Upgrade Streaming Replication Setup using pg_upgrade

OS:Redhat Linux

OS Version:9

Postgresql old version:9.5

Postgresql new version:16

Old Cluster: /app/data

New Cluster: /var/lib/pgsql/16/data

Primary server:172.31.95.130

Standby server:172.31.42.140

Step1) Collect database tables, tablespace and users, etc.. list

```
postgres=# \dt
          List of relations
 Schema |  Name   | Type  |  Owner
--------+---------+-------+----------
 public | dept    | table | postgres
 public | emp     | table | postgres
 public | mytable | table | postgres
(3 rows)

postgres=# \du
                             List of roles
 Role name |                         Attributes
-----------+------------------------------------------------------------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS
 repuser   | Replication

postgres=# \db+
                                       List of tablespaces
    Name    |  Owner   |    Location     | Access privileges | Options |  Size  | Description
------------+----------+-----------------+-------------------+---------+--------+-------------
 pg_default | postgres |                 |                   |         | 22 MB  |
 pg_global  | postgres |                 |                   |         | 565 kB |
 tbl_space1 | postgres | /app/tbl_space1 |                   |         | 16 kB  |
 tbl_space2 | postgres | /app/tbl_space2 |                   |         | 16 kB  |
(4 rows)
```

Step2) move the old cluster.

mv /usr/local/pgsql /usr/local/pgsql_old

Step3) Install the new PostgreSQL

sudo dnf install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-9-x86_64/pgdg-redhat-repo-latest.noarch.rpm

sudo dnf -qy module disable postgresql

sudo dnf install -y postgresql16-server

Step4) Initialize the new PostgreSQL cluster

sudo /usr/pgsql-16/bin/postgresql-16-setup initdb

Step5) Install contrib extension.

#If you are using extensions install postgresql16-contrib for extension updates

sudo dnf install -y postgresql16-contrib

Step6) Stop both servers

pg_ctl -D /var/lib/pgsql/16/data stop --->new cluster

pg_ctl -D /app/data stop--->old cluster

Note: Streaming replication and log-shipping standby servers must be running during this shutdown, so they receive all changes.

Step7)Run pg_upgrade

#Always run the pg_upgrade binary of the new server, not the old one. pg_upgrade requires the specification of the old and new cluster's data and executable (bin) directories.

/usr/pgsql-16/bin/pg_upgrade --old-bindir "old-bindir_path" --new-bindir "new-bindir_path" --old-datadir "old-datadir_path" --new-datadir "new-datadir_path"

#perform only the checks(option -c)

/usr/pgsql-16/bin/pg_upgrade -b /usr/local/pgsql_old/bin -B /usr/pgsql-16/bin -d /app/data -D /var/lib/pgsql/16/data -c

```
[postgres@ip-172-31-95-130 ~]$ /usr/pgsql-16/bin/pg_upgrade -b /usr/local/pgsql_old/bin -B /usr/pgsql-16/bin -d /app/data -D /var/lib/pgsql/16/data -c
Performing Consistency Checks
-----------------------------
Checking cluster versions                                   ok
Checking database user is the install user                  ok
Checking database connection settings                       ok
Checking for prepared transactions                          ok
Checking for system-defined composite types in user tables  ok
Checking for reg* data types in user tables                 ok
Checking for contrib/isn with bigint-passing mismatch       ok
Checking for incompatible "aclitem" data type in user tables  ok
Checking for removed "abstime" data type in user tables     ok
Checking for removed "reltime" data type in user tables     ok
Checking for removed "tinterval" data type in user tables   ok
Checking for user-defined encoding conversions              ok
Checking for user-defined postfix operators                 ok
Checking for incompatible polymorphic functions             ok
Checking for tables WITH OIDS                               ok
Checking for invalid "sql_identifier" user columns          ok
Checking for invalid "unknown" user columns                 ok
Checking for hash indexes                                   ok
Checking for roles starting with "pg_"                      ok
Checking for presence of required libraries                 ok
Checking database user is the install user                  ok
Checking for prepared transactions                          ok
Checking for new cluster tablespace directories             ok

*Clusters are compatible*
```

#use hard links(option -k) instead of copying files to the new cluster

/usr/pgsql-16/bin/pg_upgrade -b /usr/local/pgsql_old/bin -B /usr/pgsql-16/bin -d /app/data -D /var/lib/pgsql/16/data -k

```
[postgres@ip-172-31-95-130 ~]$ /usr/pgsql-16/bin/pg_upgrade -b /usr/local/pgsql_old/bin -B /usr/pgsql-16/bin -d /app/data -D /var/lib/pgsql/16/data -k
Performing Consistency Checks
-----------------------------
Checking cluster versions                                   ok
Checking database user is the install user                  ok
Checking database connection settings                       ok
Checking for prepared transactions                          ok
Checking for system-defined composite types in user tables  ok
Checking for reg* data types in user tables                 ok
Checking for contrib/isn with bigint-passing mismatch       ok
Checking for incompatible "aclitem" data type in user tables  ok
Checking for removed "abstime" data type in user tables     ok
Checking for removed "reltime" data type in user tables     ok
Checking for removed "tinterval" data type in user tables   ok
Checking for user-defined encoding conversions              ok
Checking for user-defined postfix operators                 ok
Checking for incompatible polymorphic functions             ok
Checking for tables WITH OIDS                               ok
Checking for invalid "sql_identifier" user columns          ok
Checking for invalid "unknown" user columns                 ok
Checking for roles starting with "pg_"                      ok
Creating dump of global objects                             ok
Creating dump of database schemas
                                                            ok
Checking for presence of required libraries                 ok
Checking database user is the install user                  ok
Checking for prepared transactions                          ok
Checking for new cluster tablespace directories             ok

If pg_upgrade fails after this point, you must re-initdb the
new cluster before continuing.

Performing Upgrade
------------------
Setting locale and encoding for new cluster                 ok
Analyzing all rows in the new cluster                       ok
Freezing all rows in the new cluster                        ok
Deleting files from new pg_xact                             ok
Copying old pg_clog to new server                           ok
Setting oldest XID for new cluster                          ok
Setting next transaction ID and epoch for new cluster       ok
Deleting files from new pg_multixact/offsets                ok
Copying old pg_multixact/offsets to new server              ok
```

```
Performing Upgrade
------------------
Setting locale and encoding for new cluster                   ok
Analyzing all rows in the new cluster                         ok
Freezing all rows in the new cluster                          ok
Deleting files from new pg_xact                               ok
Copying old pg_clog to new server                             ok
Setting oldest XID for new cluster                            ok
Setting next transaction ID and epoch for new cluster         ok
Deleting files from new pg_multixact/offsets                  ok
Copying old pg_multixact/offsets to new server                ok
Deleting files from new pg_multixact/members                  ok
Copying old pg_multixact/members to new server                ok
Setting next multixact ID and offset for new cluster          ok
Resetting WAL archives                                        ok
Setting frozenxid and minmxid counters in new cluster         ok
Restoring global objects in the new cluster                   ok
Restoring database schemas in the new cluster
                                                              ok
Adding ".old" suffix to old global/pg_control                 ok

If you want to start the old cluster, you will need to remove
the ".old" suffix from /app/data/global/pg_control.old.
Because "link" mode was used, the old cluster cannot be safely
started once the new cluster has been started.
Linking user relation files
                                                              ok
Setting next OID for new cluster                              ok
Sync data directory to disk                                   ok
Creating script to delete old cluster                         ok
Checking for hash indexes                                     ok
Checking for extension updates                                ok

Upgrade Complete
----------------
Optimizer statistics are not transferred by pg_upgrade.
Once you start the new server, consider running:
    /usr/pgsql-16/bin/vacuumdb --all --analyze-in-stages
Running this script will delete the old cluster's data files:
    ./delete_old_cluster.sh
```

Step8) Upgrade streaming replication and log-shipping standby servers

#Install the new PostgreSQL binaries on standby servers

sudo dnf install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-9-x86_64/pgdg-redhat-repo-latest.noarch.rpm

sudo dnf -qy module disable postgresql

sudo dnf install -y postgresql16-server

#Make sure the new standby data directories do not exist

Make sure the new standby data directories do not exist or are empty. If initdb was run, delete the standby servers' new data directories.

rm -rf recursive /var/lib/pgsql/16/data/*

#Install contrib extension

sudo dnf install -y postgresql16-contrib

#Stop standby servers

pg_ctl -D /app/data stop--->old cluster

#Run rsync

When using link mode, standby server can be quickly upgraded using rsync ,run this on the primary for standby server:

rsync --archive --delete --hard-links --size-only --no-inc-recursive /app/data /var/lib/pgsql/16/data

rsync --archive --delete --hard-links --size-only --no-inc-recursive /var/lib/pgsql/16/data postgres@172.31.42.140:/var/lib/pgsql/16

#If you have tablespaces, you will need to run a similar rsync command for each tablespace directory

rsync --archive --delete --hard-links --size-only --no-inc-recursive /app/tbl_space1 postgres@172.31.42.140:/app

rsync --archive --delete --hard-links --size-only --no-inc-recursive /app/tbl_space2 postgres@172.31.42.140:/app

Step9) Configure streaming replication and log-shipping standby server

#Create a file standby.signal in the standby's cluster data directory

cd /var/lib/pgsql/16/data

touch standby.signal

#Edit postgresql.conf

vim /var/lib/pgsql/16/data/postgresql.conf

listen_addresses = '*'

port = 5432

primary_conninfo = 'host=172.31.95.130 port=5432 user=repuser password=reppass@123 options=''-c wal_sender_timeout=5000'''

hot_standby = on

restore_command = 'rsync -a postgres@172.31.95.130:/mnt/server/archivedir/%f %p'

save&exit

Step10) Start the new server.

#Start the new upgraded primary server

sudo systemctl enable postgresql-16

sudo systemctl start postgresql-16

#Check the status

sudo systemctl status postgresql-16

Step11) Configure primary server for streaming replication and log-shipping

#Edit postgresql.conf file

vim /var/lib/pgsql/16/data/postgresql.conf

listen_addresses = '*'

port = 5432

wal_level = replica

archive_mode = on

archive_command = 'rsync -a %p /mnt/server/archivedir/%f && rsync -a %p postgres@172.31.42.140:/mnt/server/archivedir/%f'

save&exit

#Edit pg_hba.conf file

vim /var/lib/pgsql/16/data/pg_hba.conf

 # replication privilege.

host replication repuser 172.31.42.140/32 md5

save&exit

#Restart postgresql server

sudo systemctl restart postgresql-16

Step12) Start rsync'ed standby server.

#Start the postgresql

sudo systemctl start postgresql-16

#Check the status

sudo systemctl status postgresql-16

Step13) Verify the replication status.

#on primary database instance

select * from pg_stat_replication;

```
postgres=# select * from pg_stat_replication;
-[ RECORD 1 ]----+------------------------------
pid              | 7255
usesysid         | 16401
usename          | repuser
application_name | walreceiver
client_addr      | 172.31.42.140
client_hostname  |
client_port      | 50124
backend_start    | 2024-04-06 04:56:12.655389+00
backend_xmin     |
state            | streaming
sent_lsn         | 0/F0314B0
write_lsn        | 0/F0314B0
flush_lsn        | 0/F0314B0
replay_lsn       | 0/F0314B0
write_lag        |
flush_lag        |
replay_lag       |
sync_priority    | 0
sync_state       | async
reply_time       | 2024-04-06 06:06:41.452469+00

postgres=#
```

#on Standby database instance

select * from pg_stat_wal_receiver ;

```
postgres=# select * from pg_stat_wal_receiver ;
-[ RECORD 1 ]---------+------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------
pid                   | 28381
status                | streaming
receive_start_lsn     | 0/F000000
receive_start_tli     | 1
written_lsn           | 0/F0314B0
flushed_lsn           | 0/F0314B0
received_tli          | 1
last_msg_send_time    | 2024-04-06 06:04:46.274195+00
last_msg_receipt_time | 2024-04-06 06:04:46.274899+00
latest_end_lsn        | 0/F0314B0
latest_end_time       | 2024-04-06 05:04:10.517546+00
slot_name             |
sender_host           | 172.31.95.130
sender_port           | 5432
conninfo              | user=repuser password=******** channel_binding=prefer dbname=replication host=172.31.95.130 port=5432 options=-c wal_sender_timeout=5
000 fallback_application_name=walreceiver sslmode=prefer sslcompression=0 sslcertmode=allow sslsni=1 ssl_min_protocol_version=TLSv1.2 gssencmode=prefer krbsr
vname=postgres gssdelegation=0 target_session_attrs=any load_balance_hosts=disable
```

Step14) Delete old cluster on both primary and standby.

rm -rf '/app/data'

rm -rf '/app/tbl_space1/PG_9.5_201510051'

rm -rf '/app/tbl_space2/PG_9.5_201510051'

Step15) Check database tables, tablespace, and users, etc. list

```
postgres=# \dt
          List of relations
 Schema |  Name    | Type  |  Owner
--------+---------+-------+----------
 public | dept     | table | postgres
 public | emp      | table | postgres
 public | mytable  | table | postgres
(3 rows)

postgres=# \du
                             List of roles
 Role name |                         Attributes
-----------+------------------------------------------------------------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS
 repuser   | Replication

postgres=# \db+
                                     List of tablespaces
    Name    |  Owner   |    Location     | Access privileges | Options |  Size   | Description
------------+----------+-----------------+-------------------+---------+---------+-------------
 pg_default | postgres |                 |                   |         | 22 MB   |
 pg_global  | postgres |                 |                   |         | 565 kB  |
 tbl_space1 | postgres | /app/tbl_space1 |                   |         | 16 kB   |
 tbl_space2 | postgres | /app/tbl_space2 |                   |         | 16 kB   |
(4 rows)
```