

Continuous Archiving and Point-in-Time Recovery (PITR):

- PostgreSQL maintains WAL files for all transactions in pg_wal directory.
- PostgreSQL automatically maintains the WAL logs which are full and switched.
- Continuous archiving can be setup to keep a copy of switched WAL Logs which can be later used for recovery.
- It also enables online file system backup of a database cluster.
- To enable WAL archiving, set the wal_level configuration parameter to replica or higher, archive_mode to on, and specify the shell command to use in the archive_command configuration parameter.

Setting Up WAL Archiving:

Step1) Create archive logs directory and assign ownership.

```
mkdir -p /mnt/server/archivedir/
```

```
chown postgres:postgres /mnt/server/archivedir/
```

```
[root@ip-172-31-24-211 ec2-user]# mkdir -p /mnt/server/archivedir/
[root@ip-172-31-24-211 ec2-user]# chown postgres:postgres /mnt/server/archivedir/
```

Step2) To enable WAL archiving, edit \$PGDATA/postgresql.conf file

```
vim $PGDATA/postgresql.conf
```

```
wal_level = replica
```

```
archive_mode = on
```

```
archive_command = 'cp %p /mnt/server/archivedir/%f'
```

```
archive_timeout = 120 #number of sec(2min)
```

```
save&exit
```

Step3) Restart Postgresql service and check changes

```
sudo systemctl restart postgresql-version
```

```
postgres=# show wal_level;
wal_level
-----
 replica
(1 row)

postgres=# show archive_mode;
archive_mode
-----
 on
(1 row)

postgres=# show archive_command;
archive_command
-----
 cp %p /mnt/server/archivedir/%f
(1 row)

postgres=# show archive_timeout;
archive_timeout
-----
 2min
(1 row)
```

Making a Base Backup using pg_basebackup tool.

pg_basebackup — take a base backup of a PostgreSQL cluster.

options:

-D directory

Sets the target directory to write the output to. pg_basebackup will create this directory (and any missing parent directories) if it does not exist. If it already exists, it must be empty.

-T olddir=newdir

--tablespace-mapping=olddir=newdir

Relocates the tablespace in directory olddir to newdir during the backup.pg_basebackup will create this directory (and any missing parent directories) if it does not exist. If it already exists, it must be empty.

-v verbose

pg_basebackup -D data_directory_bkp -T olddir1=newdir1 -T olddir2=newdir2 -v -p 5432

```
[postgres@ip-172-31-24-211 ~]$ pg_basebackup -U postgres -D /var/lib/pgsql/base_bkp/data_bkp -T /var/lib/pgsql/tbl_space1=/var/lib/pgsql/base_bkp/tbl_space1_bkp -T /var/lib/pgsql/tbl_space2=/var/lib/pgsql/base_bkp/tbl_space2_bkp -v -p 5432
pg_basebackup: initiating base backup, waiting for checkpoint to complete
pg_basebackup: checkpoint completed
pg_basebackup: write-ahead log start point: 0/11000028 on timeline 1
pg_basebackup: starting background WAL receiver
pg_basebackup: created temporary replication slot "pg_basebackup_3720"
pg_basebackup: write-ahead log end point: 0/11000100
pg_basebackup: waiting for background process to finish streaming ...
pg_basebackup: syncing data to disk ...
pg_basebackup: renaming backup_manifest.tmp to backup_manifest
pg_basebackup: base backup completed
```

Get current timestamp and drop tables to perform recovery:

DROP TABLE table_name CASCADE;

```
[postgres@ip-172-31-24-211 ~]$ psql dvdrental
psql (16.2)
Type "help" for help.

dvdrental=# select current_timestamp;
           current_timestamp
-----
2024-04-17 03:33:24.659928+00
(1 row)

dvdrental=# \dt+
               List of relations
Schema | Name          | Type  | Owner  | Persistence | Access method | Size  | Description
-----|-----|-----|-----|-----|-----|-----|-----
public | actor         | table | postgres | permanent   | heap          | 40 kB |
public | address       | table | postgres | permanent   | heap          | 88 kB |
public | category      | table | postgres | permanent   | heap          | 8192 bytes |
public | city          | table | postgres | permanent   | heap          | 64 kB |
public | country       | table | postgres | permanent   | heap          | 8192 bytes |
public | customer      | table | postgres | permanent   | heap          | 96 kB |
public | film          | table | postgres | permanent   | heap          | 736 kB |
public | film_actor    | table | postgres | permanent   | heap          | 272 kB |
public | film_category | table | postgres | permanent   | heap          | 72 kB |
public | inventory     | table | postgres | permanent   | heap          | 232 kB |
public | language      | table | postgres | permanent   | heap          | 8192 bytes |
public | payment       | table | postgres | permanent   | heap          | 896 kB |
public | rental        | table | postgres | permanent   | heap          | 1232 kB |
public | staff         | table | postgres | permanent   | heap          | 16 kB |
public | store         | table | postgres | permanent   | heap          | 8192 bytes |
(15 rows)

dvdrental=# DROP TABLE actor CASCADE;
NOTICE: drop cascades to 4 other objects
DETAIL: drop cascades to view actor_info
drop cascades to view film_list
drop cascades to view nicer_but_slower_film_list
drop cascades to constraint film_actor_actor_id_fkey on table film_actor
DROP TABLE
dvdrental=#
```

Run `pg_switch_wal()` function to allow current WAL file to be archived:

```
select pg_switch_wal();
```

```
ERROR: must be superuser
dvdrental=# select pg_switch_wal();
pg_switch_wal
-----
0/14000078
(1 row)

dvdrental=# \q
[postgres@ip-172-31-24-211 ~]$ ls -ltrh /mnt/server/archivedir/
total 321M
-rw-r-----. 1 postgres postgres 16M Apr 17 02:45 00000001000000000000000000000001
-rw-r-----. 1 postgres postgres 16M Apr 17 02:51 00000001000000000000000000000002
-rw-r-----. 1 postgres postgres 16M Apr 17 02:53 00000001000000000000000000000003
-rw-r-----. 1 postgres postgres 16M Apr 17 03:07 00000001000000000000000000000004
-rw-r-----. 1 postgres postgres 16M Apr 17 03:09 00000001000000000000000000000005
-rw-r-----. 1 postgres postgres 16M Apr 17 03:13 00000001000000000000000000000006
-rw-r-----. 1 postgres postgres 16M Apr 17 03:14 00000001000000000000000000000007
-rw-r-----. 1 postgres postgres 16M Apr 17 03:14 00000001000000000000000000000008
-rw-r-----. 1 postgres postgres 338 Apr 17 03:14 00000001000000000000000000000008.00000028.backup
-rw-r-----. 1 postgres postgres 16M Apr 17 03:16 00000001000000000000000000000009
-rw-r-----. 1 postgres postgres 16M Apr 17 03:16 0000000100000000000000000000000A
-rw-r-----. 1 postgres postgres 16M Apr 17 03:18 0000000100000000000000000000000B
-rw-r-----. 1 postgres postgres 16M Apr 17 03:26 0000000100000000000000000000000C
-rw-r-----. 1 postgres postgres 16M Apr 17 03:27 0000000100000000000000000000000D
-rw-r-----. 1 postgres postgres 16M Apr 17 03:28 0000000100000000000000000000000E
-rw-r-----. 1 postgres postgres 338 Apr 17 03:28 0000000100000000000000000000000E.00000110.backup
-rw-r-----. 1 postgres postgres 16M Apr 17 03:30 0000000100000000000000000000000F
-rw-r-----. 1 postgres postgres 16M Apr 17 03:31 00000001000000000000000000000010
-rw-r-----. 1 postgres postgres 16M Apr 17 03:31 00000001000000000000000000000011
-rw-r-----. 1 postgres postgres 341 Apr 17 03:31 00000001000000000000000000000011.00000028.backup
-rw-r-----. 1 postgres postgres 16M Apr 17 03:33 00000001000000000000000000000012
-rw-r-----. 1 postgres postgres 16M Apr 17 03:35 00000001000000000000000000000013
-rw-r-----. 1 postgres postgres 16M Apr 17 03:38 00000001000000000000000000000014
[postgres@ip-172-31-24-211 ~]$ date
Wed Apr 17 03:38:56 AM UTC 2024
[postgres@ip-172-31-24-211 ~]$
```

PITR Using timestamp:

Step1) Edit `postgresql.conf` file in a base backup data directory.

```
vim data_directory_bkp/postgresql.conf
```

```
port=6432
```

```
restore_command = 'cp /mnt/server/archivedir/%f %p'
```

```
recovery_target_time = '2024-04-17 03:33:24' #(select current_timestamp; query output)
```

Comment below parameters.

```
#wal_level = replica
```

```
#archive_mode = on
```

```
#archive_command = 'cp %p /mnt/server/archivedir/%f'
```

```
#archive_timeout = 120
```

Save&exit

Step2) create a file `recovery.signal` in the base backup data directory

```
touch data_directory_bkp/recovery.signal
```

```
[postgres@ip-172-31-24-211 data_bkp]$ pwd
/var/lib/pgsql/base_bkp/data_bkp
[postgres@ip-172-31-24-211 data_bkp]$ touch recovery.signal
[postgres@ip-172-31-24-211 data_bkp]$ ls -ltrh recovery.signal
-rw-r--r--. 1 postgres postgres 0 Apr 17 03:48 recovery.signal
[postgres@ip-172-31-24-211 data_bkp]$
```

Step3) Start the base backup cluster using pg_ctl tool.

pg_ctl -D data_directory_bkp -l logfile start

```
[postgres@ip-172-31-24-211 data_bkp]$ /usr/pgsql-16/bin/pg_ctl -D /var/lib/pgsql/base_bkp/data_bkp -l logfile start
waiting for server to start.... done
server started
```

Step4) Check the logs.

less data_directory_bkp /log/postgresql.log

```
2024-04-17 03:51:50.494 UTC [3848] LOG: starting PostgreSQL 16.2 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 11.4.1 20230605 (Red Hat 11.4.1-2), 64-bit
2024-04-17 03:51:50.495 UTC [3848] LOG: listening on IPv4 address "0.0.0.0", port 6432
2024-04-17 03:51:50.495 UTC [3848] LOG: listening on IPv6 address ":::", port 6432
2024-04-17 03:51:50.497 UTC [3848] LOG: listening on Unix socket "/run/postgresql/.s.PGSQL.6432"
2024-04-17 03:51:50.500 UTC [3848] LOG: listening on Unix socket "/tmp/.s.PGSQL.6432"
2024-04-17 03:51:50.504 UTC [3852] LOG: database system was interrupted; last known up at 2024-04-17 03:31:10 UTC
cp: cannot stat '/mnt/server/archivedir/00000002.history': No such file or directory
2024-04-17 03:51:50.520 UTC [3852] LOG: starting point-in-time recovery to 2024-04-17 03:33:24+00
2024-04-17 03:51:50.520 UTC [3852] LOG: starting backup recovery with redo LSN 0/11000028, checkpoint LSN 0/11000060, on timeline ID 1
2024-04-17 03:51:50.523 UTC [3852] LOG: restored log file "000000010000000000000011" from archive
2024-04-17 03:51:50.529 UTC [3852] LOG: redo starts at 0/11000028
2024-04-17 03:51:50.532 UTC [3852] LOG: restored log file "000000010000000000000012" from archive
2024-04-17 03:51:50.539 UTC [3852] LOG: restored log file "000000010000000000000013" from archive
2024-04-17 03:51:50.552 UTC [3852] LOG: completed backup recovery with redo LSN 0/11000028 and end LSN 0/11000100
2024-04-17 03:51:50.552 UTC [3852] LOG: consistent recovery state reached at 0/11000100
2024-04-17 03:51:50.552 UTC [3848] LOG: database system is ready to accept read-only connections
2024-04-17 03:51:50.555 UTC [3852] LOG: restored log file "000000010000000000000014" from archive
2024-04-17 03:51:50.561 UTC [3852] LOG: restored log file "000000010000000000000015" from archive
cp: cannot stat '/mnt/server/archivedir/000000010000000000000016': No such file or directory
2024-04-17 03:51:50.567 UTC [3852] LOG: recovery stopping before commit of transaction 1348, time 2024-04-17 03:34:00.868255+00
2024-04-17 03:51:50.567 UTC [3852] LOG: pausing at the end of recovery
2024-04-17 03:51:50.567 UTC [3852] HINT: Execute pg_wal_replay_resume() to promote.
```

Step5) Run pg_wal_replay_resume() to complete recovery and check logs.

select pg_wal_replay_resume();

```
[postgres@ip-172-31-24-211 log]$ psql dvdrental -p 6432
psql (16.2)
Type "help" for help.

dvdrental=# select pg_wal_replay_resume();
 pg_wal_replay_resume
-----
(1 row)
```

```
2024-04-17 04:01:53.259 UTC [3852] LOG: archive recovery complete
2024-04-17 04:01:53.261 UTC [3850] LOG: checkpoint starting: end-of-recovery immediate wait
2024-04-17 04:01:53.276 UTC [3850] LOG: checkpoint complete: wrote 4 buffers (0.0%); 0 WAL file(s) added, 0 removed, 1 recycled; write=0.001 s, sync=0.003 s, total=0.017 s; sync files=3, longest=0.002 s, average=0.001 s; distance=16572 kB, estimate=16572 kB; lsn=0/13031360, redo lsn=0/13031360
2024-04-17 04:01:53.281 UTC [3848] LOG: database system is ready to accept connections
```

Step6) Connect the database with port 6432 and check the data has been recovered or not.

```
[postgres@ip-172-31-24-211 log]$ psql dvdrental -p 6432
psql (16.2)
Type "help" for help.

dvdrental=# \dt+
              List of relations
Schema | Name          | Type  | Owner  | Persistence | Access method | Size  | Description
-----|-----|-----|-----|-----|-----|-----|-----
public | actor         | table | postgres | permanent   | heap          | 40 kB |
public | address       | table | postgres | permanent   | heap          | 88 kB |
public | category      | table | postgres | permanent   | heap          | 8192 bytes |
public | city          | table | postgres | permanent   | heap          | 64 kB |
public | country       | table | postgres | permanent   | heap          | 8192 bytes |
public | customer      | table | postgres | permanent   | heap          | 96 kB |
public | film          | table | postgres | permanent   | heap          | 736 kB |
public | film_actor    | table | postgres | permanent   | heap          | 272 kB |
public | film_category | table | postgres | permanent   | heap          | 72 kB |
public | inventory     | table | postgres | permanent   | heap          | 232 kB |
public | language      | table | postgres | permanent   | heap          | 8192 bytes |
public | payment       | table | postgres | permanent   | heap          | 896 kB |
public | rental        | table | postgres | permanent   | heap          | 1232 kB |
public | staff         | table | postgres | permanent   | heap          | 16 kB |
public | store         | table | postgres | permanent   | heap          | 8192 bytes |
(15 rows)

dvdrental=# select * from actor limit 2;
 actor_id | first_name | last_name | last_update
-----+-----+-----+-----
      1 | Penelope  | Guinness | 2013-05-26 14:47:57.62
      2 | Nick      | Wahlberg | 2013-05-26 14:47:57.62
(2 rows)
```