Install

----------postgresql installation----------

First we have to create directory.
mkdir postgresql

1. Download file:-

wget https://ftp.postgresql.org/pub/source/v14.16/postgresql-14.16.tar.gz

Explanation: This command downloads the specified version of PostgreSQL in a compressed tarball format.

2. Untar file:-

tar -xzvf postgresql-14.16.tar.gz
cd postgresql-14.16

Explanation: The tar command extracts the contents of the tarball, and cd changes the directory to the extracted folder.

3. User creation:-

adduser postgres
password:-Postgres@123

useradd -m -s /bin/bash -g postgres postgres

Feature	useradd	adduser
Command Type	Low-level binary	High-level Perl script
Default Home Directory	Not created (unless -m is used)	Created automatically
Interactive Mode	No	Yes
Available In	Most Linux distros	Primarily Debian-based distros
Ease of Use	Requires options for full setup	User-friendly

Explanation: Create postgres user

4. Dependces install:-

Ubantu:-

apt-get install -y libreadline-dev build-essential zlib1g-dev libxml2 libxml2-dev libxslt-dev libssl-dev libldap2-dev uuid-dev libossp-uuid-dev

Redhat:-

sudo yum install -y epel-release gcc readline-devel zlib-devel zlib libxml2 libxml2-devel libxslt libxslt-devel openldap-devel libuuid-devel ossp-uuid-devel libicu-devel && sudo yum groupinstall -y "Development Tools"

sudo dnf install -y gcc epel-release gcc readline-devel zlib-devel zlib libxml2 libxml2-devel libxslt libxslt-devel openldap-devel libuuid-devel libicu-devel && sudo dnf groupinstall -y "Development Tools"
dnf install -y bison
sudo dnf install -y flex
dnf install ossp-uuid-devel

Explanation: These commands install the essential development tools and libraries required to compile PostgreSQL from source.

5. Configure and Compile PostgreSQL
./configure --prefix=/usr/lib/pgsql-16.7 --with-ldap --with-libxml --with-libxslt --with-openssl --with-uuid=e2fs

make

make install

Error:-

make -C ./src/backend generated-headers make[1]: Entering directory '/postgresql/postgresql-17.5/src/backend' make -C ../include/catalog generated-headers make[2]: Entering directory '/postgresql/postgresql-17.5/src/include/catalog' '/usr/bin/perl' ../../../src/backend/catalog/genbki.pl --include-path=../../../src/include/ \ --set-version=17 ../../../src/include/catalog/pg_proc.h ../../../src/include/catalog/pg_type.h ../../../src/include/catalog/pg_attribute.h ../../../src/include/catalog/pg_class.h ../../../src/include/catalog/pg_attrdef.h ../../../src/include/catalog/pg_constraint.h ../../../src/include/catalog/pg_inherits.h
make[2]: *** [Makefile:141: bki-stamp] Error 2 make[2]: Leaving directory '/postgresql/postgresql-17.5/src/include/catalog' make[1]: *** [Makefile:121: submake-catalog-headers] Error 2 make[1]: Leaving directory '/postgresql/postgresql-17.5/src/backend' make: *** [src/Makefile.global:384: submake-generated-headers] Error 2

Sol:-

For RHEL / CentOS / Rocky / AlmaLinux
sudo dnf install perl-core perl-Module-CoreList -y
sudo dnf install perl -y

For Ubuntu / Debian
sudo apt-get update
sudo apt-get install perl-base perl-modules -y

Contrib:-

cd postgresql-14.16/contrib

make && make install

Explanation:

- **The ./configure command prepares the build environment, and**
- **make compiles the source code into executable binaries.**

Explanation:

- **make install command installs the compiled binaries and libraries to the specified prefix directory.**

1. make world:- everything including contrib, documentation and man pages

2. make world-bin:- everything, except documentation

If we have to remove everything:-

1.make clean

2 ./configure --prefix=/usr/lib/pgsql-15.6.0 --with-openssl --with-ldap --with-libxml --with-libxslt --with-openssl --with-uuid=e2fs

3. sudo apt-get install -y libssl-dev
4. make && make install

6. Setting Up and Configuring the PostgreSQL Data Directory

mkdir -p /data/pgdata_16----------(data directory)

chown -R postgres:postgres /data

chmod -R 700 /data/pgdata_14

chown -R postgres:postgres data/: Ensures that the postgres user and group own the directory and its contents, allowing PostgreSQL to function correctly.

chmod -R O700 data/: Grants full access to postgres user for the data directory (not recommended for production; consider using chmod 0700 for better security in production).

Explanation: mkdir data: Creates a new directory where PostgreSQL will store its database files.

7. Initialize and Start PostgreSQL Server:-

sudo su - postgres

/usr/lib/pgsql-14.16/bin/initdb -D /data/pgdata_14

/usr/lib/pgsql-14.16/bin/pg_ctl -D /data/pgdata_14 start -l log_file

Explanation: The pg_ctl command is used to start, stop, or restart the PostgreSQL server. In this case, we use it to start the server.

- **-D /pgdata/data specifies the location of the database data directory, which we set earlier with initdb.**
- **-l logfile specifies the log file where PostgreSQL will write its output and any errors.**
- **start tells pg_ctl to start the PostgreSQL server using the specified data directory and log file.**

8. set bash profile:-

vi .bash_profile ----------(postgres,root user)

```
export PATH=/usr/lib/pgsql-14.16/bin:$PATH
export PGDATA=/data/pgdata_14
export PGPORT=5432
export PGDATABASE=postgres
export PGHOST=/tmp
```

source .bash_profile ----------(postgres,root user)

9. create service file:-

vi /etc/systemd/system/postgresql-14.service

```
[Unit]
Description=PostgreSQL Database Server
After=network.target

[Service]
Type=forking
User=postgres
ExecStart=/usr/lib/pgsql-14.16/bin/pg_ctl -D /data/pgdata_14 start
ExecStop=/usr/lib/pgsql-14.16/bin/pg_ctl -D /data/pgdata_14 stop
ExecReload=/usr/lib/pgsql-14.16/bin/pg_ctl -D /data/pgdata_14 reload
TimeoutSec=-10

[Install]
WantedBy=multi-user.target
```

(OR)

vi /etc/systemd/system/postgresql-14.service

```
[Unit]
Description=PostgreSQL Database Server
After=network.target

[Service]
Type=forking
```

User=postgres
Group=postgres
ExecStart=/usr/lib/postgresql/14/bin/pg_ctl start -D /pg_data/db/postgresql/14/main -o "-c config_file=/etc/postgresql/14/main/postgresql.conf"
ExecStop=/usr/lib/postgresql/14/bin/pg_ctl stop -D /pg_data/db/postgresql/14/main
ExecReload=/usr/lib/postgresql/14/bin/pg_ctl reload -D /pg_data/db/postgresql/14/main
TimeoutSec=0
PIDFile=/pg_data/db/postgresql/14/main/postmaster.pid
Restart=always
RestartSec=5

[Install]
WantedBy=multi-user.target

systemctl daemon-reload
systemctl enable postgresql.service
systemctl start postgresql.service

---

10. Achival parameter:-

Archive mode:-

archive_mode=on
archive_command = 'cp %p /application/pg_archive/%f'

Using pg_switch_wal()

SELECT pg_switch_wal();

---

11. Extension Installation:-

Pgcron & pgpartman(extension installation) :-

git clone https://github.com/pgpartman/pg_partman.git

git clone https://github.com/citusdata/pg_cron.git

cd postgresql-14.16/contrib

Make && make install

Pg_cron:-

#--
#PG_CRON
#--
cron.timezone='Asia/Kolkata'
#cron.log_statement = 'all'
#cron.database_name = 'postgres'

create extension pg_cron;

create extension pg_partman;

----------uuid
sudo apt-get update
sudo apt-get install uuid-dev libossp-uuid-dev build-essential

cd /path/to/postgresql-15.6
./configure --prefix=/usr/lib/pgsql-15.6 --with-uuid=e2fs # Or --with-uuid=ossp if using OSSP UUID

cd /data/postgresql-15/postgresql-15.7/contrib/uuid-ossp

make && make install

create extension "uuid-ossp";

---

12. Repliaction set:-

update postgresql.conf file:-

1.archive_mode=on
2.archive_command = 'cp %p /application/pg_archive/%f'
3.max_wal_sender = 10
4.wal_level = repllica
5.wal_keep_size = 50
6.wal_log_hint = on
8.listen_address = *

::::: Explanation of the options :::::

- **listen_addresses = '*'**

- Tells PostgreSQL to listen on all network interfaces.
- Required so the standby server can connect to the primary server for replication.

- **wal_level = replica**

- Sets the level of information written to the WAL (Write-Ahead Log).
- The replica level is the minimum required for physical replication.
- Enables WAL archiving and streaming replication.

- **max_wal_senders = 10**

- Defines how many WAL sender processes the primary server can run simultaneously.
- Each standby server that connects consumes one WAL sender.
- Set this according to the number of standbys you plan to support.

- **wal_keep_size = 256**

- Specifies the minimum size (in MB) of WAL files to keep.
- Helps prevent replication failure due to missing WAL files if the standby is delayed.

- **hot_standby = on**

- Allows the standby server to accept read-only queries while in recovery mode.
- This setting must be enabled on the standby node.

/usr/lib/pgsql-14.16/bin/pg_basebackup --checkpoint=fast -D /data/pgdata_14 -h 10.20.31.11 -p 5432 -Xs -R -P > /data/backup.log 2>&1 &

==========

13. Parameter set according to cpu & RAM:-

CPU CORE:- 16 core
RAM :- 32 GB

==========
max_connections = 996
shared_buffers = 8GB
effective_cache_size = 24GB
maintenance_work_mem = 2GB
checkpoint_completion_target = 0.9
wal_buffers = 16MB
default_statistics_target = 100
random_page_cost = 1.1
effective_io_concurrency = 200
work_mem = 2105kB
huge_pages = try
min_wal_size = 2GB
max_wal_size = 8GB
max_worker_processes = 16
max_parallel_workers_per_gather = 4
max_parallel_workers = 16
max_parallel_maintenance_workers = 4

==========

14. Log_parameter set:-

Log parameter:-

log_destination = 'stderr'
logging_collector = on
log_directory = '/var/log/edb/as<epas_version>'
log_filename = postgresql-%a.log'
log_truncate_on_rotation = on
log_line_prefix = %t [%p-%l]: user=%u,db=%d,app=%a,client=%h
log_rotation = on

15. Auto vacuum parameter set:-

Autovacuum paratmeter=

autovacuum	on
autovacuum_analyze_scale_factor	0.1
autovacuum_analyze_threshold	50
autovacuum_freeze_max_age	2000000000
autovacuum_max_workers	3
autovacuum_multixact_freeze_max_age	4000000000
autovacuum_naptime	900
autovacuum_vacuum_cost_delay	2
autovacuum_vacuum_cost_limit	200
autovacuum_vacuum_insert_scale_factor	0.2
autovacuum_vacuum_insert_threshold	20000
autovacuum_vacuum_scale_factor	0.2
autovacuum_vacuum_threshold	50
autovacuum_work_mem	-1

---

16. Move Wal_files from $PGDATA to New Location

1) Create Postgresql Directory
mkdir -p /u03/app/16.2/wal_files

2) Change ownership to postgres
chown -h postgres:postgres /u03/app/16.2/wal_files

3) Stop postgresql
./pg_ctl stop

4)rsync all files from $PGDATA/pg_wal to new location
rsync -av /u02/app/16.2/data/pg_wal/* /u03/app/16.2/wal_files

5) Check all files are synced
ls -la /u03/app/16.2/wal_files

6) Take a backup of pg_wal folder
mv /u02/app/16.2/data/pg_wal /u02/app/16.2/data/pg_wal-backup

7) Create a Symbolic link
sudo ln -s /u03/app/16.2/wal_files/ /u02/app/16.2/data/pg_wal

8) Start Postgresql
./pg_ctl start

9) Verify DB connection using your db credentials/information
psql -h localhost -U postgres -p 5432
select pg_switch_Wal() (check wal files in new location)

10) Remove the old folder
rm -rf /u02/app/16.2/data/pg_wal-backup

---

17. Move TEMP Files/Tables From Default Location to New Location:

1) create temporary table test1 (empno int);
2) select pg_relation_filepath('test1');
3) mkdir -p /u05/app/16.2/ Temp_files
4) create tablespace temp1 location '/u05/app/16.2/temp_files';
5) alter system set temp_tablespaces = 'temp1';
6) show temp_tablespaces;
7) select pg_reload_conf();
8) create temporary table test2 (empno int);
9) select pg_relation_filepath('test2');