15- backup using pgbackrest

pgBackRest is a backup solution specifically designed for disaster recovery in PostgreSQL databases. Unlike other backup solutions like pg_dump, pgBackRest supports point-in-time recovery and offers many additional features.

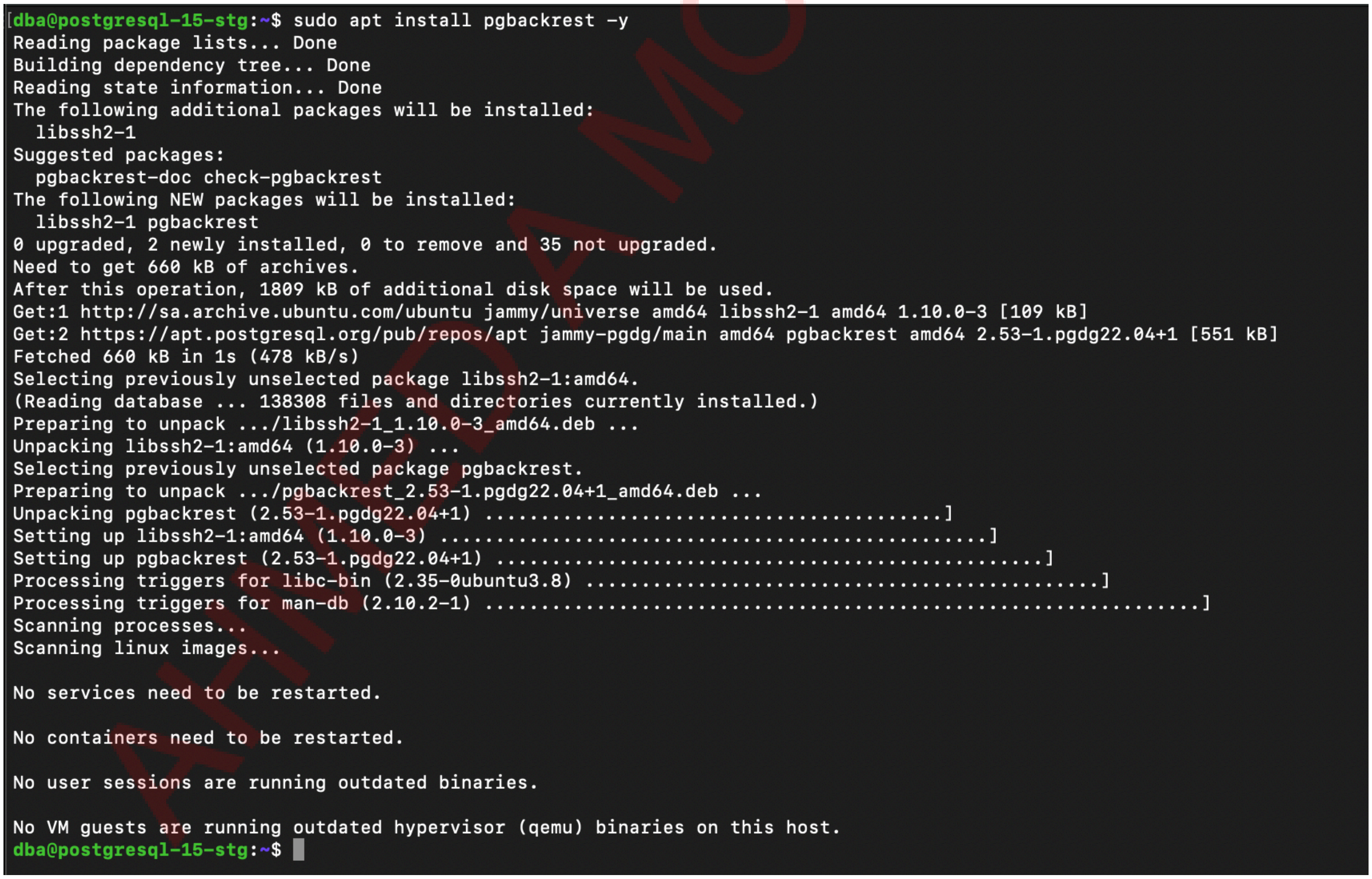
pgBackRest can perform three types of backups:

- Full backups - these copy the entire contents of the database cluster to the backup.
- Differential backups - this copies only the database cluster files that have changed since the last full backup
- Incremental backups - which copy only the database cluster files that have changed since the last full, differential, or incremental.
- Creating a Delta Restore which will use database files already present and updated based on WAL segments. This makes potential restores much faster, especially if you have a large database and don't want to restore the entire thing.
- Letting you have multiple backup repositories - say one local or one remote for redundancy.
  With all its features, pgBackRest is undoubtedly the best option for backing up PostgreSQL.

installing and configuring

We will begin by installing pgBackRest on an Ubuntu Server 22 using the following command:

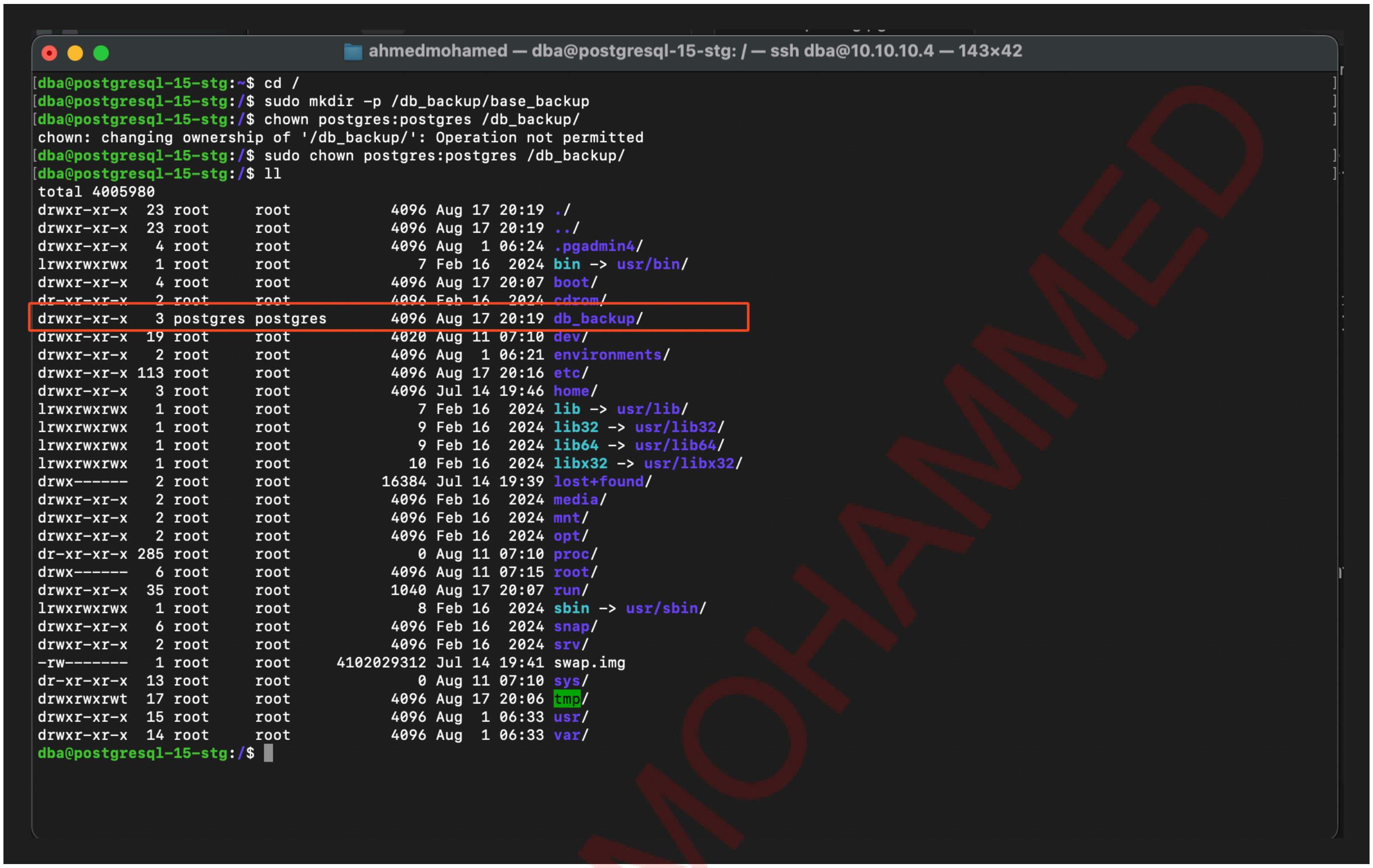
```
sudo apt install pgbackrest
```



```
dba@postgresql-15-stg:~$ sudo apt install pgbackrest -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libssh2-1
Suggested packages:
  pgbackrest-doc check-pgbackrest
The following NEW packages will be installed:
  libssh2-1 pgbackrest
0 upgraded, 2 newly installed, 0 to remove and 35 not upgraded.
Need to get 660 kB of archives.
After this operation, 1809 kB of additional disk space will be used.
Get:1 http://sa.archive.ubuntu.com/ubuntu jammy/universe amd64 libssh2-1 amd64 1.10.0-3 [109 kB]
Get:2 https://apt.postgresql.org/pub/repos/apt jammy-pgdg/main amd64 pgbackrest amd64 2.53-1.pgdg22.04+1 [551 kB]
Fetched 660 kB in 1s (478 kB/s)
Selecting previously unselected package libssh2-1:amd64.
(Reading database ... 138308 files and directories currently installed.)
Preparing to unpack .../libssh2-1_1.10.0-3_amd64.deb ...
Unpacking libssh2-1:amd64 (1.10.0-3) ...
Selecting previously unselected package pgbackrest.
Preparing to unpack .../pgbackrest_2.53-1.pgdg22.04+1_amd64.deb ...
Unpacking pgbackrest (2.53-1.pgdg22.04+1) ..........................................]
Setting up libssh2-1:amd64 (1.10.0-3) ..................................................]
Setting up pgbackrest (2.53-1.pgdg22.04+1) ..................................................]
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...............................................]
Processing triggers for man-db (2.10.2-1) .................................................................]
Scanning processes...
Scanning linux images...

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
dba@postgresql-15-stg:~$
```

Next, we will create a directory to store the backups. It is always best practice to store backups on a separate mount point from the root directory to avoid interfering with the operating system.

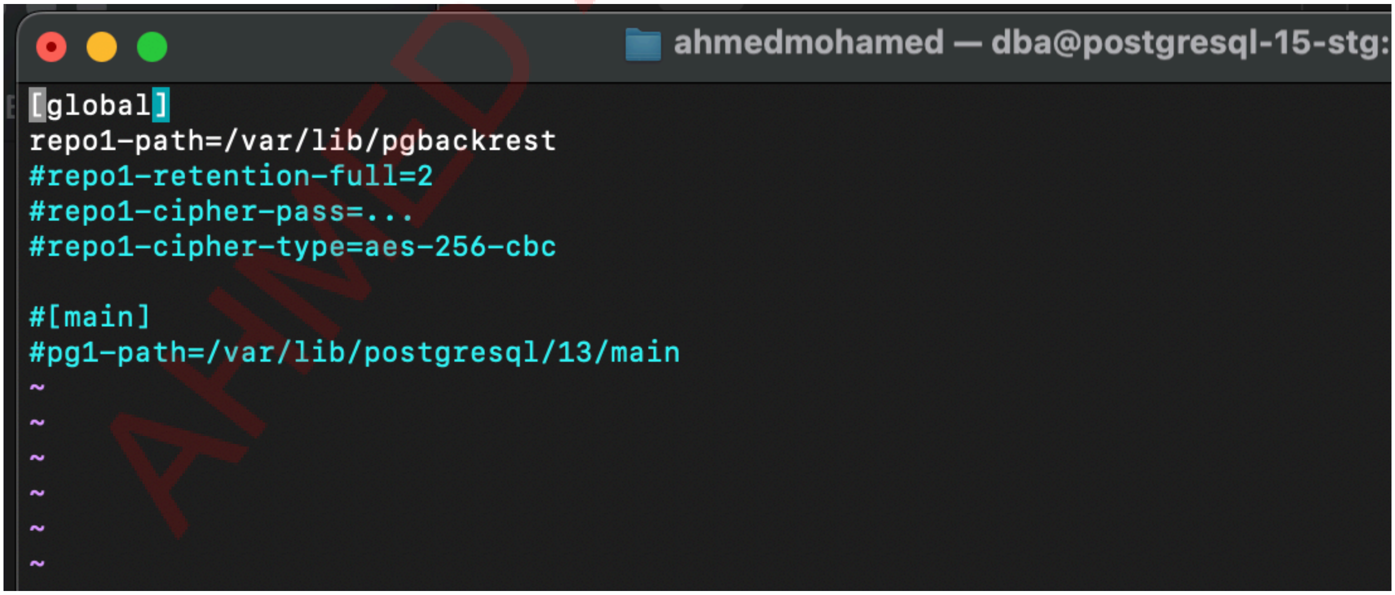
```
mkdir -p /db_backup/base_backup

chown postgres:postgres /db_backup/
```


ahmedmohamed — dba@postgresql-15-stg: / — ssh dba@10.10.10.4 — 143×42

```
dba@postgresql-15-stg:~$ cd /
dba@postgresql-15-stg:/$ sudo mkdir -p /db_backup/base_backup
dba@postgresql-15-stg:/$ chown postgres:postgres /db_backup/
chown: changing ownership of '/db_backup/': Operation not permitted
dba@postgresql-15-stg:/$ sudo chown postgres:postgres /db_backup/
dba@postgresql-15-stg:/$ ll
total 4005980
drwxr-xr-x  23 root     root           4096 Aug 17 20:19 ./
drwxr-xr-x  23 root     root           4096 Aug 17 20:19 ../
drwxr-xr-x   4 root     root           4096 Aug  1 06:24 .pgadmin4/
lrwxrwxrwx   1 root     root              7 Feb 16  2024 bin -> usr/bin/
drwxr-xr-x   4 root     root           4096 Aug 17 20:07 boot/
dr-xr-xr-x   2 root     root           4096 Feb 16  2024 cdrom/
drwxr-xr-x   3 postgres postgres       4096 Aug 17 20:19 db_backup/
drwxr-xr-x  19 root     root           4020 Aug 11 07:10 dev/
drwxr-xr-x   2 root     root           4096 Aug  1 06:21 environments/
drwxr-xr-x 113 root     root           4096 Aug 17 20:16 etc/
drwxr-xr-x   3 root     root           4096 Jul 14 19:46 home/
lrwxrwxrwx   1 root     root              7 Feb 16  2024 lib -> usr/lib/
lrwxrwxrwx   1 root     root              9 Feb 16  2024 lib32 -> usr/lib32/
lrwxrwxrwx   1 root     root              9 Feb 16  2024 lib64 -> usr/lib64/
lrwxrwxrwx   1 root     root             10 Feb 16  2024 libx32 -> usr/libx32/
drwx------   2 root     root          16384 Jul 14 19:39 lost+found/
drwxr-xr-x   2 root     root           4096 Feb 16  2024 media/
drwxr-xr-x   2 root     root           4096 Feb 16  2024 mnt/
drwxr-xr-x   2 root     root           4096 Feb 16  2024 opt/
dr-xr-xr-x 285 root     root              0 Aug 11 07:10 proc/
drwx------   6 root     root           4096 Aug 11 07:15 root/
drwxr-xr-x  35 root     root           1040 Aug 17 20:07 run/
lrwxrwxrwx   1 root     root              8 Feb 16  2024 sbin -> usr/sbin/
drwxr-xr-x   6 root     root           4096 Feb 16  2024 snap/
drwxr-xr-x   2 root     root           4096 Feb 16  2024 srv/
-rw-------   1 root     root     4102029312 Jul 14 19:41 swap.img
dr-xr-xr-x  13 root     root              0 Aug 11 07:10 sys/
drwxrwxrwt  17 root     root           4096 Aug 17 20:06 tmp/
drwxr-xr-x  15 root     root           4096 Aug  1 06:33 usr/
drwxr-xr-x  14 root     root           4096 Aug  1 06:33 var/
dba@postgresql-15-stg:/$
```

Now, we will configure pgBackRest. Open the `/etc/pgbackrest.conf` file using your preferred text editor. I will be using `vi`

```
sudo vi /etc/pgbackrest.conf
```



```
[global]
repo1-path=/var/lib/pgbackrest
#repo1-retention-full=2
#repo1-cipher-pass=...
#repo1-cipher-type=aes-256-cbc

#[main]
#pg1-path=/var/lib/postgresql/13/main
~
~
~
~
~
~
```

We will remove the `#` from `[main]` and `pg1-path` and update them with the correct data directory for our PostgreSQL database.

If you don't know the data directory path, use the command `pg_lscluster` to find it. We will also add the user that pgBackRest will use to connect to PostgreSQL and specify the port number.

In the `[global]` section, we will update the retention settings for full and incremental backups, and specify the location for storing the backups.

Below is the complete configuration I have specified for pgBackRest:

```
[global]

repo1-block=y

repo1-bundle=y

repo1-path=/db_backup/base_backup

repo1-retention-diff=1

repo1-retention-full=2

start-fast=y

compress-level=6



[main]

pg1-path=/var/lib/postgresql/15/main

pg1-user=postgres

pg1-port=5432
```

I won't go into detail about each option in the configuration, but in summary, I have specified the compression level as well as the retention settings for both full and incremental backups.

to view other option please visit EDB website https://www.enterprisedb.com/docs/supported-open-source/pgbackrest/04-recommended_settings/

Now, we'll update the PostgreSQL configuration related to archiving, which is necessary for point-in-time recovery.

Open the `postgresql.conf` file using either `vi` or `nano`.

```
sudo vi /etc/postgresql/15/main/postgresql.conf
```

Update the following parameters as follows:

```
listen_addresses = '*'

wal_level = replica

archive_mode = on

archive_command = 'pgbackrest --stanza=main archive-push %p'
```

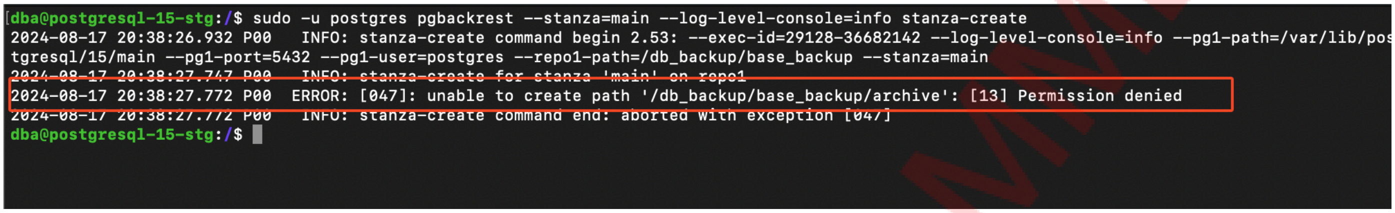
after that restart postgresql services

```
systemctl restart postgresql
```

creating stanza

A stanza is the configuration for a PostgreSQL database cluster that defines its location. To start, we will use the following command, which will create a stanza based on the configuration we have specified:

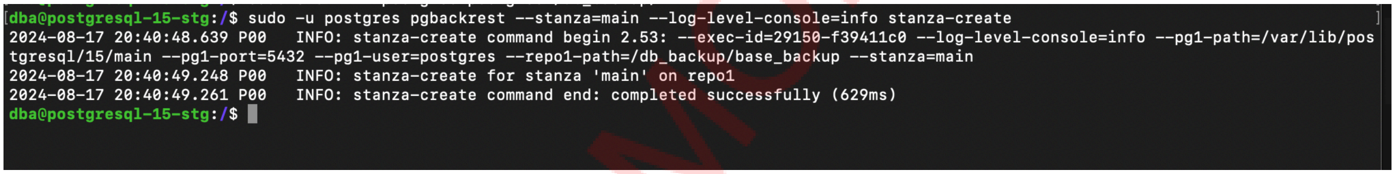
```
sudo -u postgres pgbackrest --stanza=main --log-level-console=info stanza-create
```



```
dba@postgresql-15-stg:/$ sudo -u postgres pgbackrest --stanza=main --log-level-console=info stanza-create
2024-08-17 20:38:26.932 P00   INFO: stanza-create command begin 2.53: --exec-id=29128-36682142 --log-level-console=info --pg1-path=/var/lib/pos
tgresql/15/main --pg1-port=5432 --pg1-user=postgres --repo1-path=/db_backup/base_backup --stanza=main
2024-08-17 20:38:27.747 P00   INFO: stanza-create for stanza 'main' on repo1
2024-08-17 20:38:27.772 P00  ERROR: [047]: unable to create path '/db_backup/base_backup/archive': [13] Permission denied
2024-08-17 20:38:27.772 P00   INFO: stanza-create command end: aborted with exception [047]
dba@postgresql-15-stg:/$
```

You may encounter a similar error. If so, simply apply ownership to the `postgres` user again and rerun the command; it should then work.

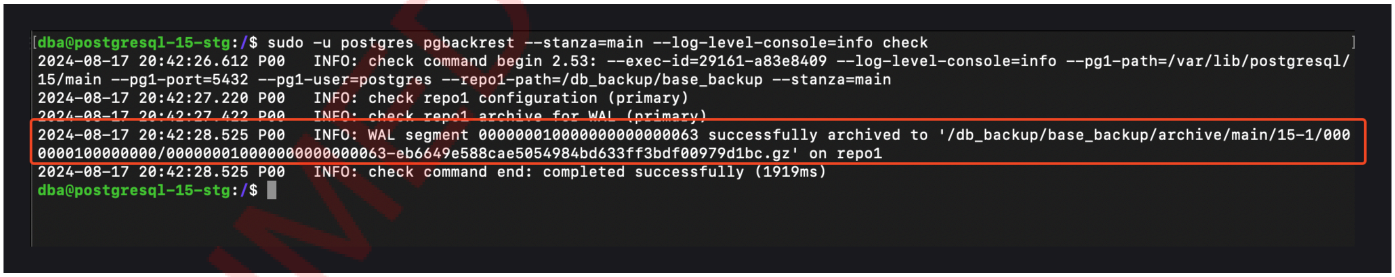
```
sudo chown -R postgres:postgres /db_backup/
```



```
dba@postgresql-15-stg:/$ sudo -u postgres pgbackrest --stanza=main --log-level-console=info stanza-create
2024-08-17 20:40:48.639 P00   INFO: stanza-create command begin 2.53: --exec-id=29150-f39411c0 --log-level-console=info --pg1-path=/var/lib/pos
tgresql/15/main --pg1-port=5432 --pg1-user=postgres --repo1-path=/db_backup/base_backup --stanza=main
2024-08-17 20:40:49.248 P00   INFO: stanza-create for stanza 'main' on repo1
2024-08-17 20:40:49.261 P00   INFO: stanza-create command end: completed successfully (629ms)
dba@postgresql-15-stg:/$
```

The following command will check our configuration and ensure that our backup will be taken without errors:

```
sudo -u postgres pgbackrest --stanza=main --log-level-console=info check
```



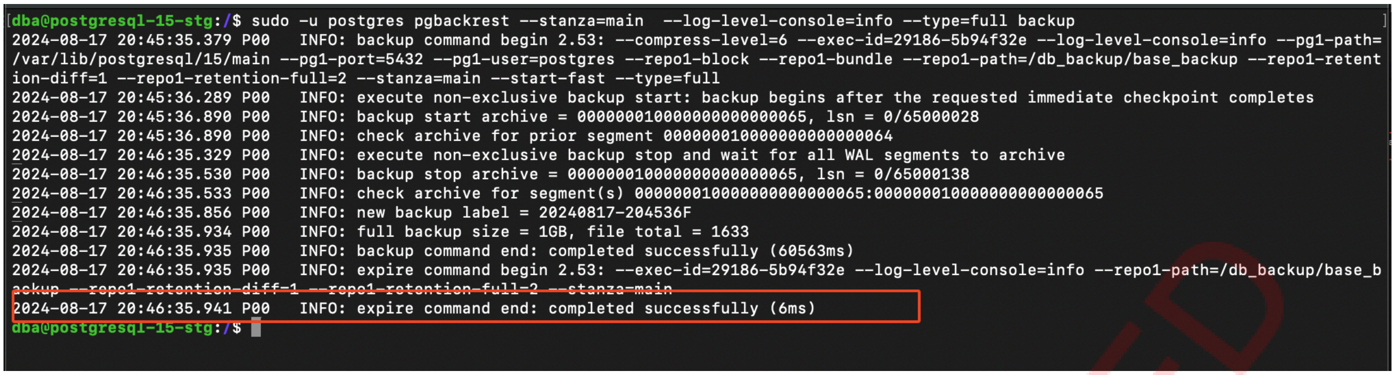
```
dba@postgresql-15-stg:/$ sudo -u postgres pgbackrest --stanza=main --log-level-console=info check
2024-08-17 20:42:26.612 P00   INFO: check command begin 2.53: --exec-id=29161-a83e8409 --log-level-console=info --pg1-path=/var/lib/postgresql/
15/main --pg1-port=5432 --pg1-user=postgres --repo1-path=/db_backup/base_backup --stanza=main
2024-08-17 20:42:27.220 P00   INFO: check repo1 configuration (primary)
2024-08-17 20:42:27.422 P00   INFO: check repo1 archive for WAL (primary)
2024-08-17 20:42:28.525 P00   INFO: WAL segment 000000010000000000000063 successfully archived to '/db_backup/base_backup/archive/main/15-1/000
0000100000000/000000010000000000000063-eb6649e588cae5054984bd633ff3bdf00979d1bc.gz' on repo1
2024-08-17 20:42:28.525 P00   INFO: check command end: completed successfully (1919ms)
dba@postgresql-15-stg:/$
```

We are all set now. We can start taking a full backup and test the restoration process.

backup and restore

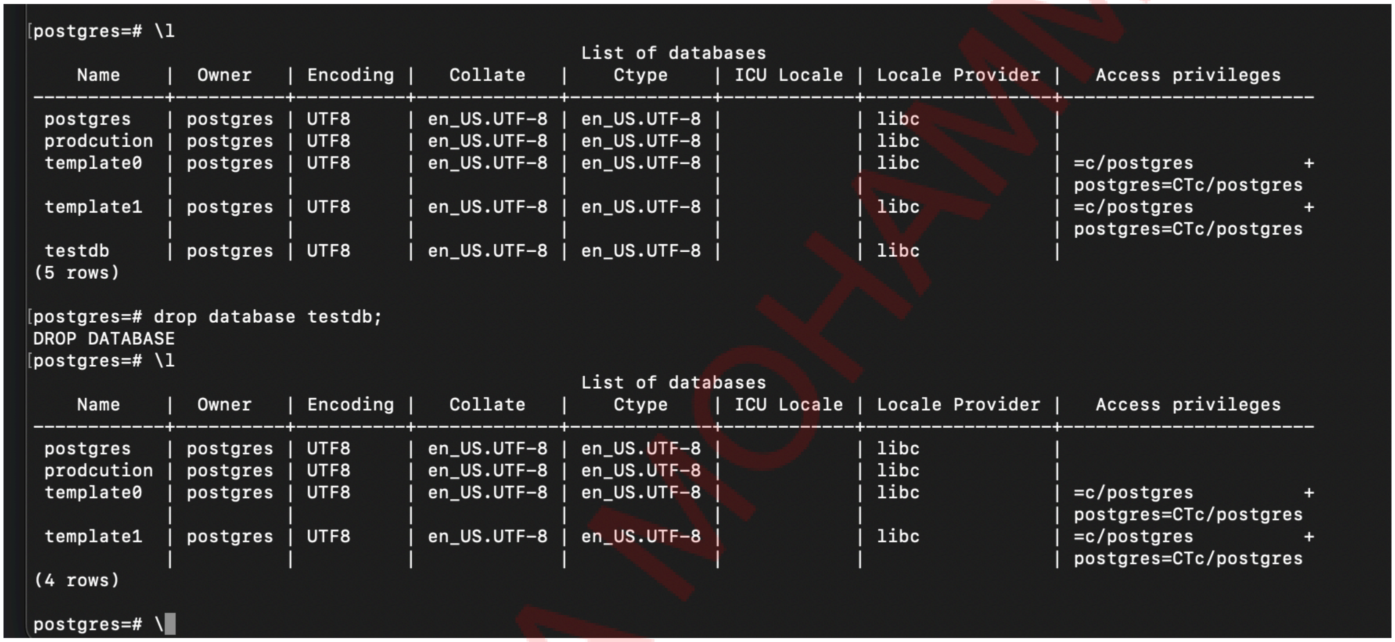
For testing purposes, I will log in to `psql` and drop the database. However, before doing that, I will take a full backup using the following command:

```
sudo -u postgres pgbackrest --stanza=main  --log-level-console=info --type=full backup
```



```
[dba@postgresql-15-stg:/$ sudo -u postgres pgbackrest --stanza=main  --log-level-console=info --type=full backup
2024-08-17 20:45:35.379 P00   INFO: backup command begin 2.53: --compress-level=6 --exec-id=29186-5b94f32e --log-level-console=info --pg1-path=/var/lib/postgresql/15/main --pg1-port=5432 --pg1-user=postgres --repo1-block --repo1-bundle --repo1-path=/db_backup/base_backup --repo1-retention-diff=1 --repo1-retention-full=2 --stanza=main --start-fast --type=full
2024-08-17 20:45:36.289 P00   INFO: execute non-exclusive backup start: backup begins after the requested immediate checkpoint completes
2024-08-17 20:45:36.890 P00   INFO: backup start archive = 000000010000000000000065, lsn = 0/65000028
2024-08-17 20:45:36.890 P00   INFO: check archive for prior segment 000000010000000000000064
2024-08-17 20:46:35.329 P00   INFO: execute non-exclusive backup stop and wait for all WAL segments to archive
2024-08-17 20:46:35.530 P00   INFO: backup stop archive = 000000010000000000000065, lsn = 0/65000138
2024-08-17 20:46:35.533 P00   INFO: check archive for segment(s) 000000010000000000000065:000000010000000000000065
2024-08-17 20:46:35.856 P00   INFO: new backup label = 20240817-204536F
2024-08-17 20:46:35.934 P00   INFO: full backup size = 1GB, file total = 1633
2024-08-17 20:46:35.935 P00   INFO: backup command end: completed successfully (60563ms)
2024-08-17 20:46:35.935 P00   INFO: expire command begin 2.53: --exec-id=29186-5b94f32e --log-level-console=info --repo1-path=/db_backup/base_backup --repo1-retention-diff=1 --repo1-retention-full=2 --stanza=main
2024-08-17 20:46:35.941 P00   INFO: expire command end: completed successfully (6ms)
dba@postgresql-15-stg:/$
```

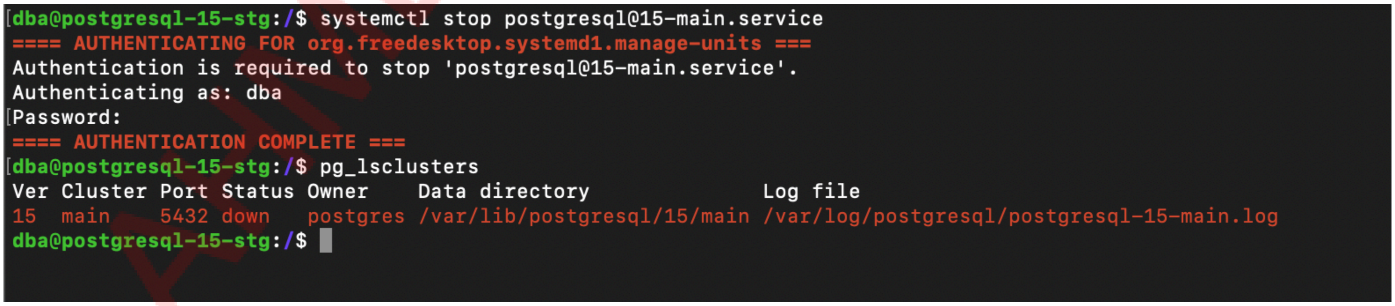
The backup has been taken successfully. Now, let's test the restoration process. Before doing so, I will log in to `psql` and drop a random database.



```
[postgres=# \l
                                                    List of databases
    Name    |  Owner   | Encoding |   Collate   |    Ctype    | ICU Locale | Locale Provider |   Access privileges
------------+----------+----------+-------------+-------------+------------+-----------------+-----------------------
 postgres   | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            |
 prodcution | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            |
 template0  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            | =c/postgres          +
            |          |          |             |             |            |                 | postgres=CTc/postgres
 template1  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            | =c/postgres          +
            |          |          |             |             |            |                 | postgres=CTc/postgres
 testdb     | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            |
(5 rows)

[postgres=# drop database testdb;
DROP DATABASE
[postgres=# \l
                                                    List of databases
    Name    |  Owner   | Encoding |   Collate   |    Ctype    | ICU Locale | Locale Provider |   Access privileges
------------+----------+----------+-------------+-------------+------------+-----------------+-----------------------
 postgres   | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            |
 prodcution | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            |
 template0  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            | =c/postgres          +
            |          |          |             |             |            |                 | postgres=CTc/postgres
 template1  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |            | libc            | =c/postgres          +
            |          |          |             |             |            |                 | postgres=CTc/postgres
(4 rows)

postgres=# \
```

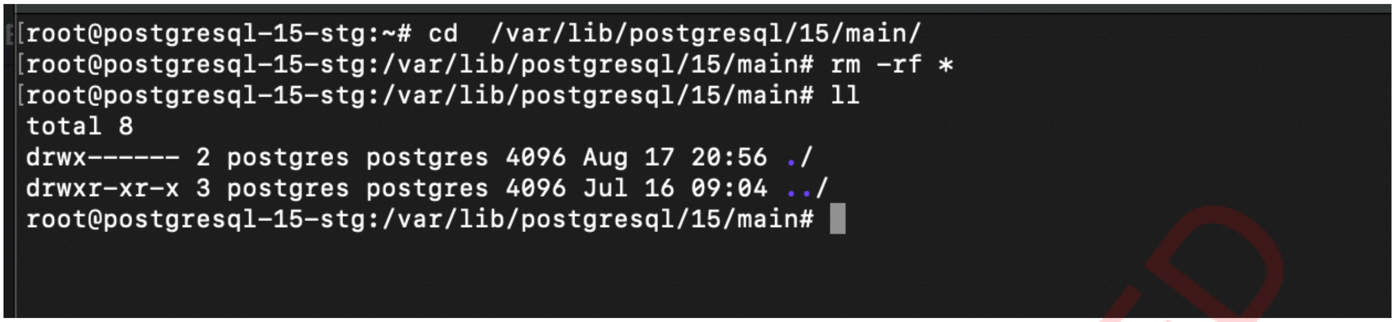
Let's restore our backup. One key point to remember is to ensure that the PostgreSQL service is shut down and verify that the database cluster is not running.

```
systemctl stop postgresql

pg_lscluster
```



```
[dba@postgresql-15-stg:/$ systemctl stop postgresql@15-main.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to stop 'postgresql@15-main.service'.
Authenticating as: dba
[Password:
==== AUTHENTICATION COMPLETE ===
[dba@postgresql-15-stg:/$ pg_lsclusters
Ver Cluster Port Status Owner    Data directory              Log file
15  main    5432 down   postgres /var/lib/postgresql/15/main /var/log/postgresql/postgresql-15-main.log
dba@postgresql-15-stg:/$
```

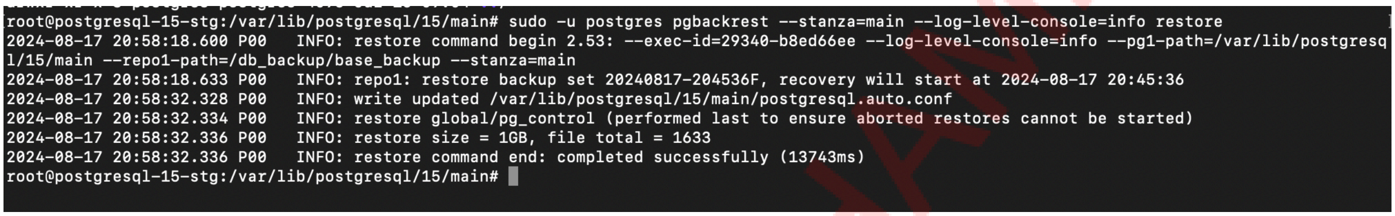
Now, go to your data directory and remove all the files there.



```
[root@postgresql-15-stg:~# cd  /var/lib/postgresql/15/main/
[root@postgresql-15-stg:/var/lib/postgresql/15/main# rm -rf *
[root@postgresql-15-stg:/var/lib/postgresql/15/main# ll
total 8
drwx------ 2 postgres postgres 4096 Aug 17 20:56 ./
drwxr-xr-x 3 postgres postgres 4096 Jul 16 09:04 ../
root@postgresql-15-stg:/var/lib/postgresql/15/main# 
```

Run the following command to restore the latest backup. Note that by default, pgBackRest will restore the most recent backup available.

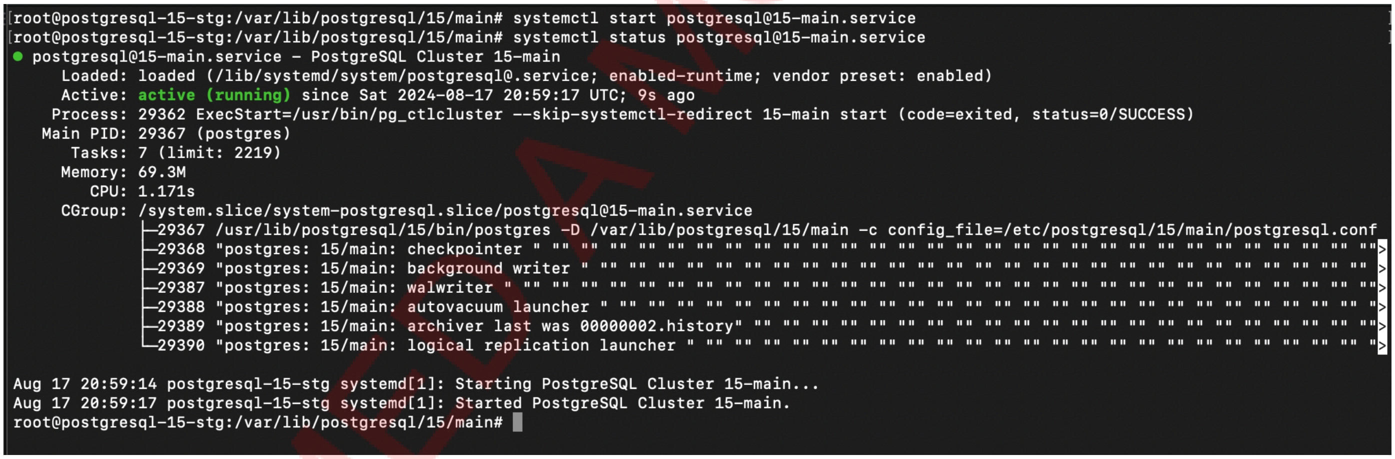
```
sudo -u postgres pgbackrest --stanza=main --log-level-console=info restore
```



```
root@postgresql-15-stg:/var/lib/postgresql/15/main# sudo -u postgres pgbackrest --stanza=main --log-level-console=info restore
2024-08-17 20:58:18.600 P00   INFO: restore command begin 2.53: --exec-id=29340-b8ed66ee --log-level-console=info --pg1-path=/var/lib/postgresq
l/15/main --repo1-path=/db_backup/base_backup --stanza=main
2024-08-17 20:58:18.633 P00   INFO: repo1: restore backup set 20240817-204536F, recovery will start at 2024-08-17 20:45:36
2024-08-17 20:58:32.328 P00   INFO: write updated /var/lib/postgresql/15/main/postgresql.auto.conf
2024-08-17 20:58:32.334 P00   INFO: restore global/pg_control (performed last to ensure aborted restores cannot be started)
2024-08-17 20:58:32.336 P00   INFO: restore size = 1GB, file total = 1633
2024-08-17 20:58:32.336 P00   INFO: restore command end: completed successfully (13743ms)
root@postgresql-15-stg:/var/lib/postgresql/15/main# 
```

start the services for postgresql

```
systenctl start postgresql
```



```
[root@postgresql-15-stg:/var/lib/postgresql/15/main# systemctl start postgresql@15-main.service
[root@postgresql-15-stg:/var/lib/postgresql/15/main# systemctl status postgresql@15-main.service
● postgresql@15-main.service - PostgreSQL Cluster 15-main
     Loaded: loaded (/lib/systemd/system/postgresql@.service; enabled-runtime; vendor preset: enabled)
     Active: active (running) since Sat 2024-08-17 20:59:17 UTC; 9s ago
    Process: 29362 ExecStart=/usr/bin/pg_ctlcluster --skip-systemctl-redirect 15-main start (code=exited, status=0/SUCCESS)
   Main PID: 29367 (postgres)
      Tasks: 7 (limit: 2219)
     Memory: 69.3M
        CPU: 1.171s
     CGroup: /system.slice/system-postgresql.slice/postgresql@15-main.service
             ├─29367 /usr/lib/postgresql/15/bin/postgres -D /var/lib/postgresql/15/main -c config_file=/etc/postgresql/15/main/postgresql.conf
             ├─29368 "postgres: 15/main: checkpointer " "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ">
             ├─29369 "postgres: 15/main: background writer " "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" >
             ├─29387 "postgres: 15/main: walwriter " "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ">
             ├─29388 "postgres: 15/main: autovacuum launcher " "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ">
             ├─29389 "postgres: 15/main: archiver last was 00000002.history" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ">
             └─29390 "postgres: 15/main: logical replication launcher " "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ">

Aug 17 20:59:14 postgresql-15-stg systemd[1]: Starting PostgreSQL Cluster 15-main...
Aug 17 20:59:17 postgresql-15-stg systemd[1]: Started PostgreSQL Cluster 15-main.
root@postgresql-15-stg:/var/lib/postgresql/15/main# 
```

For more information on how to use pgBackRest and explore additional options such as setting up a backup server, please visit the following URL:

https://www.enterprisedb.com/docs/supported-open-source/pgbackrest/