PostgreSQL Installation Methods

➢ Package Manager (Yum) Installation:

It is the simplest method of installation using the system's package manager. By default is installs the latest version or installs the any version that is by default available on your OS.

Steps:-

1. **Update the system (command: sudo yum update or sudo dnf update)**

```
[root@localhost ~]# sudo yum update -y
Last metadata expiration check: 2:07:32 ago on Friday 31 January 2025 11:29:53 AM.
Dependencies resolved.
================================================================================
 Package                          Architecture          Version                  Repository
================================================================================
Installing:
 kernel                           x86_64                5.14.0-554.el9           baseos
Upgrading:
 NetworkManager                   x86_64                1:1.51.6-1.el9           baseos
 NetworkManager-adsl              x86_64                1:1.51.6-1.el9           baseos
 NetworkManager-bluetooth         x86_64                1:1.51.6-1.el9           baseos
 NetworkManager-config-server     noarch                1:1.51.6-1.el9           baseos
 NetworkManager-libnm             x86_64                1:1.51.6-1.el9           baseos
 NetworkManager-team              x86_64                1:1.51.6-1.el9           baseos
 NetworkManager-tui               x86_64                1:1.51.6-1.el9           baseos
 NetworkManager-wifi              x86_64                1:1.51.6-1.el9           baseos
 NetworkManager-wwan              x86_64                1:1.51.6-1.el9           baseos
```

2. **Install PostgreSQL**
   (command:sudo yum install postgresql-server postgresql-contrib -y)

```
[root@localhost ~]# yum install postgresql-server
Last metadata expiration check: 0:01:52 ago on Thursday 30 January 2025 12:51:33 PM.
Dependencies resolved.
================================================================================
 Package                      Architecture          Version                Repository
================================================================================
Installing:
 postgresql-server            x86_64                13.18-1.el9            appstream
Installing dependencies:
 postgresql                   x86_64                13.18-1.el9            appstream
 postgresql-private-libs      x86_64                13.18-1.el9            appstream

Transaction Summary
================================================================================
Install  3 Packages

Total download size: 7.5 M
Installed size: 29 M
Is this ok [y/N]: y
Downloading Packages:
(1/3): postgresql-private-libs-13.18-1.el9.x86_64.rpm                  72 kB/s | 137 kB
```

3. **Initialize PostgreSQL Database**
   (command : sudo postgresql-setup --initdb)

```
[root@localhost ~]# postgresql-setup initdb
WARNING: using obsoleted argument syntax, try --help
WARNING: arguments transformed to: postgresql-setup --initdb --unit postgresql
 * Initializing database in '/var/lib/pgsql/data'
 * Initialized, logs are in /var/lib/pgsql/initdb_postgresql.log
```

4. **Start PostgreSQL Service**
   (command: systemctl start postgresql)

5. **Enable PostgreSQL to Start on Boot**
   (command: systemctl enable postgresql)

```
[root@localhost ~]# systemctl start postgresql.service
[root@localhost ~]# systemctl enable postgresql.service
Created symlink /etc/systemd/system/multi-user.target.wants/postgresql.service → /usr/lib/systemd/system/postgresql.service.
[root@localhost ~]# systemctl status postgresql.service
● postgresql.service - PostgreSQL database server
     Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; preset: disabled)
     Active: active (running) since Thu 2025-01-30 16:48:09 IST; 16s ago
   Main PID: 47025 (postmaster)
      Tasks: 8 (limit: 4296)
     Memory: 31.1M
        CPU: 113ms
     CGroup: /system.slice/postgresql.service
             ├─47025 /usr/bin/postmaster -D /var/lib/pgsql/data
             ├─47026 "postgres: logger "
             ├─47028 "postgres: checkpointer "
             ├─47029 "postgres: background writer "
             ├─47030 "postgres: walwriter "
             ├─47031 "postgres: autovacuum launcher "
             ├─47032 "postgres: stats collector "
             └─47033 "postgres: logical replication launcher "
```

6. **Switch to postgres user**
   (if user is not exist create a user as postgres)

7. **Take Access to PostgreSQL using psql**

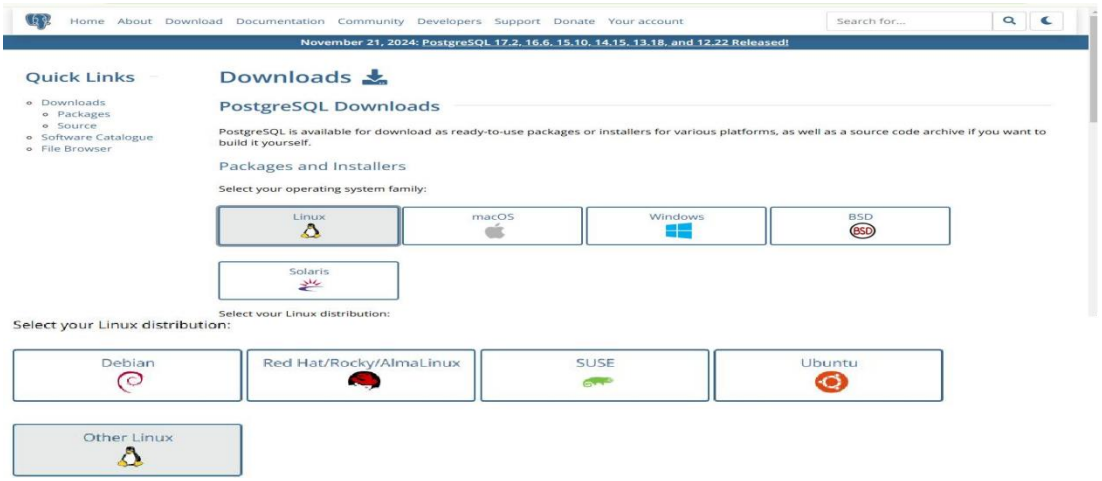
```
[root@localhost ~]# su - postgres
[postgres@localhost ~]$ psql
psql (13.18)
Type "help" for help.

postgres=#
```

➢ Repository (rpm) installation :

This method involves installing PostgreSQL from the PostgreSQL Global Development Group (PGDG) repository.

Steps:-

1. **Go to the official site of the postgresql.org .**
2. **Select your OS (operating system) & linux distribution.**



3. **select version**

4. **Select the repo package link and Download**

```
[root@localhost ~]# wget https://download.postgresql.org/pub/repos/yum/reporpms/EL-9-x86_64/pgdg-redhat-repo-latest.noarch.rpm
--2025-02-04 08:45:26--  https://download.postgresql.org/pub/repos/yum/reporpms/EL-9-x86_64/pgdg-redhat-repo-latest.noarch.rpm
Resolving download.postgresql.org (download.postgresql.org)... 87.238.57.227, 147.75.85.69, 217.196.149.55, ...
Connecting to download.postgresql.org (download.postgresql.org)|87.238.57.227|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12172 (12K) [application/x-redhat-package-manager]
Saving to: 'pgdg-redhat-repo-latest.noarch.rpm'

pgdg-redhat-repo-latest.noarch.rpm        100%[=====================================================================================>]

2025-02-04 08:45:28 (274 MB/s) - 'pgdg-redhat-repo-latest.noarch.rpm' saved [12172/12172]
```

5. **To download the necessary packages Go to Direct Rpm download click on direct download link**

6. **Download the all the necessary packages listed below.**

i. Postgresql server package
ii. Postgresql libraries package
iii. Postgresql contrib packages
iv. Postgresql global development packages
v. Postgresql development(devel) packages

i. PostgreSQL server package

- Purpose: This is the core package that provides the PostgreSQL database server itself.
- Includes:

PostgreSQL server binaries (postgres, initdb, pg_ctl, etc.)

Data directory initialization tools

Systemd service files (for starting/stopping PostgreSQL)

- Installed when: You want to actually run a PostgreSQL instance on the machine.
- Examples:

RHEL/CentOS: postgresql15-server

Debian/Ubuntu: postgresql-15

ii. PostgreSQL libraries package

- Purpose: Provides the client libraries needed for applications to connect to PostgreSQL.
- Includes:

libpq (PostgreSQL C client library)

Shared libraries required by other client applications or drivers (like ODBC, JDBC, psycopg2 in Python).

- Installed when: You need client connectivity or when developing software that links to PostgreSQL.
- Examples:

RHEL/CentOS: postgresql15-libs

Debian/Ubuntu: libpq15

iii. PostgreSQL contrib packages

- Purpose: Contains a set of additional extensions and tools contributed by the PostgreSQL community.
- Includes:

Useful extensions like pg_stat_statements, tablefunc, dblink, hstore, uuid-ossp, etc.

Extra utilities that are not part of the core server.

- Installed when: You want extra functionality beyond the core PostgreSQL.
- Examples:

RHEL/CentOS: postgresql15-contrib

Debian/Ubuntu: postgresql-contrib

iv. PostgreSQL Global Development Group (PGDG) packages

- Purpose: These are the official packages provided by the PostgreSQL Global Development Group (PGDG) repository.

- Difference:

OS vendors (RHEL, Debian, Ubuntu) provide their own PostgreSQL packages, but these are often older versions.

PGDG packages are maintained directly by the PostgreSQL community and provide newer versions.

- Installed when: You want the latest stable release or versions not available in default OS repos.

- Examples:

PGDG repo RPM/DEB packages (e.g., pgdg-redhat-repo, pgdg-common)

v. PostgreSQL development (devel) packages

- Purpose: Provides header files and static libraries required for compiling and developing applications or extensions against PostgreSQL.

- Includes:

C header files (libpq-fe.h, etc.)

Development libraries for building client applications

- Installed when:

You want to compile PostgreSQL extensions (like postgis, timescaledb, etc.)

You're a developer writing C programs using PostgreSQL's client library.

- Examples:

RHEL/CentOS: postgresql15-devel

Debian/Ubuntu: libpq-dev

✅In summary:

Server → Runs PostgreSQL database

Libraries → Needed for clients/apps to connect

Contrib → Extra extensions and tools

PGDG → Community-provided packages with latest versions

Devel → Needed to compile extensions or apps against PostgreSQL

```
[root@localhost ~]# wget https://download.postgresql.org/pub/repos/yum/13/redhat/rhel-9-x86_64/postgresql13-server-13.18-1PGDG.rhel9.x86_64.rpm
--2025-02-04 08:55:05--  https://download.postgresql.org/pub/repos/yum/13/redhat/rhel-9-x86_64/postgresql13-server-13.18-1PGDG.rhel9.x86_64.rpm
Resolving download.postgresql.org (download.postgresql.org)... 217.196.149.55, 147.75.85.69, 87.238.57.227, ...
Connecting to download.postgresql.org (download.postgresql.org)|217.196.149.55|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5912600 (5.6M) [application/x-redhat-package-manager]
Saving to: 'postgresql13-server-13.18-1PGDG.rhel9.x86_64.rpm'

postgresql13-server-13.18-1PGDG.rhel9.x86 100%[=============================================================================================>]   5.64M  2.3
```

```
[root@localhost ~]# wget https://download.postgresql.org/pub/repos/yum/13/redhat/rhel-9-x86_64/postgresql13-libs-13.18-1PGDG.rhel9.x86_64.rpm
--2025-02-04 08:54:40--  https://download.postgresql.org/pub/repos/yum/13/redhat/rhel-9-x86_64/postgresql13-libs-13.18-1PGDG.rhel9.x86_64.rpm
Resolving download.postgresql.org (download.postgresql.org)... 147.75.85.69, 87.238.57.227, 72.32.157.246, ...
Connecting to download.postgresql.org (download.postgresql.org)|147.75.85.69|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 787725 (769K) [application/x-redhat-package-manager]
Saving to: 'postgresql13-libs-13.18-1PGDG.rhel9.x86_64.rpm'

postgresql13-libs-13.18-1PGDG.rhel9.x86_6 100%[=============================================================================================>] 769.26K  1.02MB/s
```

```
[root@localhost ~]# wget https://download.postgresql.org/pub/repos/yum/13/redhat/rhel-9-x86_64/postgresql13-13.18-1PGDG.rhel9.x86_64.rpm
--2025-02-04 09:09:38--  https://download.postgresql.org/pub/repos/yum/13/redhat/rhel-9-x86_64/postgresql13-13.18-1PGDG.rhel9.x86_64.rpm
Resolving download.postgresql.org (download.postgresql.org)... 72.32.157.246, 147.75.85.69, 87.238.57.227, ...
Connecting to download.postgresql.org (download.postgresql.org)|72.32.157.246|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1437036 (1.4M) [application/x-redhat-package-manager]
Saving to: 'postgresql13-13.18-1PGDG.rhel9.x86_64.rpm'

postgresql13-13.18-1PGDG.rhel9.x86_64.rpm 100%[=============================================================================================>]
```

```
[root@localhost ~]# wget https://download.postgresql.org/pub/repos/yum/13/redhat/rhel-9-x86_64/postgresql13-devel-13.18-1PGDG.rhel9.x86_64.rpm
--2025-02-04 08:54:20--  https://download.postgresql.org/pub/repos/yum/13/redhat/rhel-9-x86_64/postgresql13-devel-13.18-1PGDG.rhel9.x86_64.rpm
Resolving download.postgresql.org (download.postgresql.org)... 217.196.149.55, 72.32.157.246, 87.238.57.227, ...
Connecting to download.postgresql.org (download.postgresql.org)|217.196.149.55|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5101171 (4.9M) [application/x-redhat-package-manager]
Saving to: 'postgresql13-devel-13.18-1PGDG.rhel9.x86_64.rpm'

postgresql13-devel-13.18-1PGDG.rhel9.x86_ 100%[=============================================================================================>]   4.86M  3

2025-02-04 08:54:23 (3.82 MB/s) - 'postgresql13-devel-13.18-1PGDG.rhel9.x86_64.rpm' saved [5101171/5101171]
```

7. Install all the downloaded packages using rpm command (command: rpm -ivh packages) Install it sequence wise!

i. Repo package

```
[root@localhost ~]# rpm -ivh pgdg-redhat-repo-latest.noarch.rpm
warning: pgdg-redhat-repo-latest.noarch.rpm: Header V4 RSA/SHA256 Signature, key ID 08b40d20: NOKEY
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:pgdg-redhat-repo-42.0-46PGDG     ################################# [100%]
```

ii. Libs package

```
[root@localhost ~]# rpm -ivh postgresql13-libs-13.18-1PGDG.rhel9.x86_64.rpm
warning: postgresql13-libs-13.18-1PGDG.rhel9.x86_64.rpm: Header V4 RSA/SHA256 Signature, key ID 08b40d20: NOKEY
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:postgresql13-libs-13.18-1PGDG.rhe################################# [100%]
```

iii. PGDG package

```
[root@localhost ~]# rpm -ivh postgresql13-13.18-1PGDG.rhel9.x86_64.rpm
warning: postgresql13-13.18-1PGDG.rhel9.x86_64.rpm: Header V4 RSA/SHA256 Signature, key ID 08b40d20: NOKEY
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:postgresql13-13.18-1PGDG.rhel9   ################################# [100%]
```

iv. Server package

```
[root@localhost ~]# rpm -ivh postgresql13-server-13.18-1PGDG.rhel9.x86_64.rpm
warning: postgresql13-server-13.18-1PGDG.rhel9.x86_64.rpm: Header V4 RSA/SHA256 Signature, key ID 08b40d20: NOKEY
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:postgresql13-server-13.18-1PGDG.r################################# [100%]
```

v. Devel package

```
[root@localhost ~]# rpm -ivh postgresql13-devel-13.18-1PGDG.rhel9.x86_64.rpm
warning: postgresql13-devel-13.18-1PGDG.rhel9.x86_64.rpm: Header V4 RSA/SHA256 Signature, key ID 08b40d20: NOKEY
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:postgresql13-devel-13.18-1PGDG.rh################################# [100%]
```

8. **Initialize the Database Cluster:**
   (command: /usr/pgsql-13/bin/postgresql-<version>-setup initdb)

```
[root@localhost ~]# sudo /usr/pgsql-13/bin/postgresql-13-setup initdb
Initializing database ... OK

[root@localhost ~]#
```

9. **Enable PostgreSQL to Start on Boot**
10. **Start PostgreSQL Service.**

```
[root@localhost ~]# systemctl enable postgresql-13.service
Created symlink /etc/systemd/system/multi-user.target.wants/postgresql-13.service → /usr/lib/systemd/system/postgresql-13.service.
[root@localhost ~]# systemctl start postgresql-13.service
[root@localhost ~]# systemctl status postgresql-13.service
● postgresql-13.service - PostgreSQL 13 database server
     Loaded: loaded (/usr/lib/systemd/system/postgresql-13.service; enabled; preset: disabled)
     Active: active (running) since Tue 2025-02-04 10:11:28 IST; 7s ago
       Docs: https://www.postgresql.org/docs/13/static/
    Process: 50951 ExecStartPre=/usr/pgsql-13/bin/postgresql-13-check-db-dir ${PGDATA} (code=exited, status=0/SUCCESS)
   Main PID: 50956 (postmaster)
      Tasks: 8 (limit: 4296)
     Memory: 16.8M
        CPU: 60ms
     CGroup: /system.slice/postgresql-13.service
             ├─50956 /usr/pgsql-13/bin/postmaster -D /var/lib/pgsql/13/data/
             ├─50957 "postgres: logger "
             ├─50959 "postgres: checkpointer "
             ├─50960 "postgres: background writer "
             ├─50961 "postgres: walwriter "
             ├─50962 "postgres: autovacuum launcher "
             ├─50963 "postgres: stats collector "
             ├─50964 "postgres: logical replication launcher "
```

8. **Switch to postgres user**
   (if user is not exist create a user as postgres)

9. **Take Access to PostgreSQL using psql**

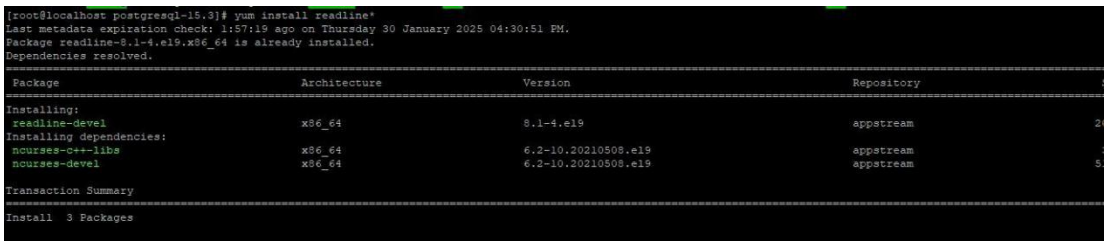
```
[root@localhost ~]# su - postgres
[postgres@localhost ~]$ psql
psql (13.18)
Type "help" for help.

postgres=#
```

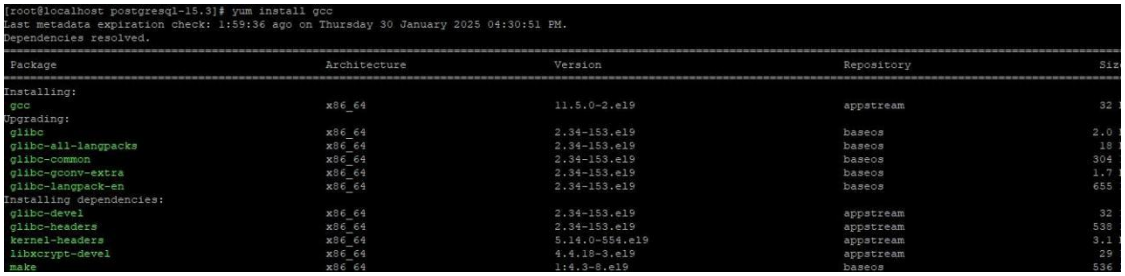
➢ Source code installation:

source code installation provides greater flexibility and control, especially if you need to enable specific configurations or customizations.

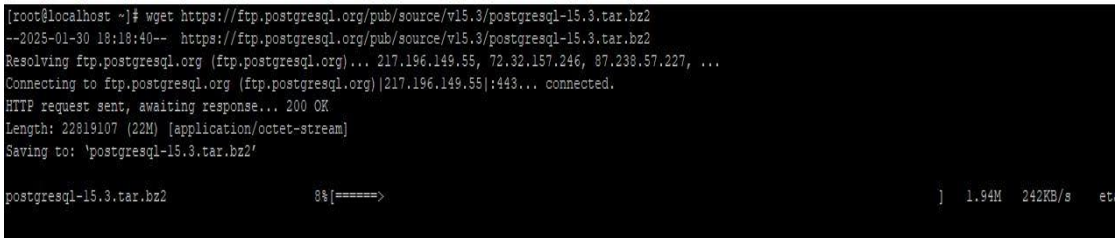
Steps:

1. **Install the required development tools and libraries:**
   Install gcc, readline, zlib.

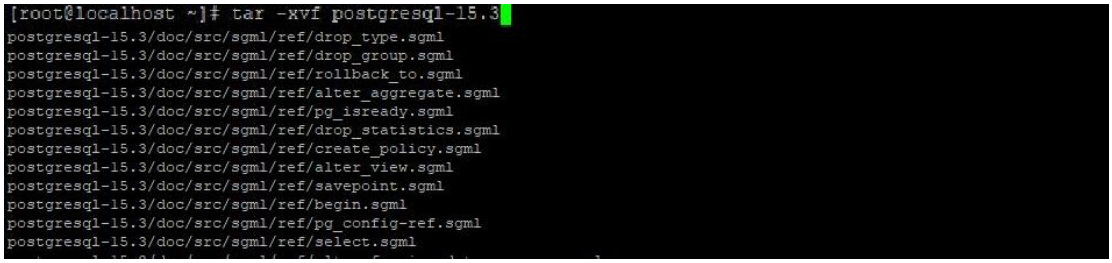




2. **Download Source Code**
   (command: wget <source code link>)



3. **Extract the Tar file**
   (command: tar -xvf postgresql-<version>.tar.gz)



4. **After extracting you will a new directory of your postgresql version is created**
   Go to that directory.

5. Execute the configure file . (command: ./configure)

If you didn't installed the required libraries before the postgresql installation , It will show you an error message for that library . install it and again execute the configure file.

If you don't need the libraries you can skip using a flag --without-name of that library.

```
[root@localhost postgresql-15.3]# ls
aclocal.m4  config  configure  configure.ac  contrib  COPYRIGHT  doc  GNUmakefile.in  HISTORY  INSTALL  Makefile  README  src
[root@localhost postgresql-15.3]# ./configure
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 gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
```

6. Compile PostgreSQL (command: make)

If you try to execute this step executing the configure file it will show an error.

```
[root@localhost postgresql-15.3]# make
make -C ./src/backend generated-headers
make[1]: Entering directory '/root/postgresql-15.3/src/backend'
make -C catalog distprep generated-header-symlinks
make[2]: Entering directory '/root/postgresql-15.3/src/backend/catalog'
make[2]: Nothing to be done for 'distprep'.
prereqdir=`cd './' >/dev/null && pwd` && \
cd '../../../src/include/catalog/' && for file in pg_proc_d.h pg_type_d.h pg_attribute_d.h pg_class_d.h pg_attrdef_d.h pg_constraint_d.h pg_inh
_operator_d.h pg_opfamily_d.h pg_opclass_d.h pg_am_d.h pg_amop_d.h pg_amproc_d.h pg_language_d.h pg_largeobject_metadata_d.h pg_largeobject_d.h
istic_d.h pg_statistic_ext_d.h pg_statistic_ext_data_d.h pg_rewrite_d.h pg_trigger_d.h pg_event_trigger_d.h pg_description_d.h pg_cast_d.h pg_e
pg_conversion_d.h pg_depend_d.h pg_database_d.h pg_db_role_setting_d.h pg_tablespace_d.h pg_authid_d.h pg_auth_members_d.h pg_shdepend_d.h pg_s
nfig_d.h pg_ts_config_map_d.h pg_ts_dict_d.h pg_ts_parser_d.h pg_ts_template_d.h pg_extension_d.h pg_foreign_data_wrapper_d.h pg_foreign_server
pg_foreign_table_d.h pg_policy_d.h pg_replication_origin_d.h pg_default_acl_d.h pg_init_privs_d.h pg_seclabel_d.h pg_shseclabel_d.h pg_collatio
h pg_partitioned_table_d.h pg_range_d.h pg_transform_d.h pg_sequence_d.h pg_publication_d.h pg_publication_namespace_d.h pg_publication_rel_d.h
```

7. Install PostgreSQL (command: make install)

```
[root@localhost postgresql-15.3]# make install
make -C ./src/backend generated-headers
make[1]: Entering directory '/root/postgresql-15.3/src/backend'
make -C catalog distprep generated-header-symlinks
make[2]: Entering directory '/root/postgresql-15.3/src/backend/catalog'
make[2]: Nothing to be done for 'distprep'.
make[2]: Nothing to be done for 'generated-header-symlinks'.
make[2]: Leaving directory '/root/postgresql-15.3/src/backend/catalog'
make -C utils distprep generated-header-symlinks
make[2]: Entering directory '/root/postgresql-15.3/src/backend/utils'
```

8. Initialize PostgreSQL Database

```
[postgres@localhost ~]$ /usr/local/pgsql/bin/initdb -D /usr/local/pgsql/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 /usr/local/pgsql/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

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

Success. You can now start the database server using:
```

9. Start the postgrsql service

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

10. Take Access to PostgreSQL using psql

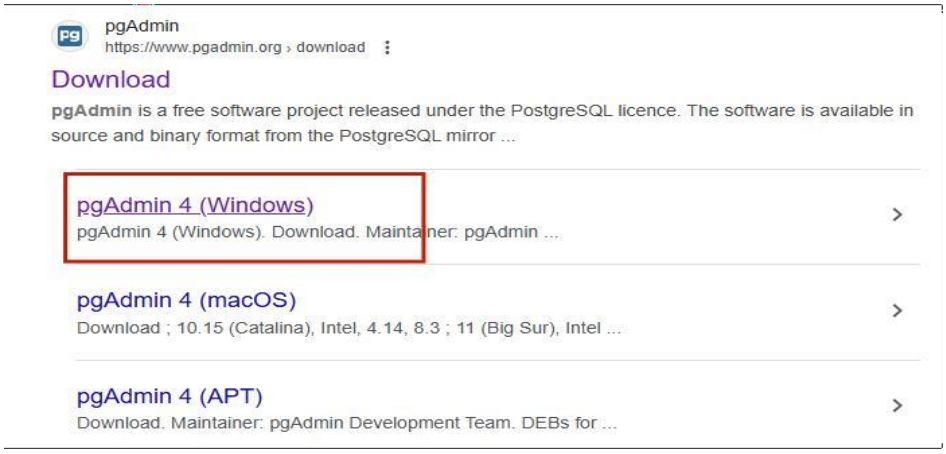
```
[postgres@localhost ~]$ /usr/local/pgsql/bin/psql
psql (15.3)
Type "help" for help.

postgres=#
```

➢ GUI Based Installation :

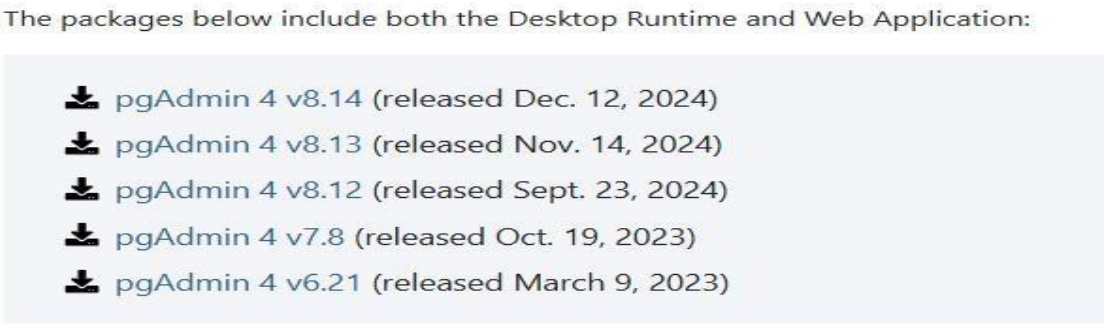
Steps:

1. **Install the postgresql server on OS using any method**

2. **Download the pgAdmin Tool from the browser**



3. **Select version**

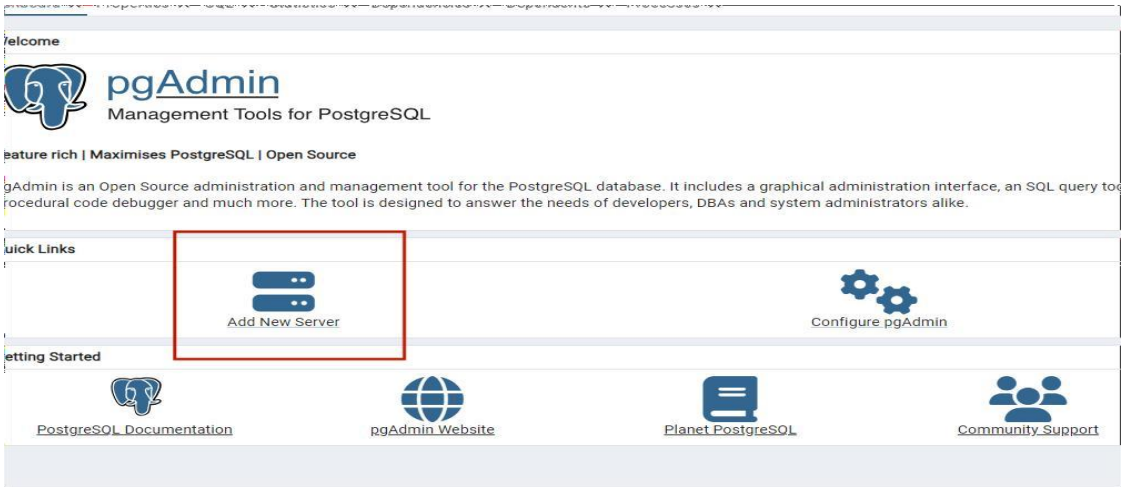


4. **Install pgAdmin (GUI Tool):**

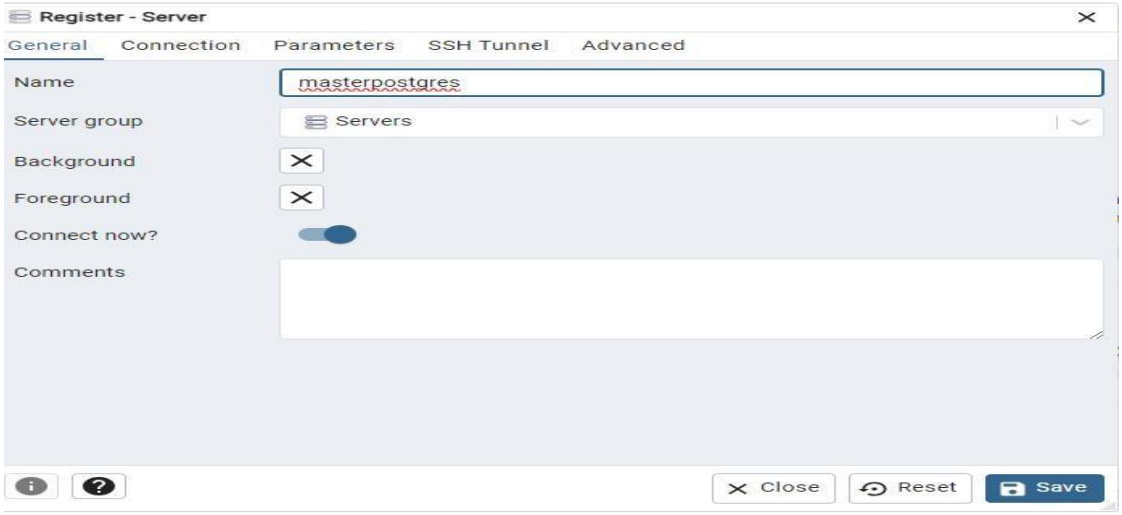
Next-Next-Next

5. Connect the Database

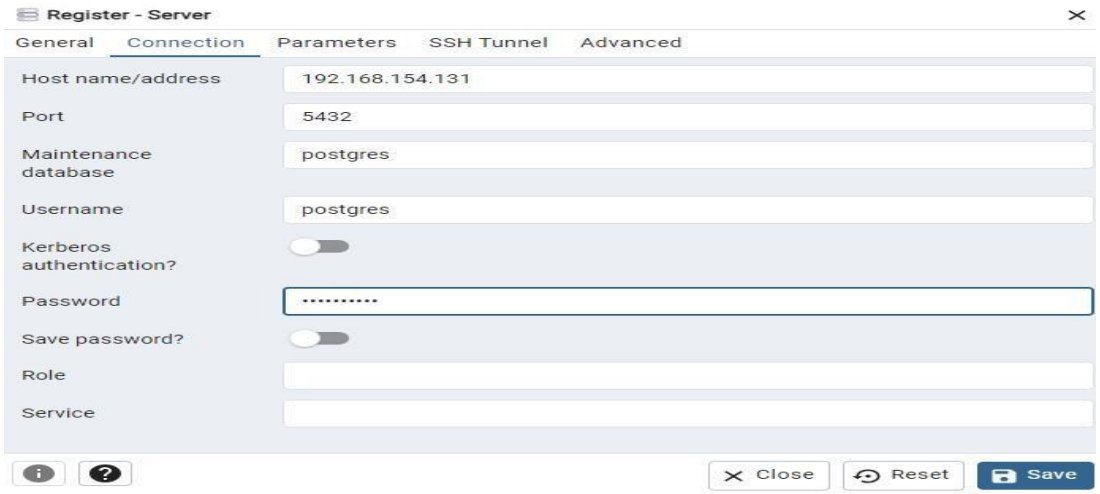
i. Click on Add new server



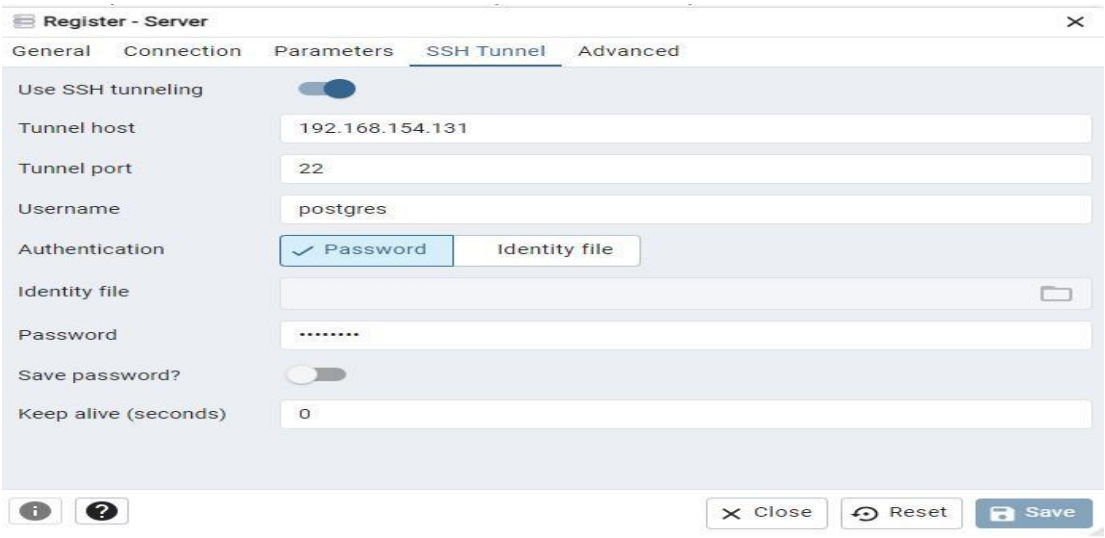
ii. Give any name to the server



iii. Next, go on the connection tab & fill the details (hostname/ip address , port no, username,password**)**



iv. Go to ssh tunnel fill details (Host IP , username, password**)**



v. Save & and get connected to the database