostssl

This record matches connection attempts using SSL over TCP/IP. host records will match either SSL or non-SSL connection attempts, but hostssl records require SSL connections.

To be able make use of this option the server must be built with SSL support enabled. Furthermore, SSL must be enabled by enabling the option ssl in postgresql.conf

database

Specifies which databases this record matches. The value all specifies that it matches all databases

user

Specifies which PostgreSQL users this record matches. The value all specifies that it matches all users. Otherwise, this is the name of a specific PostgreSQL user.

IP-address
IP-mask

These two fields contain IP address/mask values in standard dotted decimal notation. (IP addresses can only be specified numerically, not as domain or host names.) Taken together they specify the client machine IP addresses that this record matches. The precise logic is that (*actual-IP-address* xor *IP-address-field*) and *IP-mask-field* must be zero for the record to match. (Of course IP addresses can be spoofed but this consideration is beyond the scope of PostgreSQL.)

These fields only apply to host and hostssl records.

authentication-method

Specifies the authentication method to use when connecting via this record.

trust

The connection is allowed unconditionally.

reject

The connection is rejected unconditionally

md5

Requires the client to supply an MD5 encrypted password for authentication

password

Same as "md5", but the password is sent in clear text over the network

The pg_hba.conf file is read on start-up and when the postmaster receives a SIGHUP signal. If you edit the file on an active system, you will need to signal the postmaster (using pg_ctl reload or kill -HUP) to make it re-read the file.

Example of file

```
# Allow any user on the local system to connect to any database under
 # any user name using Unix-domain sockets (the default for local
 # connections).
 #
 # TYPE  DATABASE    USER    IP-ADDRESS        IP-MASK           METHOD
 local   all         all                                          trust

 # The same using local loopback TCP/IP connections.
 #
 # TYPE  DATABASE    USER    IP-ADDRESS        IP-MASK           METHOD
 host    all         all     127.0.0.1         255.255.255.255   trust
 host    all         all     10.11.13.0        255.255.255.0     trust
 host    all         all     202.157.79.51     255.255.255.255   trust

 # Allow any user from any host with IP address 192.168.93.x to connect
 # to database "template1" as the same user name that ident reports for
 # the connection (typically the Unix user name).
 #
```

```
# TYPE  DATABASE    USER    IP-ADDRESS        IP-MASK           METHOD
host    template1   all    192.168.93.0   255.255.255.0  identsameuser

host    all         all         202.157.72.250   255.255.255.255   trust
host    all         all         202.157.79.34     255.255.255.255   trust
```

2)   Change postgresql.conf file:

File path is same as above file

Change the following Parameter as given below:

```
   tcpip_socket          = true
   max_connections       = 100 (editable as required)
   port                  = 5432 (default editable )
   shared_buffer         = 200 (editable as required)
   sort_mem              = 1024         # min 64, size in KB
   vacuum_mem            = 4096         # min 1024, size in KB
```

this file consists of lines of the form : name = value

the ‘=’ is optional .white space may be used. comments are introduced with # any where in line .The complete list of option name and allowed values can be found in the postgresql documentation .The commented-out setting shown in this file represent the default values.

3) start postgres server with following command

```
/usr/local/pgsql/bin/initdb – D <new database path>
                                    -- with postgres user
or

/usr/local/pgsql/bin/postmaster – I – D <new database path>
                                   -- with postgres user
```

4) creating user/groups accounts on linux

```
su  root
                              --login as root  on linux

useradd  <username(OS)>
                              --create user on linux

passwd <password>
                              --assign password to user
```