Install CYBERTEC PostgreSQL 14 TDE (RHEL 8)

1. Create a new repository file

```
vi /etc/yum.repos.d/cybertec-pg14.repo
```

2. Add the following lines with your Username and password

```
[cybertec_pg14]
name=Cybertec PostgreSQL 14 for RHEL/CentOS $releasever -
$basearch
baseurl=http://repository.cybertec.at/non-public/14/redhat
/$releasever/$basearch
username=myusername
password=mypassword
enabled=1
gpgcheck=0
```

3. Run

```
yum update
```

4. Verification, you should get a output like this with the line “cybertec_pg14”

```
[root@pgee yum.repos.d]# yum update
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: centos.anexia.at
 * extras: centos.anexia.at
 * updates: centos.anexia.at
base
| 3.6 kB  00:00:00
cybertec_pg14
| 3.0 kB  00:00:00
extras
| 2.9 kB  00:00:00
updates
| 2.9 kB  00:00:00
No packages marked for update
[root@pgee yum.repos.d]#
```

Installation

If you are using Red Hat/CentOS/Rocky Linux 8, you need to disable the built-in Postgres instance

```
sudo dnf -qy module disable postgresql
```

1. Installing PostgreSQL TDE with yum Package Manager

```
yum install postgresql14-tde-server -y
```

Additional Packages

Packagename	Description
postgresql14-tde.x86_64	PostgreSQL client programs and libraries
postgresql14-tde-contrib.x86_64	Contributed source and binaries distributed with PostgreSQL
postgresql14-tde-devel.x86_64	PostgreSQL development header files and libraries
postgresql14-tde-docs.x86_64	Extra documentation for PostgreSQL
postgresql14-tde-libs.x86_64	The shared libraries required for any PostgreSQL clients
postgresql14-tde-llvmjit.x86_64	Just-in-time compilation support for PostgreSQL
postgresql14-tde-plperl.x86_64	The Perl procedural language for PostgreSQL
postgresql14-tde-plperl.x86_64	The Perl procedural language for PostgreSQL
postgresql14-tde-plpython.x86_64	The Python procedural language for PostgreSQL
postgresql14-tde-pltcl.x86_64	The Tcl procedural language for PostgreSQL
postgresql14-tde-server.x86_64	The programs needed to create and run a PostgreSQL server
postgresql14-tde-test.x86_64	• The test suite distributed with PostgreSQL

Default Path's

Type	Path
Binaries	/usr/pgsql-14-tde/bin/
Libraries	/usr/pgsql-14-tde/lib/
Shared	/usr/pgsql-14-tde/shared/
PGDATA	/var/lib/pgsql/14/data

Database Instance initialisation

Encryption Key Script

Create a simple Bash script that returns a hexadecimal sting with a length of 32.

1. Generate a Key

```
openssl rand -hex 16
```

   a. For Example:

```
34a27e892ec49d10aae6edfaccb7f224
```

2. Create a Bash script

```
vi /var/lib/pgsql/tde_key.sh
```

3. with the following content

```
#!/usr/bin/bash
set -euo pipefail
echo '34a27e892ec49d10aae6edfaccb7f224'
```

4. Save it and change the Permissions

```
chmod 700 /var/lib/pgsql/tde_key.sh
```

Create a encrypted Database Instance

To create an encrypted Database instance, call `initdb` with the option `-K` and the path to the command that will retrieve the encryption key.

```
/usr/pgsql-14-tde/bin/initdb -D /var/lib/pgsql/14/data -K /var/lib
/pgsql/tde_key.sh --data-checksums
```

`initdb` puts the encryption key command into the `postgresql.conf` file, as the value of the `encryption_key_command` parameter. That ensures that you don't need to pass it to `pg_ctl`each time you want to start the server.

More about Transparent Database Encryption

After the Database is successfully created you will see something like that

```
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.
Data encryption is enabled.

fixing permissions on existing directory /var/lib/pgsql/14/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 ... Europe/Vienna
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
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 /var/lib/pgsql/14/data -l logfile start
```

Before you start check if in the Output contains this line `Data encryption is enabled.` and `Success. You can now start the database server using: /usr/pgsql-14-tde/bin/pg_ctl -D /var/lib/pgsql/14/data -l logdatei start`

Ready to start

Start your PostgreSQL TDE Edition Database.