NOTE
FOR
POSTGRESQL
INSTALLATION
&
CONFIGURATION
(Sourcecode)

Session Contents

- Source Installation
- Initializing a PostgreSQL Cluster
- Starting PostgreSQL Cluster
- Common Issues & Troubleshooting

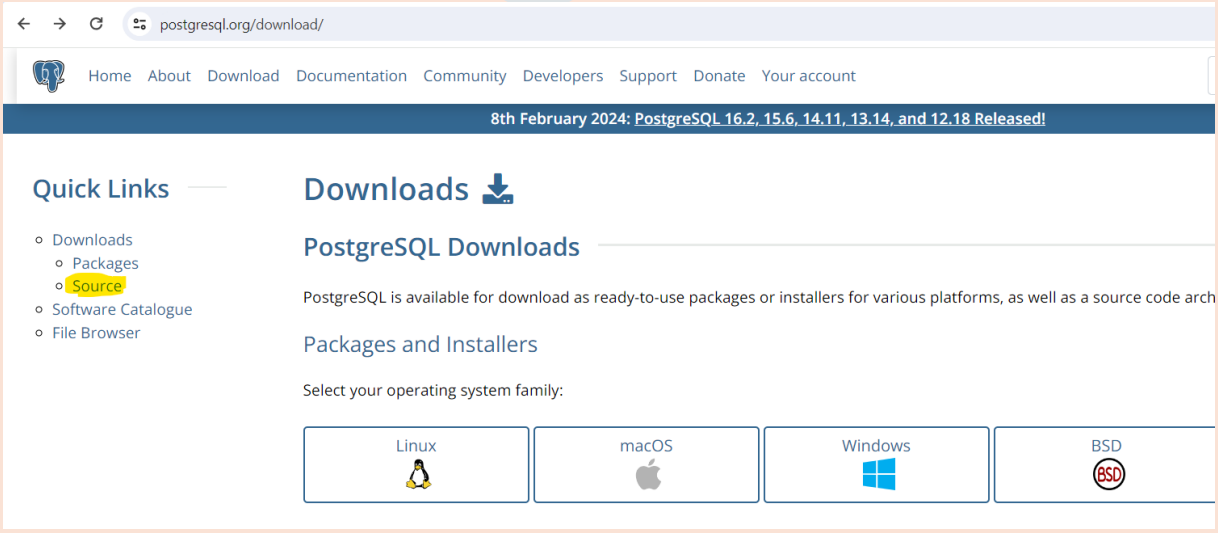
PostgreSQL Download & Installation Steps for Linux (Redhat/CentOS)

Installation Type	Sourcecode
IP address	192.168.47.160
Environment	RHEL 9
Memory	2gb
PostgreSQL Version	15.2
Data_direrctory	/PostgreSQL_base/data/
Bin_directory	/PostgreSQL_base/bin/

Step -1

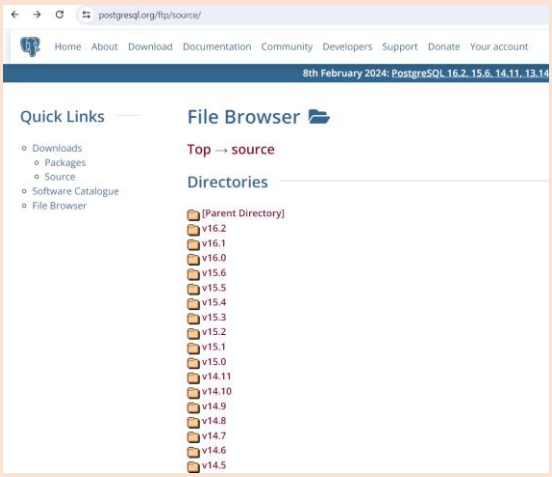
Go to postgressql.org/download/

Select source from the left corner.



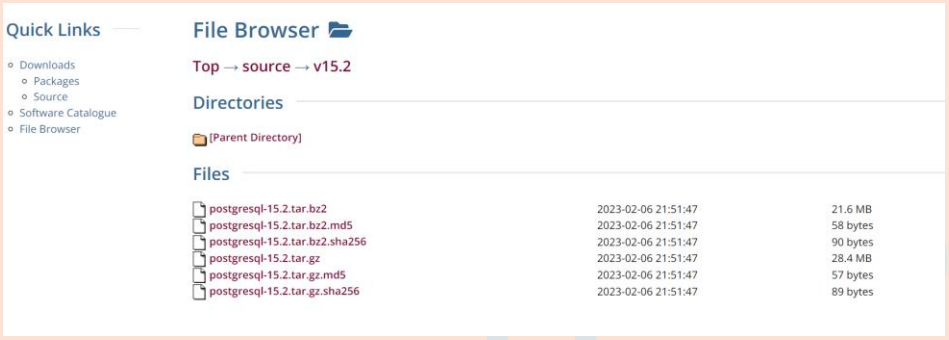
Step -2

Select the required version



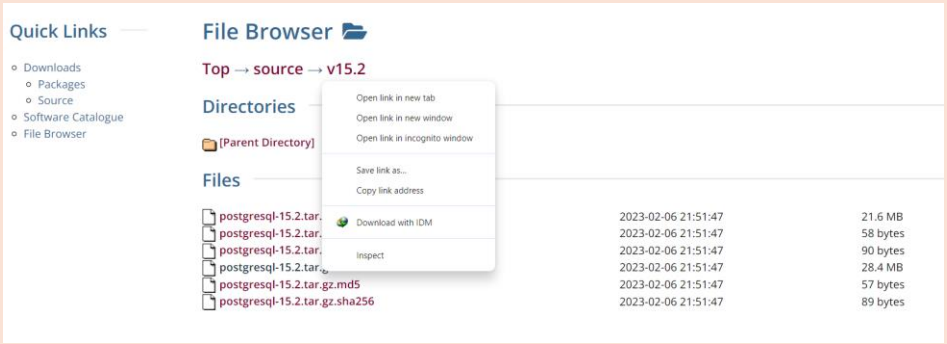
Step -3

Select the required tar.gz file



Step -4

Right click on the file select copy link address



Step -5

Connect the server, download the file using wget on the server

[root@localhost download] # wget https://ftp.postgresql.org/pub/source/v15.2/postgresql-15.2.tar.gz

```
[root@localhost download]# wget https://ftp.postgresql.org/pub/source/v15.2/postgresql-15.2.tar.gz
--2024-03-12 12:32:35--  https://ftp.postgresql.org/pub/source/v15.2/postgresql-15.2.tar.gz
Resolving ftp.postgresql.org (ftp.postgresql.org)... 147.75.85.69, 217.196.149.55, 72.32.157.246, ...
Connecting to ftp.postgresql.org (ftp.postgresql.org)|147.75.85.69|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 29811750 (28M) [application/octet-stream]
Saving to: 'postgresql-15.2.tar.gz'

postgresql-15.2.tar.gz                      100%[=========================================================================================>]  28.43M  1.96MB/s    in 17s

2024-03-12 12:32:53 (1.68 MB/s) - 'postgresql-15.2.tar.gz' saved [29811750/29811750]

[root@localhost download]#
```

Step -6

Untar and unzip the download file

[root@localhost download]# tar -xvf postgresql-16.1.tar.gz

Will get the below output

```
-rw-r--r--. 1 root root 29811750 Feb  7  2023 postgresql-15.2.tar.gz
[root@localhost download]#
[root@localhost download]# tar -xvf postgresql-15.2.tar.gz
```

Step -7

Post untar postgresql-15.1 directory will appear

```
postgresql-15.2/doc/src/sgml/stylesheet.xsl
postgresql-15.2/doc/src/sgml/contrib.sgml
postgresql-15.2/doc/src/sgml/man-stamp
postgresql-15.2/doc/src/sgml/basic-archive.sgml
postgresql-15.2/doc/src/sgml/bki.sgml
postgresql-15.2/doc/src/sgml/unaccent.sgml
postgresql-15.2/doc/src/sgml/btree.sgml
postgresql-15.2/doc/src/sgml/external-projects.sgml
postgresql-15.2/doc/src/sgml/problems.sgml
postgresql-15.2/doc/src/sgml/arch-dev.sgml
postgresql-15.2/doc/src/sgml/pageinspect.sgml
postgresql-15.2/doc/src/sgml/info.sgml
postgresql-15.2/doc/src/sgml/installation.sgml
postgresql-15.2/doc/src/sgml/xplang.sgml
postgresql-15.2/doc/src/Makefile
postgresql-15.2/doc/KNOWN_BUGS
postgresql-15.2/doc/Makefile
postgresql-15.2/doc/TODO
postgresql-15.2/doc/MISSING_FEATURES
postgresql-15.2/.cirrus.yml
postgresql-15.2/HISTORY
postgresql-15.2/Makefile
postgresql-15.2/README
postgresql-15.2/COPYRIGHT
postgresql-15.2/GNUmakefile.in
postgresql-15.2/.gitattributes
postgresql-15.2/aclocal.m4
postgresql-15.2/INSTALL
```

Change to postgresql-15.1 directory by using

[root@localhost download]# cd postgresql-15.1

```
[root@localhost download]# ls -lrth
total 29M
drwxrwxrwx. 6 1107 1107 4.0K Feb  7  2023 postgresql-15.2
-rw-r--r--. 1 root root  29M Feb  7  2023 postgresql-15.2.tar.gz
[root@localhost download]#
[root@localhost download]# cd postgresql-15.2
```

Step -8

Run the below query to start installation

[root@localhost postgresql-15.2]# ./configure --prefix /PostgreSQL_base/bin/

Note: Configure: Navigate to the directory where you extract the source code and run the configure script. This script checks your system for dependencies

```
[root@localhost yum.repos.d]# ./configure --prefix /PostgreSQL_base/bin/
-bash: ./configure: No such file or directory
[root@localhost yum.repos.d]# cd /PostgreSQL_base/download/postgresql-15.2/
[root@localhost postgresql-15.2]# ./configure --prefix /PostgreSQL_base/bin/
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking which template to use... linux
checking whether NLS is wanted... no
checking for default port number... 5432
checking for block size... 8kB
checking for segment size... 1GB
checking for WAL block size... 8kB
```

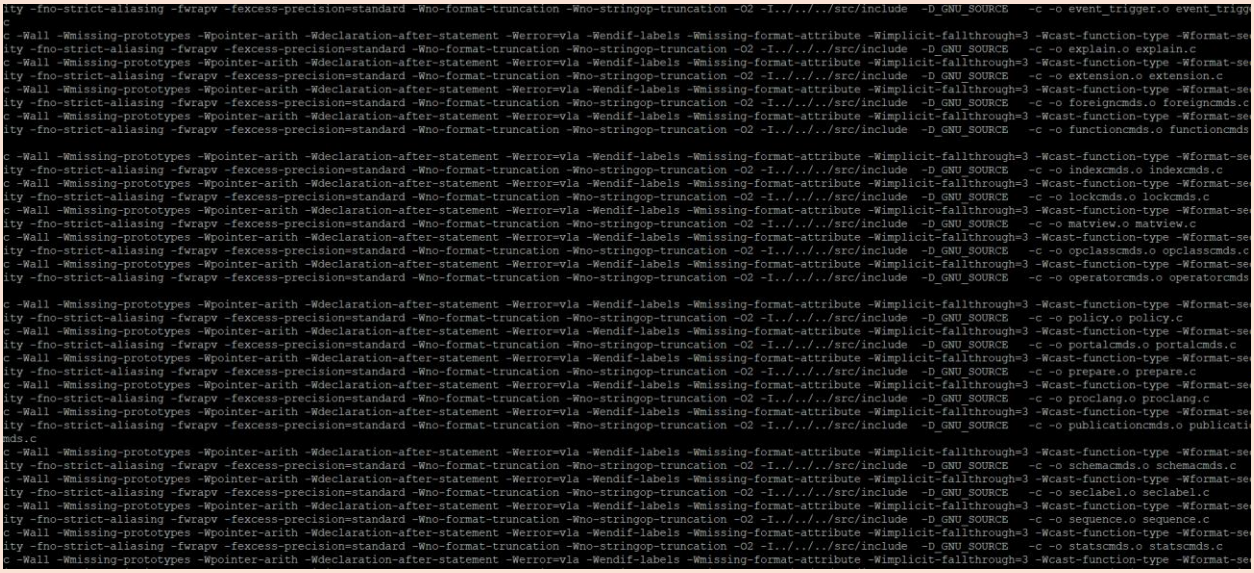
```
checking for dbtoepub... no
checking whether gcc supports -Wl,--as-needed... yes
configure: using compiler=gcc (GCC) 11.3.1 20221121 (Red Hat 11.3.1-4)
configure: using CFLAGS=-Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-afte
on-type -Wformat-security -fno-strict-aliasing -fwrapv -fexcess-precision=standard -W
configure: using CPPFLAGS= -D_GNU_SOURCE
configure: using LDFLAGS=  -Wl,--as-needed
configure: creating ./config.status
config.status: creating GNUmakefile
config.status: creating src/Makefile.global
config.status: creating src/include/pg_config.h
config.status: creating src/include/pg_config_ext.h
config.status: creating src/interfaces/ecpg/include/ecpg_config.h
config.status: linking src/backend/port/tas/dummy.s to src/backend/port/tas.s
config.status: linking src/backend/port/posix_sema.c to src/backend/port/pg_sema.c
config.status: linking src/backend/port/sysv_shmem.c to src/backend/port/pg_shmem.c
config.status: linking src/include/port/linux.h to src/include/pg_config_os.h
config.status: linking src/makefiles/Makefile.linux to src/Makefile.port
[root@localhost postgresql-15.2]#
```

Step -9

run the below two command

[root@localhost postgresql-15.2]# make world

[root@localhost postgresql-15.2]# make world install



```
make[3]: Entering directory '/PostgreSQL_base/download/postgresql-15.2/src/interfaces/libpq'
make -C ../../../src/port all
make[4]: Entering directory '/PostgreSQL_base/download/postgresql-15.2/src/port'
make[4]: Nothing to be done for 'all'.
make[4]: Leaving directory '/PostgreSQL_base/download/postgresql-15.2/src/port'
make -C ../../../src/common all
make[4]: Entering directory '/PostgreSQL_base/download/postgresql-15.2/src/common'
make[4]: Nothing to be done for 'all'.
make[4]: Leaving directory '/PostgreSQL_base/download/postgresql-15.2/src/common'
make[3]: Leaving directory '/PostgreSQL_base/download/postgresql-15.2/src/interfaces/libpq'
make -C ../../../src/port all
make[3]: Entering directory '/PostgreSQL_base/download/postgresql-15.2/src/port'
make[3]: Nothing to be done for 'all'.
make[3]: Leaving directory '/PostgreSQL_base/download/postgresql-15.2/src/port'
make -C ../../../src/common all
make[3]: Entering directory '/PostgreSQL_base/download/postgresql-15.2/src/common'
make[3]: Nothing to be done for 'all'.
make[3]: Leaving directory '/PostgreSQL_base/download/postgresql-15.2/src/common'
/usr/bin/mkdir -p '/PostgreSQL_base/bin/lib/postgresql/pgxs/src/test/isolation'
/usr/bin/install -c  pg_isolation_regress '/PostgreSQL_base/bin/lib/postgresql/pgxs/src/test/isolation/pg_isolation_regress'
/usr/bin/install -c  isolationtester '/PostgreSQL_base/bin/lib/postgresql/pgxs/src/test/isolation/isolationtester'
make[2]: Leaving directory '/PostgreSQL_base/download/postgresql-15.2/src/test/isolation'
make -C test/perl install
make[2]: Entering directory '/PostgreSQL_base/download/postgresql-15.2/src/test/perl'
make[2]: Nothing to be done for 'install'.
make[2]: Leaving directory '/PostgreSQL_base/download/postgresql-15.2/src/test/perl'
/usr/bin/mkdir -p '/PostgreSQL_base/bin/lib/postgresql/pgxs/src'
/usr/bin/install -c -m 644 Makefile.global '/PostgreSQL_base/bin/lib/postgresql/pgxs/src/Makefile.global'
/usr/bin/install -c -m 644 Makefile.port '/PostgreSQL_base/bin/lib/postgresql/pgxs/src/Makefile.port'
/usr/bin/install -c -m 644 ./Makefile.shlib '/PostgreSQL_base/bin/lib/postgresql/pgxs/src/Makefile.shlib'
/usr/bin/install -c -m 644 ./nls-global.mk '/PostgreSQL_base/bin/lib/postgresql/pgxs/src/nls-global.mk'
make[1]: Leaving directory '/PostgreSQL_base/download/postgresql-15.2/src'
make -C config install
make[1]: Entering directory '/PostgreSQL_base/download/postgresql-15.2/config'
/usr/bin/mkdir -p '/PostgreSQL_base/bin/lib/postgresql/pgxs/config'
/usr/bin/install -c -m 755 ./install-sh '/PostgreSQL_base/bin/lib/postgresql/pgxs/config/install-sh'
/usr/bin/install -c -m 755 ./missing '/PostgreSQL_base/bin/lib/postgresql/pgxs/config/missing'
make[1]: Leaving directory '/PostgreSQL_base/download/postgresql-15.2/config'
[root@localhost postgresql-15.2]#
```

It will take 10 to 15 minutes

Step -10

Provide ownership and permission of data directory to postgres user

To create a postgres user

[root@localhost PostgreSQL_base]# adduser postgres

[root@localhost PostgreSQL_base]# passwd postgres (set the password)

Note: Generally, OS team provide the postgres user.

Then provide ownership and permission

[root@localhost PostgreSQL_base]# chmod 700 /PostgreSQL_base/

[root@localhost PostgreSQL_base]# chown -R postgres:postgres /PostgreSQL_base/

```
[root@localhost PostgreSQL_base]# pwd
/PostgreSQL_base
[root@localhost PostgreSQL_base]# chmod 700 /PostgreSQL_base/
[root@localhost PostgreSQL_base]# chown -R postgres:postgres /PostgreSQL_base/
[root@localhost PostgreSQL_base]# ls -lrth
total 0
drwx------. 2 postgres postgres  6 Mar 12 11:33 data
drwxr-xr-x. 2 postgres postgres  6 Mar 12 11:33 backup
drwxr-xr-x. 3 postgres postgres 59 Mar 12 12:38 download
drwxr-xr-x. 6 postgres postgres 56 Mar 12 13:37 bin
```

Step -11

Initialize database cluster & Start DB service

- Switch to postgres user by su – postgres

```
[root@localhost PostgreSQL_base]# su - postgres
[postgres@localhost ~]$
```

- Run the initialization command from the bin directory
  [postgres@localhost bin]$ ./initdb -D /PostgreSQL_base/data/

```
/PostgreSQL_base/bin/bin
[postgres@localhost bin]$ ./initdb -D /PostgreSQL_base/data/
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "en_IN.UTF-8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

fixing permissions on existing directory /PostgreSQL_base/data ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... Asia/Kolkata
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok
```

- Start the DB services

[postgres@localhost bin]$ ./pg_ctl -D /PostgreSQL_base/data/ -l logfile start

```
[postgres@localhost bin]$  ./pg_ctl -D /PostgreSQL_base/data/ -l logfile start
waiting for server to start.... done
server started
[postgres@localhost bin]$
```

Step -12

Log in to PostgreSQL console

[postgres@localhost bin]$ /PostgreSQL_base/bin/bin/psql

```
[postgres@localhost bin]$ /PostgreSQL_base/bin/bin/psql
psql (15.2)
Type "help" for help.

postgres=#
```

Common Error and Troubleshooting

Error

```
checking for library containing setproctitle... no
checking for library containing dlsym... none required
checking for library containing socket... none required
checking for library containing shl_load... no
checking for library containing getopt_long... none required
checking for library containing shm_open... none required
checking for library containing shm_unlink... none required
checking for library containing clock_gettime... none required
checking for library containing fdatasync... none required
checking for library containing shmget... none required
checking for library containing backtrace_symbols... none required
checking for library containing gethostbyname_r... none required
checking for library containing pthread_barrier_wait... none required
checking for library containing readline... no
configure: error: readline library not found
If you have readline already installed, see config.log for details on the
failure.  It is possible the compiler isn't looking in the proper directory.
Use --without-readline to disable readline support.
```

Troubleshooting

Run the below command to resolve the readline library issue

[root@localhost postgresql-15.2]# yum install -y readline-devel

Some Parameter Configuration

View Configuration

- SHOW ALL

- SELECT current_setting(parameter_name);

```
postgres=# show all;
              name                 |              setting              |                                                description
-----------------------------------+-----------------------------------+-----------------------------------------------------------------------------------------------------------
 allow_in_place_tablespaces        | off                               | Allows tablespaces directly inside pg_tblspc, for testing.
 allow_system_table_mods           | off                               | Allows modifications of the structure of system tables.
 application_name                  | psql                              | Sets the application name to be reported in statistics and logs.
 archive_cleanup_command           |                                   | Sets the shell command that will be executed at every restart point.
 archive_command                   | (disabled)                        | Sets the shell command that will be called to archive a WAL file.
 archive_library                   |                                   | Sets the library that will be called to archive a WAL file.
 archive_mode                      | off                               | Allows archiving of WAL files using archive_command.
 archive_timeout                   | 0                                 | Sets the amount of time to wait before forcing a switch to the next WAL file.
 array_nulls                       | on                                | Enable input of NULL elements in arrays.
 authentication_timeout            | 1min                              | Sets the maximum allowed time to complete client authentication.
 autovacuum                        | on                                | Starts the autovacuum subprocess.
 autovacuum_analyze_scale_factor   | 0.1                               | Number of tuple inserts, updates, or deletes prior to analyze as a fraction of reltuples.
 autovacuum_analyze_threshold      | 50                                | Minimum number of tuple inserts, updates, or deletes prior to analyze.
 autovacuum_freeze_max_age         | 200000000                         | Age at which to autovacuum a table to prevent transaction ID wraparound.
 autovacuum_max_workers            | 3                                 | Sets the maximum number of simultaneously running autovacuum worker processes.
 autovacuum_multixact_freeze_max_age | 400000000                       | Multixact age at which to autovacuum a table to prevent multixact wraparound.
 autovacuum_naptime                | 1min                              | Time to sleep between autovacuum runs.
 autovacuum_vacuum_cost_delay      | 2ms                               | Vacuum cost delay in milliseconds, for autovacuum.
 autovacuum_vacuum_cost_limit      | -1                                | Vacuum cost amount available before napping, for autovacuum.
 autovacuum_vacuum_insert_scale_factor | 0.2                           | Number of tuple inserts prior to vacuum as a fraction of reltuples.
 autovacuum_vacuum_insert_threshold | 1000                             | Minimum number of tuple inserts prior to vacuum, or -1 to disable insert vacuums.
 autovacuum_vacuum_scale_factor    | 0.2                               | Number of tuple updates or deletes prior to vacuum as a fraction of reltuples.
 autovacuum_vacuum_threshold       | 50                                | Minimum number of tuple updates or deletes prior to vacuum.
 autovacuum_work_mem               | -1                                | Sets the maximum memory to be used by each autovacuum worker process.
 backend_flush_after               | 0                                 | Number of pages after which previously performed writes are flushed to disk.
 backslash_quote                   | safe_encoding                     | Sets whether "\'" is allowed in string literals.
 backtrace_functions               |                                   | Log backtrace for errors in these functions.
 bgwriter_delay                    | 200ms                             | Background writer sleep time between rounds.
 bgwriter_flush_after              | 512kB                             | Number of pages after which previously performed writes are flushed to disk.
 bgwriter_lru_maxpages             | 100                               | Background writer maximum number of LRU pages to flush per round.
 bgwriter_lru_multiplier           | 2                                 | Multiple of the average buffer usage to free per round.
 block_size                        | 8192                              | Shows the size of a disk block
```

Common Configuration Options

- max_connections 100 (default value)
- shared_buffers (usually 25 to 40 % of the RAM) – 512MB
- effective_cache_size (usually RAM – shared_buffers) – 1536MB
- maintenance_work_mem (5 – 10 % of RAM) – 128MB
- work_mem (per connection memory) = ROUND (Total – above allocated – OS required + some buffer) / max_connection. – 1310KB
- min_wal_size – 1GB
- max_wal_size – 4GB
- wal_keep_size – 100
- listen_addresses – '*'

There are many other parameters that need to be reconfigure according to the requirement.

The pg_hba.conf File

Client authentication is controlled by a configuration file, which traditionally is named pg_hba.conf and is stored in the database cluster's data directory. (HBA stands for host-based authentication.) A default pg_hba.conf file is installed when the data directory is initialized by **initdb**. It is possible to place the authentication configuration file elsewhere, however; see the **hba_file** configuration parameter.

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 can be continued onto the next line by ending the line with a backslash. (Backslashes are not special except at the end of a line.) 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 double-quoted. Quoting one of the keywords in a database, user, or address field (e.g., all or replication) makes the word lose its special meaning, and just match a database,

user, or host with that name. Backslash line continuation applies even within quoted text or comments.

Each authentication 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.

Each record can be an include directive or an authentication record. Include directives specify files that can be included, that contain additional records. The records will be inserted in place of the include directives. Include directives only contain two fields: include, include_if_exists or include_dir directive and the file or directory to be included. The file or directory can be a relative or absolute path, and can be double-quoted. For the include_dir form, all files not starting with a . and ending with .conf will be included. Multiple files within an include directory are processed in file name order (according to C locale rules, i.e., numbers before letters, and uppercase letters before lowercase ones).

A record can have several formats:

```
local             database  user  auth-method [auth-options]
host              database  user  address     auth-method  [auth-options]
hostssl           database  user  address     auth-method  [auth-options]
hostnossl         database  user  address     auth-method  [auth-options]
hostgssenc        database  user  address     auth-method  [auth-options]
hostnogssenc      database  user  address     auth-method  [auth-options]
host              database  user  IP-address  IP-mask      auth-method  [auth-options]
hostssl           database  user  IP-address  IP-mask      auth-method  [auth-options]
hostnossl         database  user  IP-address  IP-mask      auth-method  [auth-options]
hostgssenc        database  user  IP-address  IP-mask      auth-method  [auth-options]
hostnogssenc      database  user  IP-address  IP-mask      auth-method  [auth-options]
include           file
include_if_exists file
include_dir       directory
```

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 made using TCP/IP. host records match SSL or non-SSL connection attempts as well as GSSAPI encrypted or non-GSSAPI encrypted connection attempts.

Note

Remote TCP/IP connections will not be possible unless the server is started with an appropriate value for the **listen_addresses** configuration parameter, since the default behavior is to listen for TCP/IP connections only on the local loopback address localhost.

hostssl

This record matches connection attempts made using TCP/IP, but only when the connection is made with SSL encryption.

To make use of this option the server must be built with SSL support. Furthermore, SSL must be enabled by setting the **ssl** configuration parameter (see **Section 19.9** for more information). Otherwise, the hostssl record is ignored except for logging a warning that it cannot match any connections.

hostnossl

This record type has the opposite behavior of hostssl; it only matches connection attempts made over TCP/IP that do not use SSL.

hostgssenc

This record matches connection attempts made using TCP/IP, but only when the connection is made with GSSAPI encryption.

To make use of this option the server must be built with GSSAPI support. Otherwise, the hostgssenc record is ignored except for logging a warning that it cannot match any connections.

hostnogssenc

This record type has the opposite behavior of hostgssenc; it only matches connection attempts made over TCP/IP that do not use GSSAPI encryption.

database

Specifies which database name(s) this record matches. The value all specifies that it matches all databases. The value sameuser specifies that the record matches if the requested database has the same name as the requested user. The value samerole specifies that the requested user must be a member of the role with the same name as the requested database. (samegroup is an obsolete but still accepted spelling of samerole.) Superusers are

not considered to be members of a role for the purposes of samerole unless they are explicitly members of the role, directly or indirectly, and not just by virtue of being a superuser. The value replication specifies that the record matches if a physical replication connection is requested, however, it doesn't match with logical replication connections. Note that physical replication connections do not specify any particular database whereas logical replication connections do specify it. Otherwise, this is the name of a specific PostgreSQL database or a regular expression. Multiple database names and/or regular expressions can be supplied by separating them with commas.

If the database name starts with a slash (/), the remainder of the name is treated as a regular expression. (See Section 9.7.3.1 for details of PostgreSQL's regular expression syntax.)

A separate file containing database names and/or regular expressions can be specified by preceding the file name with @.

user

Specifies which database user name(s) this record matches. The value all specifies that it matches all users. Otherwise, this is either the name of a specific database user, a regular expression (when starting with a slash (/), or a group name preceded by +. (Recall that there is no real distinction between users and groups in PostgreSQL; a + mark really means “match any of the roles that are directly or indirectly members of this role”, while a name without a + mark matches only that specific role.) For this purpose, a superuser is only considered to be a member of a role if they are explicitly a member of the role, directly or indirectly, and not just by virtue of being a superuser. Multiple user names and/or regular expressions can be supplied by separating them with commas.

If the user name starts with a slash (/), the remainder of the name is treated as a regular expression. (See Section 9.7.3.1 for details of PostgreSQL's regular expression syntax.)

A separate file containing user names and/or regular expressions can be specified by preceding the file name with @.

address

Specifies the client machine address(es) that this record matches. This field can contain either a host name, an IP address range, or one of the special key words mentioned below.

An IP address range is specified using standard numeric notation for the range's starting address, then a slash (/) and a CIDR mask length. The mask length indicates the number of high-order bits of the client IP address that must match. Bits to the right of this should be zero in the given IP address. There must not be any white space between the IP address, the /, and the CIDR mask length.

Typical examples of an IPv4 address range specified this way are 172.20.143.89/32 for a single host, or 172.20.143.0/24 for a small network, or 10.6.0.0/16 for a larger one. An

IPv6 address range might look like ::1/128 for a single host (in this case the IPv6 loopback address) or fe80::7a31:c1ff:0000:0000/96 for a small network. 0.0.0.0/0 represents all IPv4 addresses, and ::0/0 represents all IPv6 addresses. To specify a single host, use a mask length of 32 for IPv4 or 128 for IPv6. In a network address, do not omit trailing zeroes.

An entry given in IPv4 format will match only IPv4 connections, and an entry given in IPv6 format will match only IPv6 connections, even if the represented address is in the IPv4-in-IPv6 range.

You can also write all to match any IP address, samehost to match any of the server's own IP addresses, or samenet to match any address in any subnet that the server is directly connected to.

If a host name is specified (anything that is not an IP address range or a special key word is treated as a host name), that name is compared with the result of a reverse name resolution of the client's IP address (e.g., reverse DNS lookup, if DNS is used). Host name comparisons are case insensitive. If there is a match, then a forward name resolution (e.g., forward DNS lookup) is performed on the host name to check whether any of the addresses it resolves to are equal to the client's IP address. If both directions match, then the entry is considered to match. (The host name that is used in pg_hba.conf should be the one that address-to-name resolution of the client's IP address returns, otherwise the line won't be matched. Some host name databases allow associating an IP address with multiple host names, but the operating system will only return one host name when asked to resolve an IP address.)

A host name specification that starts with a dot (.) matches a suffix of the actual host name. So .example.com would match foo.example.com (but not just example.com).

When host names are specified in pg_hba.conf, you should make sure that name resolution is reasonably fast. It can be of advantage to set up a local name resolution cache such as nscd. Also, you may wish to enable the configuration parameter log_hostname to see the client's host name instead of the IP address in the log.

These fields do not apply to local records.

IP-address
IP-mask

These two fields can be used as an alternative to the **IP-address/mask-length** notation. Instead of specifying the mask length, the actual mask is specified in a separate column. For example, `255.0.0.0` represents an IPv4 CIDR mask length of 8, and `255.255.255.255` represents a CIDR mask length of 32.

These fields do not apply to `local` records.

auth-method

Specifies the authentication method to use when a connection matches this record. The possible choices are summarized here; details are in **Section 21.3**. All the options are lower case and treated case sensitively, so even acronyms like `ldap` must be specified as lower case.

trust

Allow the connection unconditionally. This method allows anyone that can connect to the PostgreSQL database server to login as any PostgreSQL user they wish, without the need for a password or any other authentication. See Section 21.4 for details.

reject

Reject the connection unconditionally. This is useful for "filtering out" certain hosts from a group, for example a reject line could block a specific host from connecting, while a later line allows the remaining hosts in a specific network to connect.

scram-sha-256

Perform SCRAM-SHA-256 authentication to verify the user's password. See Section 21.5 for details.

md5

Perform SCRAM-SHA-256 or MD5 authentication to verify the user's password. See Section 21.5 for details.

password

Require the client to supply an unencrypted password for authentication. Since the password is sent in clear text over the network, this should not be used on untrusted networks. See Section 21.5 for details.

gss

Use GSSAPI to authenticate the user. This is only available for TCP/IP connections. See Section 21.6 for details. It can be used in conjunction with GSSAPI encryption.

SSPI

Use SSPI to authenticate the user. This is only available on Windows. See Section 21.7 for details.

ident

Obtain the operating system user name of the client by contacting the ident server on the client and check if it matches the requested database user name. Ident authentication can only be used on TCP/IP connections. When specified for local connections, peer authentication will be used instead. See Section 21.8 for details.

peer

Obtain the client's operating system user name from the operating system and check if it matches the requested database user name. This is only available for local connections. See Section 21.9 for details.

ldap

Authenticate using an LDAP server. See Section 21.10 for details.

radius

Authenticate using a RADIUS server. See Section 21.11 for details.

cert

Authenticate using SSL client certificates. See Section 21.12 for details.

pam

Authenticate using the Pluggable Authentication Modules (PAM) service provided by the operating system. See Section 21.13 for details.

bsd

Authenticate using the BSD Authentication service provided by the operating system. See Section 21.14 for details.

auth-options

After the auth-method field, there can be field(s) of the form name=value that specify options for the authentication method. Details about which options are available for which authentication methods appear below.

In addition to the method-specific options listed below, there is a method-independent authentication option clientcert, which can be specified in any hostssl record. This option can be set to verify-ca or verify-full. Both options require the client to present a valid (trusted) SSL certificate, while verify-full additionally enforces that the cn (Common Name) in the certificate matches the username or an applicable mapping. This behavior is

similar to the cert authentication method (see Section 21.12) but enables pairing the verification of client certificates with any authentication method that supports hostssl entries.

On any record using client certificate authentication (i.e. one using the cert authentication method or one using the clientcert option), you can specify which part of the client certificate credentials to match using the clientname option. This option can have one of two values. If you specify clientname=CN, which is the default, the username is matched against the certificate's Common Name (CN). If instead you specify clientname=DN the username is matched against the entire Distinguished Name (DN) of the certificate. This option is probably best used in conjunction with a username map. The comparison is done with the DN in RFC 2253 format. To see the DN of a client certificate in this format, do

openssl x509 -in myclient.crt -noout -subject -nameopt RFC2253 | sed "s/^subject=//"

Care needs to be taken when using this option, especially when using regular expression matching against the DN.

include

This line will be replaced by the contents of the given file.

include_if_exists

This line will be replaced by the content of the given file if the file exists. Otherwise, a message is logged to indicate that the file has been skipped.

include_dir

This line will be replaced by the contents of all the files found in the directory, if they don't start with a . and end with .conf, processed in file name order (according to C locale rules, i.e., numbers before letters, and uppercase letters before lowercase ones).

Files included by @ constructs are read as lists of names, which can be separated by either whitespace or commas. Comments are introduced by #, just as in pg_hba.conf, and nested @ constructs are allowed. Unless the file name following @ is an absolute path, it is taken to be relative to the directory containing the referencing file.

Since the pg_hba.conf records are examined sequentially for each connection attempt, the order of the records is significant. Typically, earlier records will have tight connection match parameters and weaker authentication methods, while later records will have looser match parameters and stronger authentication methods. For example, one might wish to use trust authentication for local TCP/IP connections but require a password for remote TCP/IP connections. In this case a record specifying trust authentication for connections from 127.0.0.1

would appear before a record specifying password authentication for a wider range of allowed client IP addresses.

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

Example pg_hba.conf Entries

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

# The same using local loopback TCP/IP connections.
#
# TYPE  DATABASE        USER            ADDRESS                 METHOD
host    all             all             127.0.0.1/32            trust

# The same as the previous line, but using a separate netmask column
#
# TYPE  DATABASE        USER            IP-ADDRESS      IP-MASK
METHOD
host    all             all             127.0.0.1
255.255.255.255     trust

# The same over IPv6.
#
# TYPE  DATABASE        USER            ADDRESS                 METHOD
host    all             all             ::1/128                 trust

# The same using a host name (would typically cover both IPv4 and
IPv6).
#
# TYPE  DATABASE        USER            ADDRESS                 METHOD
host    all             all             localhost               trust

# The same using a regular expression for DATABASE, that allows
connection
# to the database db1, db2 and any databases with a name beginning
with "db"
# and finishing with a number using two to four digits (like "db1234"
or
# "db12").
#
# TYPE  DATABASE                    USER            ADDRESS
METHOD
```

```
local   db1,"/^db\d{2,4}$",db2     all             localhost
trust

# Allow any user from any host with IP address 192.168.93.x to connect
# to database "postgres" as the same user name that ident reports for
# the connection (typically the operating system user name).
#
# TYPE  DATABASE        USER            ADDRESS                 METHOD
host    postgres        all             192.168.93.0/24         ident

# Allow any user from host 192.168.12.10 to connect to database
# "postgres" if the user's password is correctly supplied.
#
# TYPE  DATABASE        USER            ADDRESS                 METHOD
host    postgres        all             192.168.12.10/32        scram-
sha-256

# Allow any user from hosts in the example.com domain to connect to
# any database if the user's password is correctly supplied.
#
# Require SCRAM authentication for most users, but make an exception
# for user 'mike', who uses an older client that doesn't support SCRAM
# authentication.
#
# TYPE  DATABASE        USER            ADDRESS                 METHOD
host    all             mike            .example.com            md5
host    all             all             .example.com            scram-
sha-256

# In the absence of preceding "host" lines, these three lines will
# reject all connections from 192.168.54.1 (since that entry will be
# matched first), but allow GSSAPI-encrypted connections from anywhere
else
# on the Internet.  The zero mask causes no bits of the host IP
address to
# be considered, so it matches any host.  Unencrypted GSSAPI
connections
# (which "fall through" to the third line since "hostgssenc" only
matches
# encrypted GSSAPI connections) are allowed, but only from
192.168.12.10.
#
# TYPE  DATABASE        USER            ADDRESS                 METHOD
host    all             all             192.168.54.1/32         reject
hostgssenc all          all             0.0.0.0/0               gss
host    all             all             192.168.12.10/32        gss

# Allow users from 192.168.x.x hosts to connect to any database, if
# they pass the ident check.  If, for example, ident says the user is
# "bryanh" and he requests to connect as PostgreSQL user "guest1", the
# connection is allowed if there is an entry in pg_ident.conf for map
# "omicron" that says "bryanh" is allowed to connect as "guest1".
```

```
#
# TYPE  DATABASE        USER            ADDRESS                 METHOD
host    all             all             192.168.0.0/16          ident
map=omicron

# If these are the only four lines for local connections, they will
# allow local users to connect only to their own databases (databases
# with the same name as their database user name) except for users
whose
# name end with "helpdesk", administrators and members of role
"support",
# who can connect to all databases.  The file $PGDATA/admins contains
a
# list of names of administrators.  Passwords are required in all
cases.
#
# TYPE  DATABASE        USER            ADDRESS                 METHOD
local   sameuser        all                                     md5
local   all             /^.*helpdesk$                           md5
local   all             @admins                                 md5
local   all             +support                                md5

# The last two lines above can be combined into a single line:
local   all             @admins,+support                        md5

# The database column can also use lists and file names:
local   db1,db2,@demodbs  all                                   md5
```