

# HOW TO INSTALL POSTGRESQL ON ROCKY LINUX

## Prerequisites

- An Rocky Linux system.
- A user with root or sudo privileges. This user will be used for installing new packages and make changes system-wide.

# Installing PostgreSQL on Rocky Linux

List out the available streams for the `postgresql` module using the `dnf` command:

```
dnf module list postgresql
```

## Output

```
[root@node02-postgresql ~]# dnf module list postgresql
```

Last metadata expiration check: 0:16:48 ago on Fri 07 Jun 2024 10:51:45 AM -03.

Rocky Linux 8 - AppStream

Name	Stream
Profiles	Summary
postgresql	9.6 client, server
[d]	PostgreSQL server and client module
postgresql	10 [d] client, server
[d]	PostgreSQL server and client module
postgresql	12 client, server
[d]	PostgreSQL server and client module
postgresql	13 client, server
[d]	PostgreSQL server and client module
postgresql	15 client, server
[d]	PostgreSQL server and client module
postgresql	16 client, server
[d]	PostgreSQL server and client module

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled

You can see in this output that there are four versions of PostgreSQL available from the **AppStream** repository: 9.6, 10, 12, and 13. The stream that provides Postgres version 10 is the default, as indicated by the [d] following it. To install that version, you could just run `sudo dnf install postgresql-server` and move on to the next step. However, even though version 10 is still maintained, this tutorial will install Postgres version 16.

To install PostgreSQL version 16, you must enable that version's module stream. When you enable a module stream, you override the default stream and make all of the packages related to the enabled stream available on the system. Note that only one stream of any given module can be enabled on a system at the same time.

To enable the module stream for Postgres version 16, run the following command:

```
sudo dnf module enable postgresql:16
```

When prompted, press `y` and then `ENTER` to confirm that you want to enable the stream:

#### Output

```
[root@node02-postgresql ~]# dnf module enable postgresql:16
```

```
Last metadata expiration check: 0:16:59 ago on Fri 07 Jun 2024 10:51:45 AM -03.
```

```
Dependencies resolved.
```

```
=====
=====
Package                                Architecture
Version                                Repository
=====
```

```
Enabling module streams:
```

```
  postgresql
  16
```

```
Transaction Summary
```

```
Is this ok [y/N]: y
```

```
Complete!
```

Install the latest version of PostgreSQL from the repository using the dnf command below.

```
dnf install postgresql-server.x86_64
```

```
[root@spf-prueba vmware-tools-distrib]# dnf install postgresql-server.x86_64
Última verificação de data de vencimento de metadados: 0:10:43 atrás em seg 08 nov 2021 19:29:55 -03.
Dependências resolvidas.
```

Pacote	Arquitetura	Versão	Repositório	Tamanho
Instalando:				
postgresql-server	x86_64	10.17-1.module+el8.4.0+548+9eccbe3f	appstream	5.1 M
Instalando dependências:				
libpq	x86_64	13.3-1.el8_4	appstream	196 k
postgresql	x86_64	10.17-1.module+el8.4.0+548+9eccbe3f	appstream	1.5 M
Ativando Fluxos de Módulos:				
postgresql		10		

```
Resumo da transação
Instalar 3 Pacotes

Tamanho total do download: 6.8 M
Tamanho depois de instalado: 26 M
Correto? [s/N]: s
Baixando pacotes:
(1/3): libpq-13.3-1.el8_4.x86_64.rpm 118 kB/s | 196 kB 00:01
(2/3): postgresql-10.17-1.module+el8.4.0+548+9eccbe3f.x86_64.rpm 692 kB/s | 1.5 MB 00:02
(3/3): postgresql-server-10.17-1.module+el8.4.0+548+9eccbe3f.x86_64.rpm 2.0 MB/s | 5.1 MB 00:02
Total 2.2 MB/s | 6.8 MB 00:03
```

## PostgreSQL Database Initialization

Next, after the PostgreSQL installation is complete, you must initialize the PostgreSQL configuration and then start and enable the PostgreSQL service.

1. Execute the following command to initialize the PostgreSQL database configuration.

```
postgresql-setup --initdb --unit postgresql
```

2. After that, start and enable the PostgreSQL service using the command below.

```
sudo systemctl enable postgresql
sudo systemctl start postgresql
```

Now the PostgreSQL service is active and running, and it will run automatically on every boot.

3. Now execute the command below to verify the PostgreSQL service.

```
systemctl status postgresql
```

If your PostgreSQL service is running, you will see the green output such as **"active(running)"** as below. Otherwise, you will see the red output such as **"failed"** following by the error message logs.

```

[[root@spf-prueba vmware-tools-distrib]# systemctl status postgresql
● postgresql.service - PostgreSQL database server
   Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2021-11-08 19:47:47 -03; 35s ago
     Process: 67960 ExecStartPre=/usr/libexec/postgresql-check-db-dir postgresql (code=exited, status=0/SUCCESS)
    Main PID: 67963 (postmaster)
      Tasks: 8 (limit: 49300)
     Memory: 16.0M
    CGroup: /system.slice/postgresql.service
            └─67963 /usr/bin/postmaster -D /var/lib/pgsql/data
              └─67964 postgres: logger process
                └─67966 postgres: checkpointing process
                  └─67967 postgres: writer process
                    └─67968 postgres: wal writer process
                      └─67969 postgres: autovacuum launcher process
                        └─67970 postgres: stats collector process
                          └─67971 postgres: bgworker: logical replication launcher

nov 08 19:47:47 spf-prueba systemd[1]: Starting PostgreSQL database server...
nov 08 19:47:47 spf-prueba postmaster[67963]: 2021-11-08 19:47:47.395 -03 [67963] LOG:  escuchando en la dirección IPv6 «::1», port 5432
nov 08 19:47:47 spf-prueba postmaster[67963]: 2021-11-08 19:47:47.395 -03 [67963] LOG:  escuchando en la dirección IPv4 «127.0.0.1», port 5432
nov 08 19:47:47 spf-prueba postmaster[67963]: 2021-11-08 19:47:47.397 -03 [67963] LOG:  escuchando en el socket Unix «/var/run/postgresql/.s.PGSQL.5432»
nov 08 19:47:47 spf-prueba postmaster[67963]: 2021-11-08 19:47:47.400 -03 [67963] LOG:  escuchando en el socket Unix «/tmp/.s.PGSQL.5432»
nov 08 19:47:47 spf-prueba postmaster[67963]: 2021-11-08 19:47:47.413 -03 [67963] LOG:  redirigiendo la salida del registro al proceso recolector de registro
nov 08 19:47:47 spf-prueba postmaster[67963]: 2021-11-08 19:47:47.413 -03 [67963] HINT:  La salida futura del registro aparecerá en el directorio «log».
nov 08 19:47:47 spf-prueba systemd[1]: Started PostgreSQL database server.

```

## Securing PostgreSQL Deployment

During the installation, PostgreSQL will create a new system user and database user name as "**postgres**". And for this stage, you will be setting up a new password for the "**postgres**" user, both for the **system user** and **database user**.

1. Change the password for default system user "postgres" using the following command.

```
passwd postgres
```

Now type the new password for the system user "postgres".

2. Next, to change the password for the "postgres" database user, you must log in to the PostgreSQL shell.

First, log in as a system user "postgres" using the following command.

```
su - postgres
```

Now login to the PostgreSQL shell using the psql command below.

```
psql
```

Execute the following query to create a new password for the default "postgres" database user.

```
ALTER USER postgres WITH PASSWORD 'strongpostgrespassword';
```

Change the string 'strongpostgrespassword' to your own password. Now type exit and press "**Ctrl+d**" to exit and log out from the 'postgres' user shell.

```
[root@spf-prueba vmware-tools-distrib]# passwd postgres
Mudando senha para o usuário postgres.
Nova senha:
Redigite a nova senha:
passwd: todos os tokens de autenticações foram atualizados com sucesso.
[root@spf-prueba vmware-tools-distrib]# su - postgres
[postgres@spf-prueba ~]$ Now login to the PostgreSQL shell using the psql command below.
-bash: Now: no se encontró la orden
[postgres@spf-prueba ~]$ psql
psql (10.17)
Digite «help» para obtener ayuda.

postgres=# ALTER USER postgres WITH PASSWORD 'strongpostgrespassword';
ALTER ROLE
postgres=# exit
postgres-# logout
postgres-# \q
[postgres@spf-prueba ~]$ logout
[root@spf-prueba vmware-tools-distrib]# █
```

## Change Authentication Method

By default, local PostgreSQL users will connect to the PostgreSQL shell using the 'peer' method. The peer authentication method will work only for local connections. In the development environment, you can use this type of authentication, but for production, consider using the password-based authentication method.

For this stage, you will learn how to change the default peer authentication method to password authentication using 'md5'.

1. First, log in to the PostgreSQL shell using the following command.

```
sudo -u postgres psql
```

Now execute the following query to check the location of the PostgreSQL configuration 'pg\_hba.conf'.

```
SHOW hba_file;
SHOW password_encryption;
```

You will see the output as below.

```
[root@spf-prueba vmware-tools-distrib]# sudo -u postgres psql
psql (10.17)
Digite "help" para ajuda.
```

```
postgres=# SHOW hba_file;
          hba_file
-----
/var/lib/pgsql/data/pg_hba.conf
(1 registro)
```

```
postgres=# SHOW password_encryption;
password_encryption
-----
md5
(1 registro)
```

```
postgres=# █
```

You will notice the PostgreSQL configuration "pg\_hba.conf" are located at the '**/var/lib/pgsql/data**' directory, and the default password encryption for PostgreSQL on RHEL based operating system is '**md5**'.

Now type '\q' to exit and quit the PostgreSQL shell.

2. Next, change the working directory to '**/var/lib/pgsql/data**' and edit the configuration '**pg\_hba.conf**' using nano editor.

```
cd /var/lib/pgsql/data/
vi pg_hba.conf
```

#	TYPE	DATABASE	USER	ADDRESS	METHOD
# "local" is for Unix domain socket connections only					
local	all		all		peer
# IPv4 local connections:					
host	all		all	127.0.0.1/32	ident
# IPv6 local connections:					
host	all		all	:::1/128	ident
# Allow replication connections from localhost, by a user with the					
# replication privilege.					
local	replication		all		peer
host	replication		all	127.0.0.1/32	ident
host	replication		all	:::1/128	ident
1 change; before #1 8 seconds ago					

At the bottom of the line, change the local authentication method to '**md5**' as below.

#	TYPE	DATABASE	USER	ADDRESS	METHOD
# "local" is for Unix domain socket connections only					
local	all		all		md5
# IPv4 local connections:					
host	all		all	127.0.0.1/32	md5
# IPv6 local connections:					
host	all		all	:::1/128	md5
# Allow replication connections from localhost, by a user with the					

Now press '**ESC**', type ':**wq**', and press "**Enter**" to save and exit.

Using this configuration, you will be prompted for the password to log in to the PostgreSQL shell.

3. Next, apply the new configuration by restarting the PostgreSQL service using the following command.

```
systemctl restart postgresql
```

Now every time you want to access the PostgreSQL shell, you must type the password for authentication.

4. To make sure of the password authentication configuration, log in to the PostgreSQL shell using the following command.

```
su - postgres
psql
```

Now you will be asked for a password for the default user 'postgres'.

Type the password for the 'postgres' database user and press '**Enter**'. If your password is correct, you will see the PostgreSQL shell as follows. Otherwise, you will see the '**FATAL**' error because the password is incorrect.

```
[postgres@spf-prueba ~]$ psql
Contraseña:
psql: FATAL:  la autenticación password falló para el usuario «postgres»
[postgres@spf-prueba ~]$ psql
```

Additionally, you can use the one-line command to log in to the PostgreSQL shell as below.

```
# Log in as default "postgres" user
sudo -u postgres psql
```

```
# Log in as another user
sudo -u postgres psql -U username
```

## Creating New User and Database for your Application

At this stage, you will learn how to create a new user and database on PostgreSQL.

1. Log in to the PostgreSQL shell by executing the command below.

```
sudo -u postgres psql
```

Now type the password for PostgreSQL user '**postgres**'.

2. Run the PostgreSQL query below to create a new user 'johndoe' with the password 'johndoestrongpassword' and give the user privileges for creating a new database and role.

```
CREATE USER spf WITH
CREATEDB
CREATEROLE
PASSWORD 'spfstrongpassword';
```

After that, verify the new user using the following query.

```
\du
```

Now you will see the new user 'spf' with the list of roles '**Create role**' and '**Create DB**' as below.

```
[postgres=# \du
```

Lista de roles		
Nome da role	Atributos	Membro de
postgres	Super-usuário, Cria role, Cria BD, Replicação, Ignora RLS	{}
spf	Cria role, Cria BD	{}

3. Next, to create a new user database on PostgreSQL, run the following query.

```
CREATE DATABASE spf OWNER spf;
```

Now verify the new database using the following query.



```
\l
```

And you will see the new database 'spf' with the owner 'spf' as the screenshot below.

```
postgres=# \l
```

Lista dos bancos de dados					
Nome	Dono	Codificação	Collate	Ctype	Privilégios de acesso
postgres	postgres	UTF8	es_MX.UTF-8	es_MX.UTF-8	
spf	spf	UTF8	es_MX.UTF-8	es_MX.UTF-8	
template0	postgres	UTF8	es_MX.UTF-8	es_MX.UTF-8	=c/postgres + postgres=CTc/postgres
template1	postgres	UTF8	es_MX.UTF-8	es_MX.UTF-8	=c/postgres + postgres=CTc/postgres

(4 registros)

```
postgres=#
```

## Reference link

<https://www.howtoforge.com/how-to-install-postgresql-on-rocky-linux/>

<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-postgresql-on-rocky-linux-8>

Revision #5

Created 8 November 2021 22:23:54 by Rondineli G. Saad

Updated 7 June 2024 14:21:10 by Rondineli G. Saad