

Installing Postgres 17 using source files

Dear readers,

In this document, I'll show you how to install PG 17 using source files.

Various methods to install PG

1. YUM: From Linux machine run yum install and required PG packages will be downloaded from the internet
2. RPM: Download required PG packages (.rpm) from internet to the Linux machine and then install those packages via rpm
3. Source files: Download required PG source files from internet to the Linux machine, compile them and then install.

Reasons to choose source files installation over other methods:

1. You want to use some customized PG options/settings e.g. Use different block size than default 8Kb or want bigger WAL files than default 16 MB etc.
2. You want to develop a custom PG release of yours.
3. You found an area to improve in PG and want to introduce that in PG
4. And many more

I have a CentOS 10 EC2 of t2.micro size (as it's available for free and suits to our purpose), so let's begin.

1. Installing the necessary packages:
 - a. gcc: To compile and execute PG core files
 - b. make: As PG is written in C language, it's core files will be compiled and executed using make utility. Makefile is a special file that has entries of the files to be compiled and then run (resulting into PG installation in our case).
 - c. zlib-devel: To support various PG compression algorithms
 - d. wget: To download PG files from the internet
 - e. libicu-devel: Development files for International Components for Unicode
 - f. bison: Needed for debugging PG parser grammar
 - g. flex: A tool for generating scanners (text pattern recognizers)
 - h. perl: Needed for OpenSSL configuration option
 - i. readline-devel: To provide history in psql (when we hit up arrow, we can see previous commands we executed)

```
[root@ip-172-31-31-68 ~]# cat /etc/redhat-release
CentOS Stream release 10 (Coughlan)
```

```
[root@ip-172-31-31-68 ~]# yum install -y gcc readline-devel make zlib-devel openssl-devel
wget libicu-devel bison flex perl
```

2. Creating postgres user and directories for PGDATA, PG binaries

```
[root@ip-172-31-31-68 ~]# useradd -d /home/postgres -U postgres
[root@ip-172-31-31-68 ~]# echo postgres | passwd "postgres" --stdin
BAD PASSWORD: The password contains the user name in some form
```

```
[root@ip-172-31-31-68 ~]# mkdir -p /pgdata/17/main
[root@ip-172-31-31-68 ~]# chown -R postgres:postgres /pgdata
[root@ip-172-31-31-68 ~]# chmod -R 755 /pgdata
[root@ip-172-31-31-68 ~]# mkdir -p /home/postgres/bin/pgsql/17
[root@ip-172-31-31-68 ~]# chmod -R 755 /home/postgres/bin/pgsql/17
[root@ip-172-31-31-68 ~]# chown postgres: /home/postgres/bin/pgsql/17
```

3. Downloading and unzipping PG binaries

```
[root@ip-172-31-31-68 ~]# cd /tmp
[root@ip-172-31-31-68 tmp]# wget https://ftp.postgresql.org/pub/source/v17.2/postgresql-17.2.tar.gz
[root@ip-172-31-31-68 tmp]# ls -lh postgresql-17*
[root@ip-172-31-31-68 tmp]# gunzip postgresql-17.2.tar.gz
[root@ip-172-31-31-68 tmp]# tar -xpf postgresql-17.2.tar
[root@ip-172-31-31-68 tmp]# cd postgresql-17.2
[root@ip-172-31-31-68 postgresql-17.2]# ls -lh
```

```
[root@ip-172-31-31-68 tmp]# ls -lh postgresql-17*
-rw-r--r--. 1 root root 27M Nov 18 20:44 postgresql-17.2.tar.gz
[root@ip-172-31-31-68 tmp]# gunzip postgresql-17.2.tar.gz
[root@ip-172-31-31-68 tmp]# tar -xpf postgresql-17.2.tar
[root@ip-172-31-31-68 tmp]# ls
null.sh                                systemd-private-0489027cba4a46178f1e091b385e3907-chronyd.service-Nsm7BF      terraform_567845506.sh
postgresql-17.2                       systemd-private-0489027cba4a46178f1e091b385e3907-dbus-broker.service-de4VTR
postgresql-17.2.tar                   systemd-private-0489027cba4a46178f1e091b385e3907-systemd-logind.service-UcH3Di
[root@ip-172-31-31-68 tmp]# cd postgresql-17.2
[root@ip-172-31-31-68 postgresql-17.2]# ls -lh
total 820K
-rw-rw-r--. 1 root root 365 Nov 18 20:32 aclocal.m4
drwxrwxr-x. 2 root root 4.0K Nov 18 20:32 config
-rwxrwxr-x. 1 root root 565K Nov 18 20:32 configure
-rw-rw-r--. 1 root root 87K Nov 18 20:32 configure.ac
drwxrwxr-x. 59 root root 4.0K Nov 18 20:32 contrib
-rw-rw-r--. 1 root root 1.2K Nov 18 20:32 COPYRIGHT
drwxrwxr-x. 3 root root 87 Nov 18 20:32 doc
-rw-rw-r--. 1 root root 4.1K Nov 18 20:32 GNUmakefile.in
-rw-rw-r--. 1 root root 277 Nov 18 20:32 HISTORY
-rw-rw-r--. 1 root root 1.8K Nov 18 20:32 Makefile
-rw-rw-r--. 1 root root 113K Nov 18 20:32 meson.build
-rw-rw-r--. 1 root root 6.4K Nov 18 20:32 meson_options.txt
-rw-rw-r--. 1 root root 983 Nov 18 20:32 README.md
drwxrwxr-x. 16 root root 4.0K Nov 18 20:32 src
[root@ip-172-31-31-68 postgresql-17.2]#
```

4. We can see configure utility that'll help us to install PG using various options. We can see various options this utility provides using below command

```
[[root@ip-172-31-31-68 postgresql-17.2]# ./configure --help
```

```
[root@ip-172-31-31-68 postgresql-17.2]# ./configure --help
`configure' configures PostgreSQL 17.2 to adapt to many kinds of systems.

Usage: ./configure [OPTION]... [VAR=VALUE]...

To assign environment variables (e.g., CC, CFLAGS...), specify them as
VAR=VALUE.  See below for descriptions of some of the useful variables.

Defaults for the options are specified in brackets.

Configuration:
  -h, --help                display this help and exit
  --help=short              display options specific to this package
  --help=recursive          display the short help of all the included packages
  -V, --version              display version information and exit
  -q, --quiet, --silent     do not print `checking ...' messages
  --cache-file=FILE         cache test results in FILE [disabled]
  -C, --config-cache         alias for `--cache-file=config.cache'
  -n, --no-create            do not create output files
  --srcdir=DIR               find the sources in DIR [configure dir or `..']

Installation directories:
  --prefix=PREFIX            install architecture-independent files in PREFIX
                             [/usr/local/pgsql]
  --exec-prefix=EPREFIX      install architecture-dependent files in EPREFIX
                             [PREFIX]
```

5. Generating PG binaries (kind of a PG home where various utilities will reside e.g. psql, pg_dump)

```
[root@ip-172-31-31-68 postgresql-17.2]# ./configure --prefix=/home/postgres/bin/pgsql/17 --
with-openssl
[root@ip-172-31-31-68 postgresql-17.2]# make
[root@ip-172-31-31-68 postgresql-17.2]# make install
```

6. Generating an environment file/profile for postgres user

```
[root@ip-172-31-31-68 postgresql-17.2]# echo '
alias lt="ls -ltrh"
alias lta="ls -ltrha"
alias cl="clear"
alias lk="ps -ef | grep "
alias lki="ps -ef | grep "
export PGDATA=/pgdata/17/main
export PATH=/home/postgres/bin/pgsql/17/bin:$PATH
export LD_LIBRARY_PATH=/home/postgres/bin/pgsql/17/lib:$LD_LIBRARY_PATH
' >> /home/postgres/.bash_profile

[root@ip-172-31-31-68 postgresql-17.2]# chmod 755 /home/postgres/.bash_profile

[root@ip-172-31-31-68 postgresql-17.2]# chown postgres: /home/postgres/.bash_profile
```

7. Initializing and starting the PG cluster via postgres user

```
[root@ip-172-31-31-68 postgresql-17.2]# sudo su - postgres
[postgres@ip-172-31-31-68 ~]$ which initdb
~/bin/pgsql/17/bin/initdb
[postgres@ip-172-31-31-68 ~]$ echo $PGDATA/
/pgdata/17/main/
[postgres@ip-172-31-31-68 ~]$ initdb -D $PGDATA -k
[postgres@ip-172-31-31-68 ~]$ pg_ctl -D $PGDATA/ start
```

```
[postgres@ip-172-31-31-68 ~]$ initdb -D $PGDATA -k
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_US.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 enabled.

fixing permissions on existing directory /pgdata/17/main ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default "max_connections" ... 100
selecting default "shared_buffers" ... 128MB
selecting default time zone ... UTC
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

initdb: warning: enabling "trust" authentication for local connections
initdb: hint: You can change this by editing pg_hba.conf or using the option -A, or --auth-local and --auth-host, the next time you run initdb.

Success. You can now start the database server using:

    pg_ctl -D /pgdata/17/main -l logfile start

[postgres@ip-172-31-31-68 ~]$ pg_ctl -D $PGDATA/ start
waiting for server to start....2025-01-01 19:02:28.961 UTC [47273] LOG:  starting PostgreSQL 17.2 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 14.2.1 202
41104 (Red Hat 14.2.1-6), 64-bit
2025-01-01 19:02:28.963 UTC [47273] LOG:  listening on IPv6 address ":::1", port 5432
2025-01-01 19:02:28.963 UTC [47273] LOG:  listening on IPv4 address "127.0.0.1", port 5432
2025-01-01 19:02:28.968 UTC [47273] LOG:  listening on Unix socket "/tmp/.s.PGSQL.5432"
2025-01-01 19:02:28.974 UTC [47276] LOG:  database system was shut down at 2025-01-01 19:01:33 UTC
2025-01-01 19:02:28.977 UTC [47273] LOG:  database system is ready to accept connections
done
server started
[postgres@ip-172-31-31-68 ~]$ █
```

8. Verification

```
[postgres@ip-172-31-31-68 ~]$ pg_ctl -D $PGDATA/ status

# from psql prompt
postgres=# select version();
postgres=# show data_checksums;
```

```
[postgres@ip-172-31-31-68 ~]$ pg_ctl -D $PGDATA/ status
pg_ctl: server is running (PID: 47273)
/home/postgres/bin/pgsql/17/bin/postgres "-D" "/pgdata/17/main"
[postgres@ip-172-31-31-68 ~]$ psql
psql (17.2)
Type "help" for help.

postgres=# select version();
               version
-----
PostgreSQL 17.2 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 14.2.1 20241104 (Red Hat 14.2.1-6), 64-bit
(1 row)

postgres=# show data_checksums;
 data_checksums
-----
 on
(1 row)
```