Upgrade PostgreSQL with pglogical:

OLD Server OS: RHEL-9.3
NEW Server OS: Ubuntu 24.04 LTS

OLD Server PostgreSQL version: postgresql-9.15

NEW Server PostgreSQL version: postgresql-15.6

OLD Server IP: 172.31.23.62

NEW Server IP: 172.31.27.122

PGLogical Setup on OLD server:

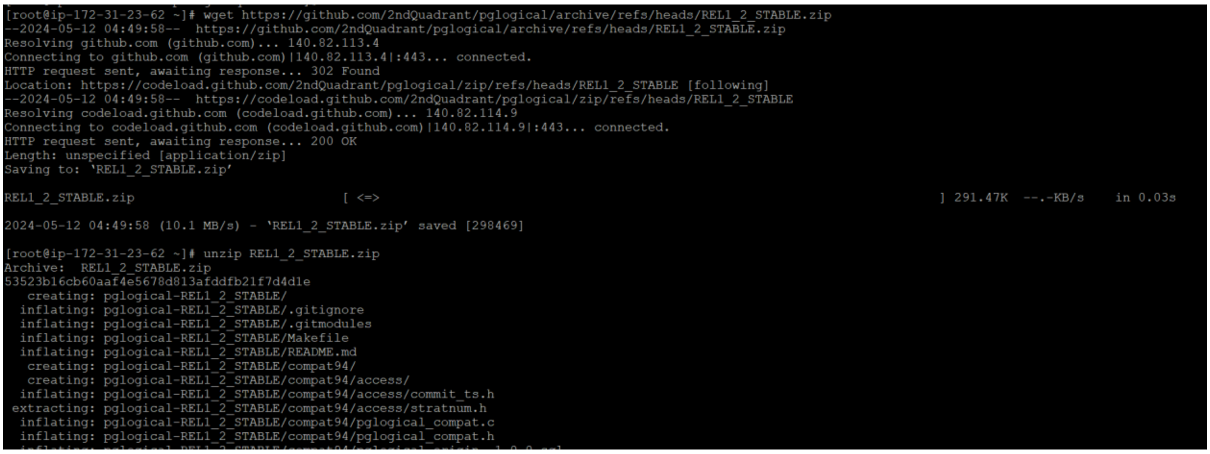
Step1) pglogical installation

#Download pglogical source code

https://github.com/2ndQuadrant/pglogical/archive/refs/heads/REL1_2_STABLE.zip

#Unzip pglogical zip file

unzip REL1_2_STABLE.zip



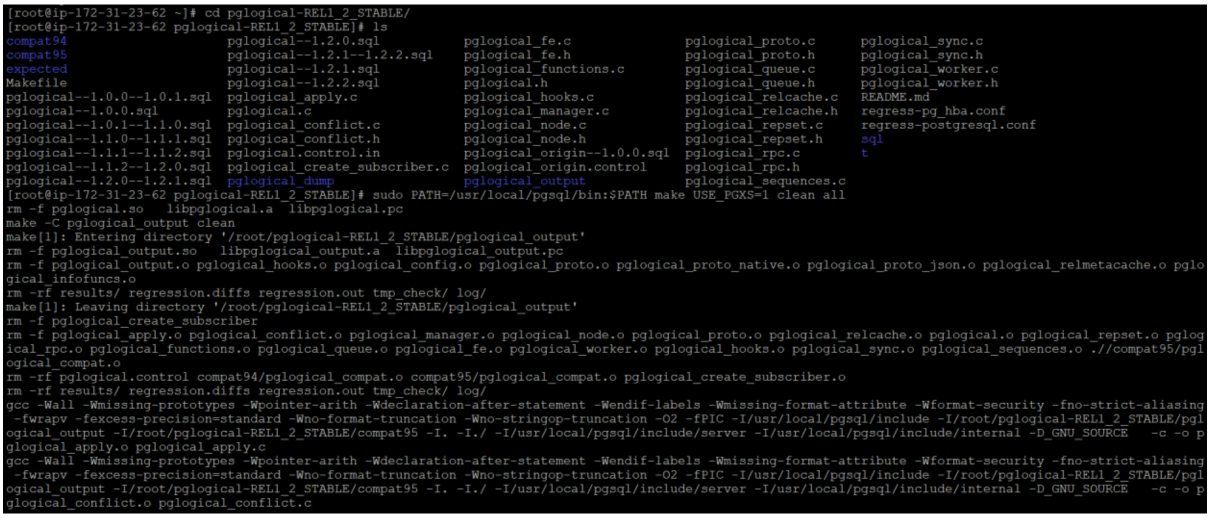
```
[root@ip-172-31-23-62 ~]# wget https://github.com/2ndQuadrant/pglogical/archive/refs/heads/REL1_2_STABLE.zip
--2024-05-12 04:49:58--  https://github.com/2ndQuadrant/pglogical/archive/refs/heads/REL1_2_STABLE.zip
Resolving github.com (github.com)... 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/2ndQuadrant/pglogical/zip/refs/heads/REL1_2_STABLE [following]
--2024-05-12 04:49:58--  https://codeload.github.com/2ndQuadrant/pglogical/zip/refs/heads/REL1_2_STABLE
Resolving codeload.github.com (codeload.github.com)... 140.82.114.9
Connecting to codeload.github.com (codeload.github.com)|140.82.114.9|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: ‘REL1_2_STABLE.zip’

REL1_2_STABLE.zip                      [ <=>                                                                              ] 291.47K  --.-KB/s    in 0.03s

2024-05-12 04:49:58 (10.1 MB/s) - ‘REL1_2_STABLE.zip’ saved [298469]

[root@ip-172-31-23-62 ~]# unzip REL1_2_STABLE.zip
Archive:  REL1_2_STABLE.zip
53523b16cb60aaf4e5678d813afddfb21f7d4d1e
   creating: pglogical-REL1_2_STABLE/
  inflating: pglogical-REL1_2_STABLE/.gitignore
  inflating: pglogical-REL1_2_STABLE/.gitmodules
  inflating: pglogical-REL1_2_STABLE/Makefile
  inflating: pglogical-REL1_2_STABLE/README.md
   creating: pglogical-REL1_2_STABLE/compat94/
   creating: pglogical-REL1_2_STABLE/compat94/access/
  inflating: pglogical-REL1_2_STABLE/compat94/access/commit_ts.h
 extracting: pglogical-REL1_2_STABLE/compat94/access/stratnum.h
  inflating: pglogical-REL1_2_STABLE/compat94/pglogical_compat.c
  inflating: pglogical-REL1_2_STABLE/compat94/pglogical_compat.h
```

#Install pglogical

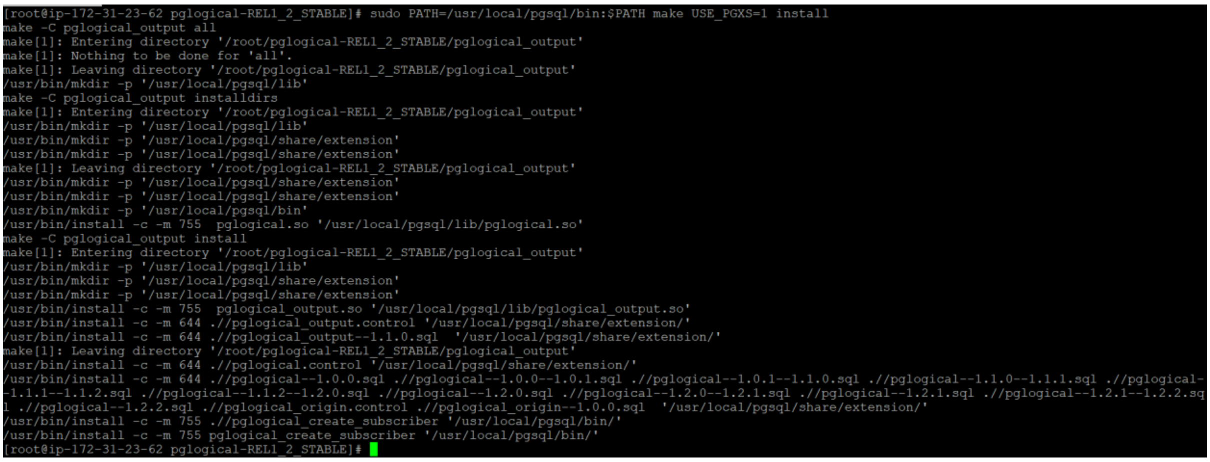
cd pglogical-REL1_2_STABLE

sudo PATH=/usr/local/pgsql/bin:$PATH make USE_PGXS=1 clean all

sudo PATH=/usr/local/pgsql/bin:$PATH make USE_PGXS=1 install



```
[root@ip-172-31-23-62 ~]# cd pglogical-REL1_2_STABLE/
[root@ip-172-31-23-62 pglogical-REL1_2_STABLE]# ls
compat94                     pglogical--1.2.0.sql           pglogical_fe.c               pglogical_proto.c      pglogical_sync.c
compat95                     pglogical--1.2.1--1.2.2.sql    pglogical_fe.h               pglogical_proto.h      pglogical_sync.h
expected                     pglogical--1.2.1.sql           pglogical_functions.c        pglogical_queue.c      pglogical_worker.c
Makefile                     pglogical--1.2.2.sql           pglogical.h                  pglogical_queue.h      pglogical_worker.h
pglogical--1.0.0--1.0.1.sql  pglogical_apply.c              pglogical_hooks.c            pglogical_relcache.c   README.md
pglogical--1.0.0.sql         pglogical.c                    pglogical_manager.c          pglogical_relcache.h   regress-pg_hba.conf
pglogical--1.0.1--1.1.0.sql  pglogical_conflict.c           pglogical_node.c             pglogical_repset.c     regress-postgresql.conf
pglogical--1.1.0--1.1.1.sql  pglogical_conflict.h           pglogical_node.h             pglogical_repset.h     sql
pglogical--1.1.1--1.1.2.sql  pglogical.control.in           pglogical_origin--1.0.0.sql  pglogical_rpc.c        t
pglogical--1.1.2--1.2.0.sql  pglogical_create_subscriber.c  pglogical_origin.control     pglogical_rpc.h
pglogical--1.2.0--1.2.1.sql  pglogical_dump                 pglogical_output             pglogical_sequences.c
[root@ip-172-31-23-62 pglogical-REL1_2_STABLE]# sudo PATH=/usr/local/pgsql/bin:$PATH make USE_PGXS=1 clean all
rm -f pglogical.so   libpglogical.a  libpglogical.pc
make -C pglogical_output clean
make[1]: Entering directory '/root/pglogical-REL1_2_STABLE/pglogical_output'
rm -f pglogical_output.so   libpglogical_output.a  libpglogical_output.pc
rm -f pglogical_output.o pglogical_hooks.o pglogical_config.o pglogical_proto.o pglogical_proto_native.o pglogical_proto_json.o pglogical_relmetacache.o pglo
gical_infofuncs.o
rm -rf results/ regression.diffs regression.out tmp_check/ log/
make[1]: Leaving directory '/root/pglogical-REL1_2_STABLE/pglogical_output'
rm -f pglogical_create_subscriber
rm -f pglogical_apply.o pglogical_conflict.o pglogical_manager.o pglogical_node.o pglogical_proto.o pglogical_relcache.o pglogical.o pglogical_repset.o pglog
ical_rpc.o pglogical_functions.o pglogical_queue.o pglogical_fe.o pglogical_worker.o pglogical_hooks.o pglogical_sync.o pglogical_sequences.o .//compat95/pgl
ogical_compat.o
rm -rf pglogical.control compat94/pglogical_compat.o compat95/pglogical_compat.o pglogical_create_subscriber.o
rm -rf results/ regression.diffs regression.out tmp_check/ log/
gcc -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement -Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-aliasing
 -fwrapv -fexcess-precision=standard -Wno-format-truncation -Wno-stringop-truncation -O2 -fPIC -I/usr/local/pgsql/include -I/root/pglogical-REL1_2_STABLE/pgl
ogical_output -I/root/pglogical-REL1_2_STABLE/compat95 -I. -I./ -I/usr/local/pgsql/include/server -I/usr/local/pgsql/include/internal -D_GNU_SOURCE   -c -o p
glogical_apply.o pglogical_apply.c
gcc -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement -Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-aliasing
 -fwrapv -fexcess-precision=standard -Wno-format-truncation -Wno-stringop-truncation -O2 -fPIC -I/usr/local/pgsql/include -I/root/pglogical-REL1_2_STABLE/pgl
ogical_output -I/root/pglogical-REL1_2_STABLE/compat95 -I. -I./ -I/usr/local/pgsql/include/server -I/usr/local/pgsql/include/internal -D_GNU_SOURCE   -c -o p
glogical_conflict.o pglogical_conflict.c
```



```
[root@ip-172-31-23-62 pglogical-REL1_2_STABLE]# sudo PATH=/usr/local/pgsql/bin:$PATH make USE_PGXS=1 install
make -C pglogical_output all
make[1]: Entering directory '/root/pglogical-REL1_2_STABLE/pglogical_output'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory '/root/pglogical-REL1_2_STABLE/pglogical_output'
/usr/bin/mkdir -p '/usr/local/pgsql/lib'
make -C pglogical_output installdirs
make[1]: Entering directory '/root/pglogical-REL1_2_STABLE/pglogical_output'
/usr/bin/mkdir -p '/usr/local/pgsql/lib'
/usr/bin/mkdir -p '/usr/local/pgsql/share/extension'
/usr/bin/mkdir -p '/usr/local/pgsql/share/extension'
make[1]: Leaving directory '/root/pglogical-REL1_2_STABLE/pglogical_output'
/usr/bin/mkdir -p '/usr/local/pgsql/share/extension'
/usr/bin/mkdir -p '/usr/local/pgsql/share/extension'
/usr/bin/mkdir -p '/usr/local/pgsql/bin'
/usr/bin/install -c -m 755  pglogical.so '/usr/local/pgsql/lib/pglogical.so'
make -C pglogical_output install
make[1]: Entering directory '/root/pglogical-REL1_2_STABLE/pglogical_output'
/usr/bin/mkdir -p '/usr/local/pgsql/lib'
/usr/bin/mkdir -p '/usr/local/pgsql/share/extension'
/usr/bin/mkdir -p '/usr/local/pgsql/share/extension'
/usr/bin/install -c -m 755  pglogical_output.so '/usr/local/pgsql/lib/pglogical_output.so'
/usr/bin/install -c -m 644 .//pglogical_output.control '/usr/local/pgsql/share/extension/'
/usr/bin/install -c -m 644 .//pglogical_output--1.1.0.sql  '/usr/local/pgsql/share/extension/'
make[1]: Leaving directory '/root/pglogical-REL1_2_STABLE/pglogical_output'
/usr/bin/install -c -m 644 .//pglogical.control '/usr/local/pgsql/share/extension/'
/usr/bin/install -c -m 644 .//pglogical--1.0.0.sql .//pglogical--1.0.0--1.0.1.sql .//pglogical--1.0.1--1.1.0.sql .//pglogical--1.1.0--1.1.1.sql .//pglogical-
-1.1.1--1.1.2.sql .//pglogical--1.1.2--1.2.0.sql .//pglogical--1.2.0.sql .//pglogical--1.2.0--1.2.1.sql .//pglogical--1.2.1.sql .//pglogical--1.2.1--1.2.2.sq
l .//pglogical--1.2.2.sql .//pglogical_origin.control .//pglogical_origin--1.0.0.sql  '/usr/local/pgsql/share/extension/'
/usr/bin/install -c -m 755 .//pglogical_create_subscriber '/usr/local/pgsql/bin/'
/usr/bin/install -c -m 755 pglogical_create_subscriber '/usr/local/pgsql/bin/'
[root@ip-172-31-23-62 pglogical-REL1_2_STABLE]#
```

#Configure $PGDATA/postgresql.conf file

vim $PGDATA/postgresql.conf

wal_level = 'logical'

max_worker_processes = 10

max_replication_slots = 10

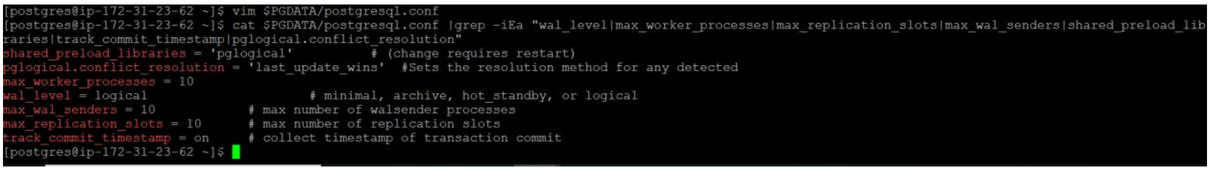
max_wal_senders = 10

shared_preload_libraries = 'pglogical'

track_commit_timestamp = on

pglogical.conflict_resolution = 'last_update_wins'

save&exit

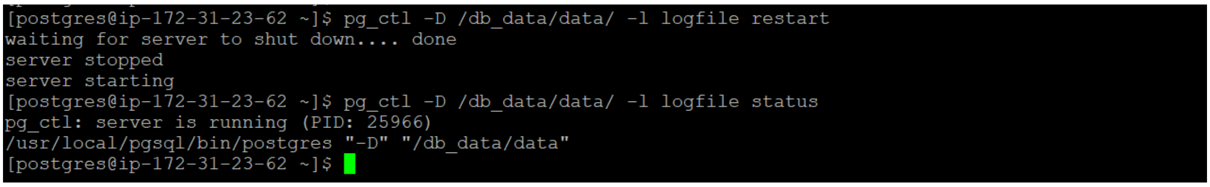


```
[postgres@ip-172-31-23-62 ~]$ vim $PGDATA/postgresql.conf
[postgres@ip-172-31-23-62 ~]$ cat $PGDATA/postgresql.conf |grep -iEa "wal_level|max_worker_processes|max_replication_slots|max_wal_senders|shared_preload_lib
raries|track_commit_timestamp|pglogical.conflict_resolution"
shared_preload_libraries = 'pglogical'          # (change requires restart)
pglogical.conflict_resolution = 'last_update_wins'  #Sets the resolution method for any detected
max_worker_processes = 10
wal_level = logical                     # minimal, archive, hot_standby, or logical
max_wal_senders = 10            # max number of walsender processes
max_replication_slots = 10      # max number of replication slots
track_commit_timestamp = on     # collect timestamp of transaction commit
[postgres@ip-172-31-23-62 ~]$
```

#Restart postgresql service and check the status

pg_ctl -D /db_data/data/ -l logfile restart

pg_ctl -D /db_data/data/ -l logfile status



```
[postgres@ip-172-31-23-62 ~]$ pg_ctl -D /db_data/data/ -l logfile restart
waiting for server to shut down.... done
server stopped
server starting
[postgres@ip-172-31-23-62 ~]$ pg_ctl -D /db_data/data/ -l logfile status
pg_ctl: server is running (PID: 25966)
/usr/local/pgsql/bin/postgres "-D" "/db_data/data"
[postgres@ip-172-31-23-62 ~]$
```

PGLogical setup on NEW server:

Step1) Install PostgreSQL on new server

# Import the repository signing key:

sudo apt install curl ca-certificates

sudo install -d /usr/share/postgresql-common/pgdg

sudo curl -o /usr/share/postgresql-common/pgdg/apt.postgresql.org.asc --fail https://www.postgresql.org/media/keys/ACCC4CF8.asc

# Create the repository configuration file:

sudo sh -c 'echo "deb [signed-by=/usr/share/postgresql-common/pgdg/apt.postgresql.org.asc] https://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'

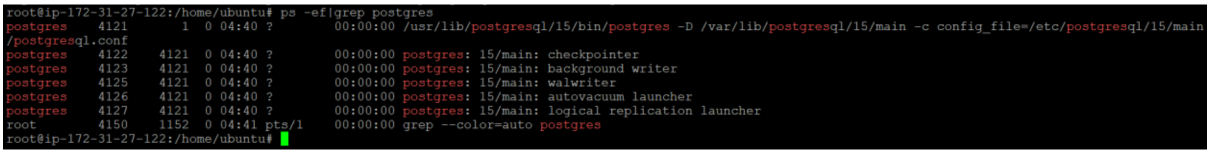
# Update the package lists:

sudo apt update

# Install the latest version of PostgreSQL:

# If you want a specific version, use 'postgresql-16' or similar instead of 'postgresql'

sudo apt -y install postgresql-15

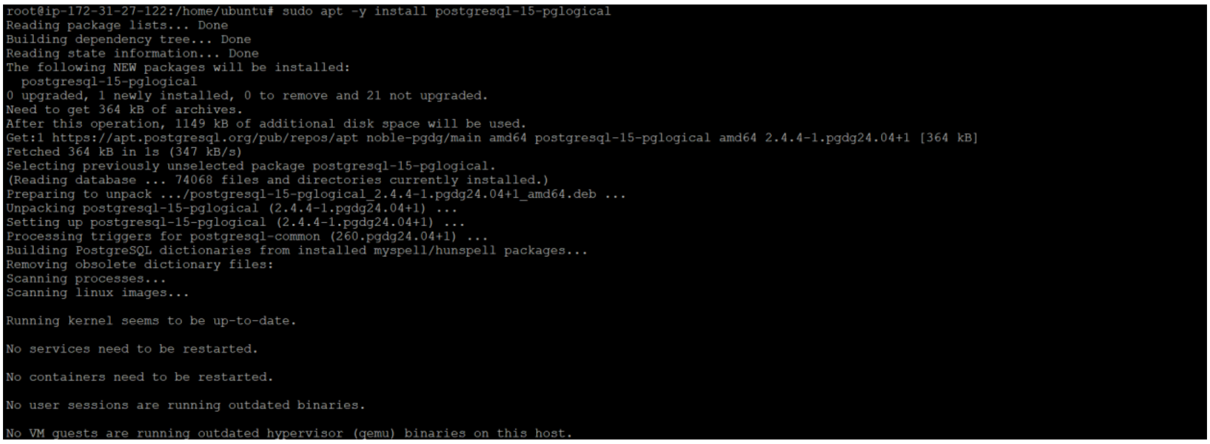


```
root@ip-172-31-27-122:/home/ubuntu# ps -ef|grep postgres
postgres    4121       1  0 04:40 ?        00:00:00 /usr/lib/postgresql/15/bin/postgres -D /var/lib/postgresql/15/main -c config_file=/etc/postgresql/15/main
/postgresql.conf
postgres    4122    4121  0 04:40 ?        00:00:00 postgres: 15/main: checkpointer
postgres    4123    4121  0 04:40 ?        00:00:00 postgres: 15/main: background writer
postgres    4125    4121  0 04:40 ?        00:00:00 postgres: 15/main: walwriter
postgres    4126    4121  0 04:40 ?        00:00:00 postgres: 15/main: autovacuum launcher
postgres    4127    4121  0 04:40 ?        00:00:00 postgres: 15/main: logical replication launcher
root        4150    1152  0 04:41 pts/1    00:00:00 grep --color=auto postgres
root@ip-172-31-27-122:/home/ubuntu#
```

Step2) Installation of pglogical

#installation

sudo apt-get install postgresql-15-pglogical



```
root@ip-172-31-27-122:/home/ubuntu# sudo apt -y install postgresql-15-pglogical
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  postgresql-15-pglogical
0 upgraded, 1 newly installed, 0 to remove and 21 not upgraded.
Need to get 364 kB of archives.
After this operation, 1149 kB of additional disk space will be used.
Get:1 https://apt.postgresql.org/pub/repos/apt noble-pgdg/main amd64 postgresql-15-pglogical amd64 2.4.4-1.pgdg24.04+1 [364 kB]
Fetched 364 kB in 1s (347 kB/s)
Selecting previously unselected package postgresql-15-pglogical.
(Reading database ... 74068 files and directories currently installed.)
Preparing to unpack .../postgresql-15-pglogical_2.4.4-1.pgdg24.04+1_amd64.deb ...
Unpacking postgresql-15-pglogical (2.4.4-1.pgdg24.04+1) ...
Setting up postgresql-15-pglogical (2.4.4-1.pgdg24.04+1) ...
Processing triggers for postgresql-common (260.pgdg24.04+1) ...
Building PostgreSQL dictionaries from installed myspell/hunspell packages...
Removing obsolete dictionary files:
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

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.
```

#Quick setup

Add below parameters in a $PGDATA/postgresq.conf file

listen_address='*'

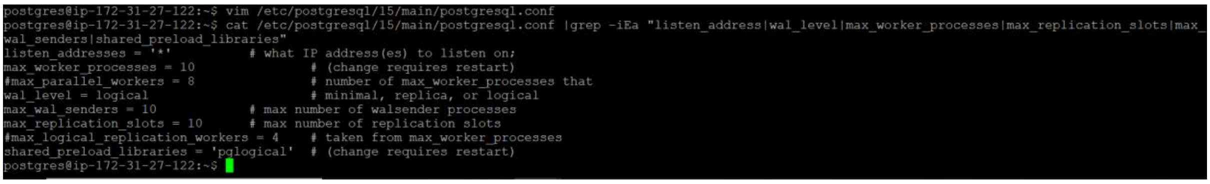
wal_level = 'logical'

max_worker_processes = 10

max_replication_slots = 10

max_wal_senders = 10

shared_preload_libraries = 'pglogical'

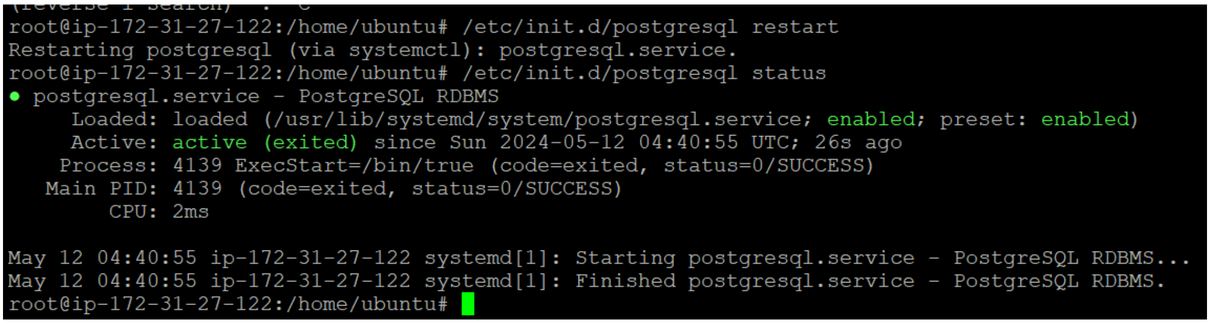


```
postgres@ip-172-31-27-122:~$ vim /etc/postgresql/15/main/postgresql.conf
postgres@ip-172-31-27-122:~$ cat /etc/postgresql/15/main/postgresql.conf |grep -iEa "listen_address|wal_level|max_worker_processes|max_replication_slots|max_
wal_senders|shared_preload_libraries"
listen_addresses = '*'          # what IP address(es) to listen on;
max_worker_processes = 10               # (change requires restart)
#max_parallel_workers = 8               # number of max_worker_processes that
wal_level = logical                     # minimal, replica, or logical
max_wal_senders = 10            # max number of walsender processes
max_replication_slots = 10      # max number of replication slots
#max_logical_replication_workers = 4    # taken from max_worker_processes
shared_preload_libraries = 'pglogical'  # (change requires restart)
postgres@ip-172-31-27-122:~$
```

#Restart the service and check the status

/etc/init.d/postgresql restart

/etc/init.d/postgresql status



```
(reverse-i-search)`': c
root@ip-172-31-27-122:/home/ubuntu# /etc/init.d/postgresql restart
Restarting postgresql (via systemctl): postgresql.service.
root@ip-172-31-27-122:/home/ubuntu# /etc/init.d/postgresql status
● postgresql.service - PostgreSQL RDBMS
     Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; preset: enabled)
     Active: active (exited) since Sun 2024-05-12 04:40:55 UTC; 26s ago
    Process: 4139 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 4139 (code=exited, status=0/SUCCESS)
        CPU: 2ms

May 12 04:40:55 ip-172-31-27-122 systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...
May 12 04:40:55 ip-172-31-27-122 systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.
root@ip-172-31-27-122:/home/ubuntu#
```

Prepare For Logical Replication:

Some key limitations and restrictions:

Superuser is required:

Currently pglogical replication and administration requires superuser privileges. It may be later extended to more granular privileges.

One database at a time:

To replicate multiple databases you must set up individual provider/subscriber relationships for each. There is no way to configure replication for all databases in a PostgreSQL install at once.

PRIMARY KEY or REPLICA IDENTITY required:

UPDATEs and DELETEs cannot be replicated for tables that lack a PRIMARY KEY or other valid replica identity such as using an index, which must be unique, not partial, not deferrable, and include only columns marked NOT NULL. Replication has no way to find the tuple that should be updated/deleted since there is no unique identifier. REPLICA IDENTITY FULL is not supported yet.

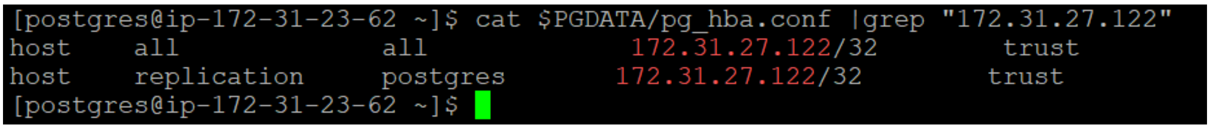
DDL:

Automatic DDL replication is not supported. Managing DDL so that the provider and subscriber database(s) remain compatible is the responsibility of the user.

Step1) Enable remote connection for new server

host all all 172.31.27.122/32 trust

host replication postgres 172.31.27.122/32 trust

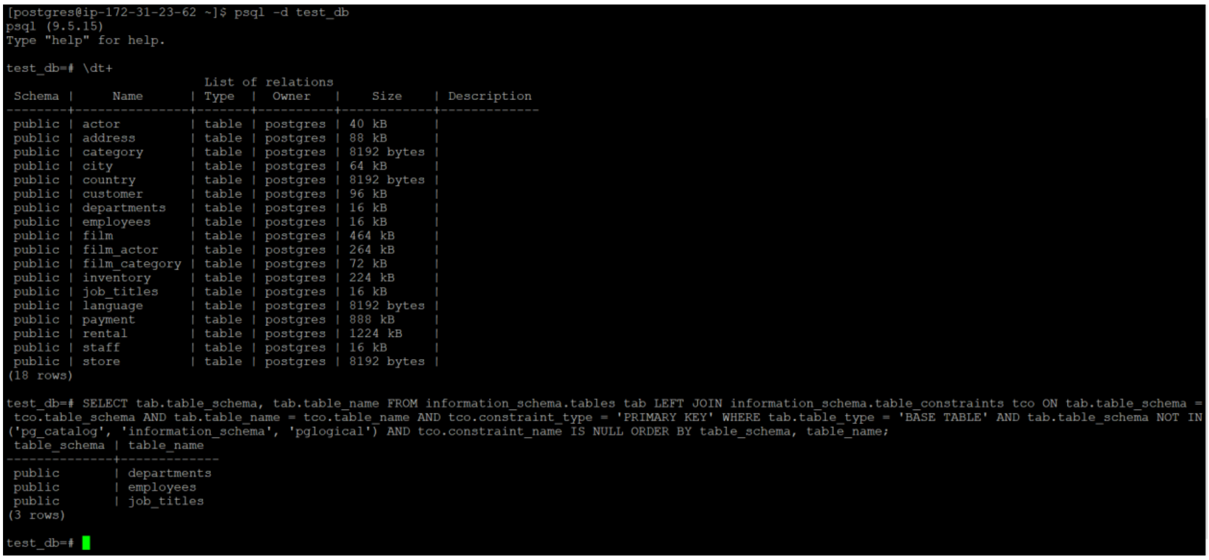


```
[postgres@ip-172-31-23-62 ~]$ cat $PGDATA/pg_hba.conf |grep "172.31.27.122"
host    all             all             172.31.27.122/32        trust
host    replication     postgres        172.31.27.122/32        trust
[postgres@ip-172-31-23-62 ~]$
```

Step2) Check for tables without a primary key and set replica identities.

#Find tables without a primary key query

SELECT tab.table_schema, tab.table_name FROM information_schema.tables tab LEFT JOIN information_schema.table_constraints tco ON tab.table_schema = tco.table_schema AND tab.table_name = tco.table_name AND tco.constraint_type = 'PRIMARY KEY' WHERE tab.table_type = 'BASE TABLE' AND tab.table_schema NOT IN('pg_catalog', 'information_schema', 'pglogical') AND tco.constraint_name IS NULL ORDER BY table_schema, table_name;



```
[postgres@ip-172-31-23-62 ~]$ psql -d test_db
psql (9.5.15)
Type "help" for help.

test_db=# \dt+
                          List of relations
 Schema |     Name      | Type  |  Owner   |    Size    | Description
--------+---------------+-------+----------+------------+-------------
 public | actor         | table | postgres | 40 kB      |
 public | address       | table | postgres | 88 kB      |
 public | category      | table | postgres | 8192 bytes |
 public | city          | table | postgres | 64 kB      |
 public | country       | table | postgres | 8192 bytes |
 public | customer      | table | postgres | 96 kB      |
 public | departments   | table | postgres | 16 kB      |
 public | employees     | table | postgres | 16 kB      |
 public | film          | table | postgres | 464 kB     |
 public | film_actor    | table | postgres | 264 kB     |
 public | film_category | table | postgres | 72 kB      |
 public | inventory     | table | postgres | 224 kB     |
 public | job_titles    | table | postgres | 16 kB      |
 public | language      | table | postgres | 8192 bytes |
 public | payment       | table | postgres | 888 kB     |
 public | rental        | table | postgres | 1224 kB    |
 public | staff         | table | postgres | 16 kB      |
 public | store         | table | postgres | 8192 bytes |
(18 rows)

test_db=# SELECT tab.table_schema, tab.table_name FROM information_schema.tables tab LEFT JOIN information_schema.table_constraints tco ON tab.table_schema =
 tco.table_schema AND tab.table_name = tco.table_name AND tco.constraint_type = 'PRIMARY KEY' WHERE tab.table_type = 'BASE TABLE' AND tab.table_schema NOT IN
('pg_catalog', 'information_schema', 'pglogical') AND tco.constraint_name IS NULL ORDER BY table_schema, table_name;
 table_schema | table_name
--------------+-------------
 public       | departments
 public       | employees
 public       | job_titles
(3 rows)

test_db=#
```

#Create unique index on without a primary key table

ALTER TABLE departments ALTER COLUMN department_id SET NOT NULL;

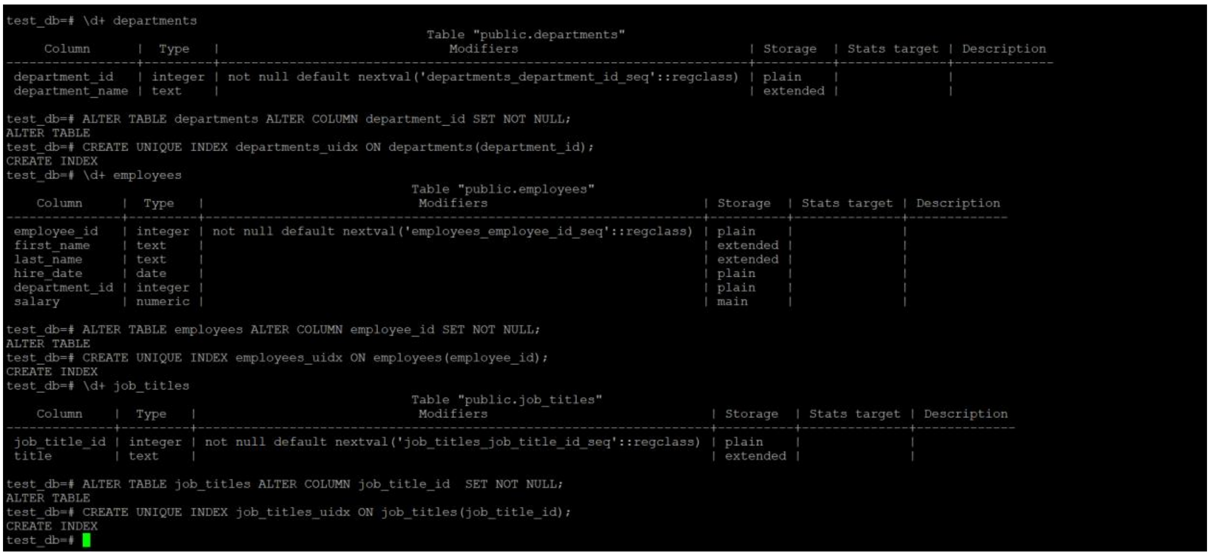
CREATE UNIQUE INDEX departments_uidx ON departments(department_id);

ALTER TABLE employees ALTER COLUMN employee_id SET NOT NULL;

CREATE UNIQUE INDEX employees_uidx ON employees(employee_id);

ALTER TABLE job_titles ALTER COLUMN job_title_id SET NOT NULL;

CREATE UNIQUE INDEX job_titles_uidx ON job_titles(job_title_id);



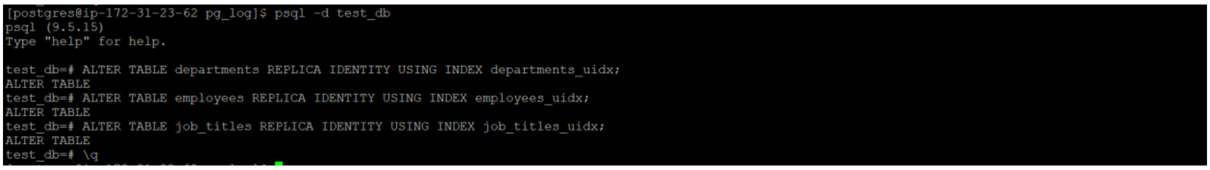
```
test_db=# \d+ departments
                                                    Table "public.departments"
     Column      |  Type   |                                Modifiers                                | Storage  | Stats target | Description
-----------------+---------+-------------------------------------------------------------------------+----------+--------------+-------------
 department_id   | integer | not null default nextval('departments_department_id_seq'::regclass) | plain    |              |
 department_name | text    |                                                                         | extended |              |

test_db=# ALTER TABLE departments ALTER COLUMN department_id SET NOT NULL;
ALTER TABLE
test_db=# CREATE UNIQUE INDEX departments_uidx ON departments(department_id);
CREATE INDEX
test_db=# \d+ employees
                                                  Table "public.employees"
    Column     |  Type   |                              Modifiers                              | Storage  | Stats target | Description
---------------+---------+---------------------------------------------------------------------+----------+--------------+-------------
 employee_id   | integer | not null default nextval('employees_employee_id_seq'::regclass) | plain    |              |
 first_name    | text    |                                                                     | extended |              |
 last_name     | text    |                                                                     | extended |              |
 hire_date     | date    |                                                                     | plain    |              |
 department_id | integer |                                                                     | plain    |              |
 salary        | numeric |                                                                     | main     |              |

test_db=# ALTER TABLE employees ALTER COLUMN employee_id SET NOT NULL;
ALTER TABLE
test_db=# CREATE UNIQUE INDEX employees_uidx ON employees(employee_id);
CREATE INDEX
test_db=# \d+ job_titles
                                                  Table "public.job_titles"
    Column    |  Type   |                               Modifiers                               | Storage  | Stats target | Description
--------------+---------+-----------------------------------------------------------------------+----------+--------------+-------------
 job_title_id | integer | not null default nextval('job_titles_job_title_id_seq'::regclass) | plain    |              |
 title        | text    |                                                                       | extended |              |

test_db=# ALTER TABLE job_titles ALTER COLUMN job_title_id  SET NOT NULL;
ALTER TABLE
test_db=# CREATE UNIQUE INDEX job_titles_uidx ON job_titles(job_title_id);
CREATE INDEX
test_db=#
```

#Set replica identities using unique index

ALTER TABLE departments REPLICA IDENTITY USING INDEX departments_uidx;

ALTER TABLE employees REPLICA IDENTITY USING INDEX employees_uidx;

ALTER TABLE job_titles REPLICA IDENTITY USING INDEX job_titles_uidx;

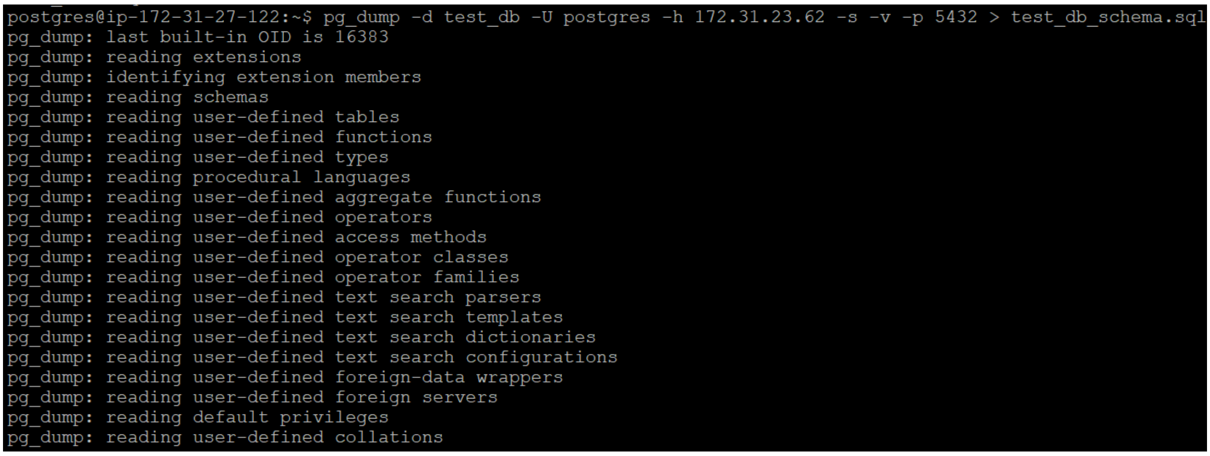


```
[postgres@ip-172-31-23-62 pg_log]$ psql -d test_db
psql (9.5.15)
Type "help" for help.

test_db=# ALTER TABLE departments REPLICA IDENTITY USING INDEX departments_uidx;
ALTER TABLE
test_db=# ALTER TABLE employees REPLICA IDENTITY USING INDEX employees_uidx;
ALTER TABLE
test_db=# ALTER TABLE job_titles REPLICA IDENTITY USING INDEX job_titles_uidx;
ALTER TABLE
test_db=# \q
```

Step3) Take Schema and roles backup from old server.

#Database schema backup

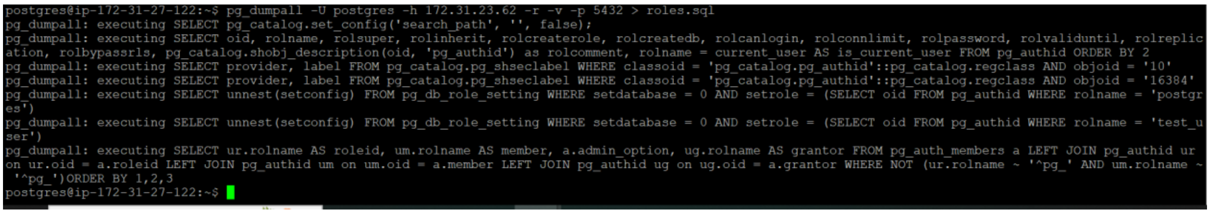
pg_dump -d test_db -U postgres -h 172.31.23.62 -s -v -p 5432 > test_db_schema.sql



```
postgres@ip-172-31-27-122:~$ pg_dump -d test_db -U postgres -h 172.31.23.62 -s -v -p 5432 > test_db_schema.sql
pg_dump: last built-in OID is 16383
pg_dump: reading extensions
pg_dump: identifying extension members
pg_dump: reading schemas
pg_dump: reading user-defined tables
pg_dump: reading user-defined functions
pg_dump: reading user-defined types
pg_dump: reading procedural languages
pg_dump: reading user-defined aggregate functions
pg_dump: reading user-defined operators
pg_dump: reading user-defined access methods
pg_dump: reading user-defined operator classes
pg_dump: reading user-defined operator families
pg_dump: reading user-defined text search parsers
pg_dump: reading user-defined text search templates
pg_dump: reading user-defined text search dictionaries
pg_dump: reading user-defined text search configurations
pg_dump: reading user-defined foreign-data wrappers
pg_dump: reading user-defined foreign servers
pg_dump: reading default privileges
pg_dump: reading user-defined collations
```

#Database cluster roles backup

pg_dumpall -U postgres -h 172.31.23.62 -r -v -p 5432 > roles.sql

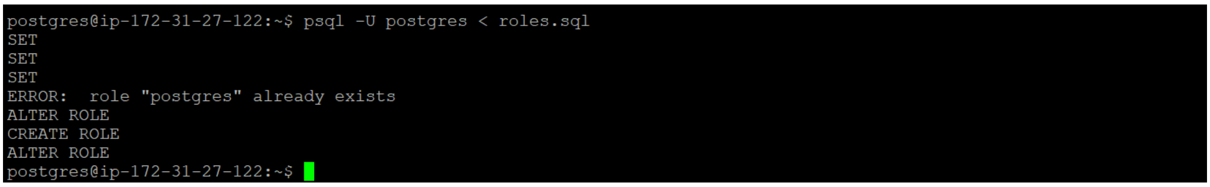


```
postgres@ip-172-31-27-122:~$ pg_dumpall -U postgres -h 172.31.23.62 -r -v -p 5432 > roles.sql
pg_dumpall: executing SELECT pg_catalog.set_config('search_path', '', false);
pg_dumpall: executing SELECT oid, rolname, rolsuper, rolinherit, rolcreaterole, rolcreatedb, rolcanlogin, rolconnlimit, rolpassword, rolvaliduntil, rolreplic
ation, rolbypassrls, pg_catalog.shobj_description(oid, 'pg_authid') as rolcomment, rolname = current_user AS is_current_user FROM pg_authid ORDER BY 2
pg_dumpall: executing SELECT provider, label FROM pg_catalog.pg_shseclabel WHERE classoid = 'pg_catalog.pg_authid'::pg_catalog.regclass AND objoid = '10'
pg_dumpall: executing SELECT provider, label FROM pg_catalog.pg_shseclabel WHERE classoid = 'pg_catalog.pg_authid'::pg_catalog.regclass AND objoid = '16384'
pg_dumpall: executing SELECT unnest(setconfig) FROM pg_db_role_setting WHERE setdatabase = 0 AND setrole = (SELECT oid FROM pg_authid WHERE rolname = 'postgr
es')
pg_dumpall: executing SELECT unnest(setconfig) FROM pg_db_role_setting WHERE setdatabase = 0 AND setrole = (SELECT oid FROM pg_authid WHERE rolname = 'test_u
ser')
pg_dumpall: executing SELECT ur.rolname AS roleid, um.rolname AS member, a.admin_option, ug.rolname AS grantor FROM pg_auth_members a LEFT JOIN pg_authid ur
on ur.oid = a.roleid LEFT JOIN pg_authid um on um.oid = a.member LEFT JOIN pg_authid ug on ug.oid = a.grantor WHERE NOT (ur.rolname ~ '^pg_' AND um.rolname ~
 '^pg_')ORDER BY 1,2,3
postgres@ip-172-31-27-122:~$
```

Step4) Restore schema and roles backup on new server.

#Restore roles

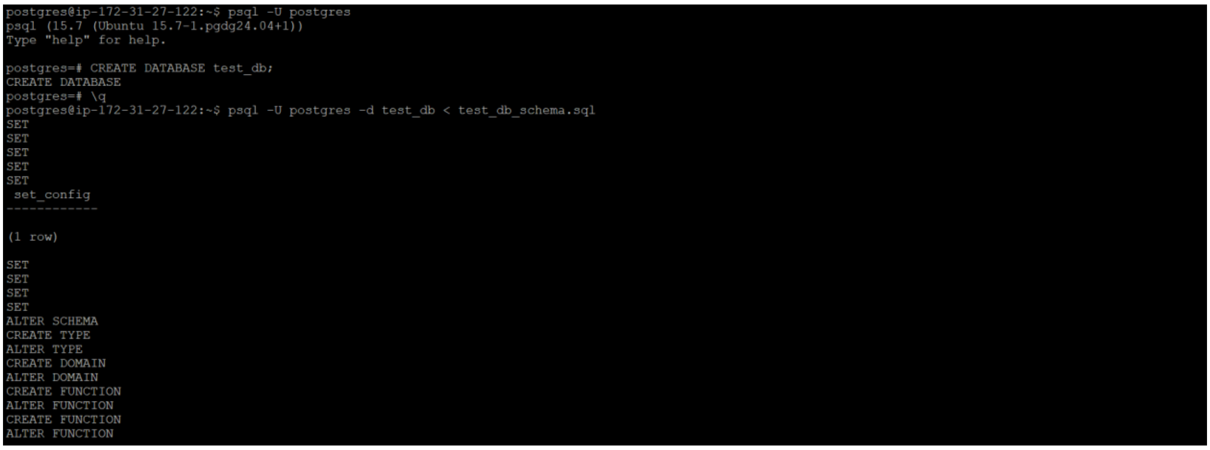
psql -U postgres < roles.sql



```
postgres@ip-172-31-27-122:~$ psql -U postgres < roles.sql
SET
SET
SET
ERROR:  role "postgres" already exists
ALTER ROLE
CREATE ROLE
ALTER ROLE
postgres@ip-172-31-27-122:~$
```

#Create new database and restore the schema backup

psql -U postgres -d test_db < test_db_schema.sql

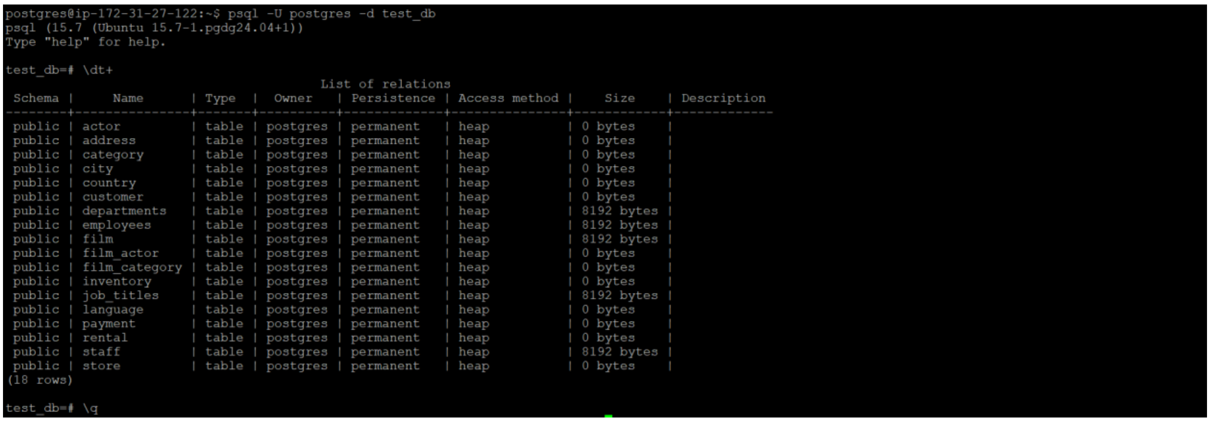


```
postgres@ip-172-31-27-122:~$ psql -U postgres
psql (15.7 (Ubuntu 15.7-1.pgdg24.04+1))
Type "help" for help.

postgres=# CREATE DATABASE test_db;
CREATE DATABASE
postgres=# \q
postgres@ip-172-31-27-122:~$ psql -U postgres -d test_db < test_db_schema.sql
SET
SET
SET
SET
SET
 set_config
------------

(1 row)

SET
SET
SET
SET
ALTER SCHEMA
CREATE TYPE
ALTER TYPE
CREATE DOMAIN
ALTER DOMAIN
CREATE FUNCTION
ALTER FUNCTION
CREATE FUNCTION
ALTER FUNCTION
```

#Verify new tables

psql -U postgres -d test_db

\dt+

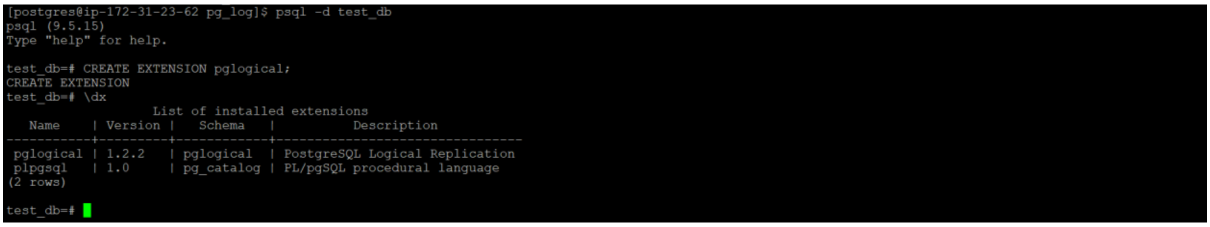


```
postgres@ip-172-31-27-122:~$ psql -U postgres -d test_db
psql (15.7 (Ubuntu 15.7-1.pgdg24.04+1))
Type "help" for help.

test_db=# \dt+
                                        List of relations
 Schema |     Name      | Type  |  Owner   | Persistence | Access method |    Size    | Description
--------+---------------+-------+----------+-------------+---------------+------------+-------------
 public | actor         | table | postgres | permanent   | heap          | 0 bytes    |
 public | address       | table | postgres | permanent   | heap          | 0 bytes    |
 public | category      | table | postgres | permanent   | heap          | 0 bytes    |
 public | city          | table | postgres | permanent   | heap          | 0 bytes    |
 public | country       | table | postgres | permanent   | heap          | 0 bytes    |
 public | customer      | table | postgres | permanent   | heap          | 0 bytes    |
 public | departments   | table | postgres | permanent   | heap          | 8192 bytes |
 public | employees     | table | postgres | permanent   | heap          | 8192 bytes |
 public | film          | table | postgres | permanent   | heap          | 8192 bytes |
 public | film_actor    | table | postgres | permanent   | heap          | 0 bytes    |
 public | film_category | table | postgres | permanent   | heap          | 0 bytes    |
 public | inventory     | table | postgres | permanent   | heap          | 0 bytes    |
 public | job_titles    | table | postgres | permanent   | heap          | 8192 bytes |
 public | language      | table | postgres | permanent   | heap          | 0 bytes    |
 public | payment       | table | postgres | permanent   | heap          | 0 bytes    |
 public | rental        | table | postgres | permanent   | heap          | 0 bytes    |
 public | staff         | table | postgres | permanent   | heap          | 8192 bytes |
 public | store         | table | postgres | permanent   | heap          | 0 bytes    |
(18 rows)

test_db=# \q
```

Step6) Create pglogical extension and setup provider node on old server.

#Create pglogical extension

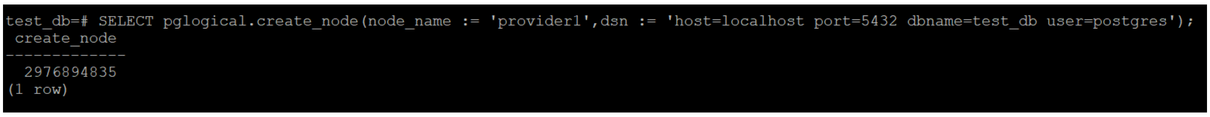
CREATE EXTENSION pglogical;



```
[postgres@ip-172-31-23-62 pg_log]$ psql -d test_db
psql (9.5.15)
Type "help" for help.

test_db=# CREATE EXTENSION pglogical;
CREATE EXTENSION
test_db=# \dx
                   List of installed extensions
   Name    | Version |   Schema   |          Description
-----------+---------+------------+------------------------------
 pglogical | 1.2.2   | pglogical  | PostgreSQL Logical Replication
 plpgsql   | 1.0     | pg_catalog | PL/pgSQL procedural language
(2 rows)

test_db=# 
```

#Create the provider node:

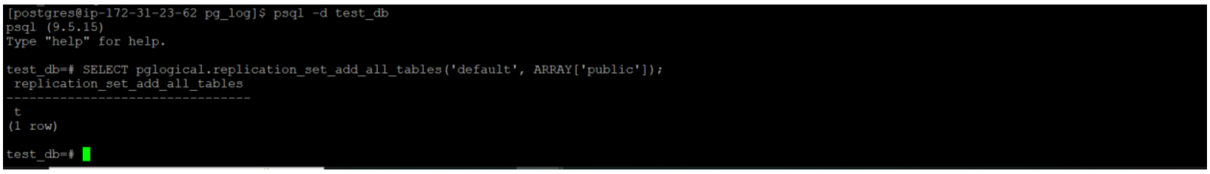
SELECT pglogical.create_node(node_name := 'provider1',dsn := 'host=localhost port=5432 dbname=test_db user=postgres');



```
test_db=# SELECT pglogical.create_node(node_name := 'provider1',dsn := 'host=localhost port=5432 dbname=test_db user=postgres');
 create_node
-------------
  2976894835
(1 row)
```

#Add all tables in public schema to the default replication set

SELECT pglogical.replication_set_add_all_tables('default', ARRAY['public']);

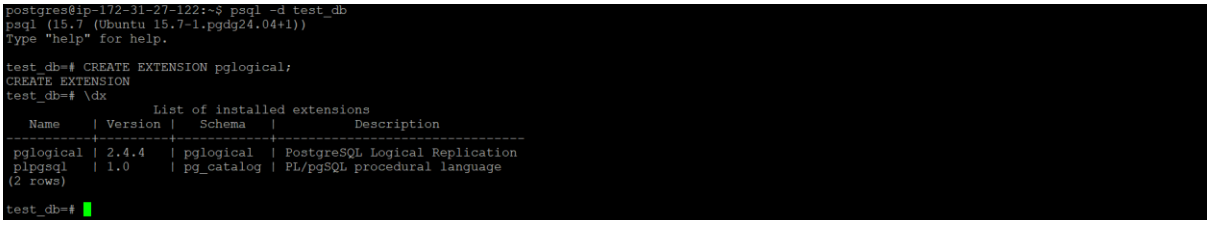


```
[postgres@ip-172-31-23-62 pg_log]$ psql -d test_db
psql (9.5.15)
Type "help" for help.

test_db=# SELECT pglogical.replication_set_add_all_tables('default', ARRAY['public']);
 replication_set_add_all_tables
--------------------------------
 t
(1 row)

test_db=#
```

Step7) Create pglogical extension and setup subscriber node on new server

#Create pglogical extension

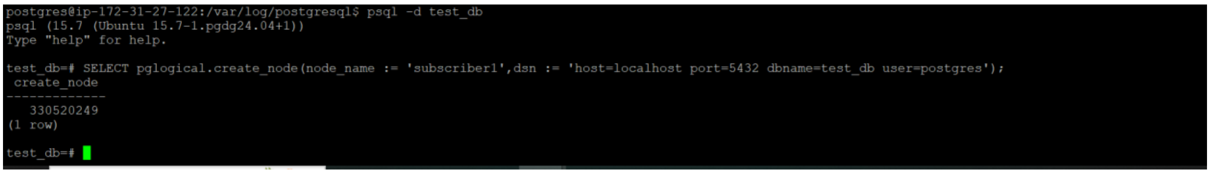
CREATE EXTENSION pglogical;



```
postgres@ip-172-31-27-122:~$ psql -d test_db
psql (15.7 (Ubuntu 15.7-1.pgdg24.04+1))
Type "help" for help.

test_db=# CREATE EXTENSION pglogical;
CREATE EXTENSION
test_db=# \dx
                  List of installed extensions
   Name    | Version |   Schema   |          Description
-----------+---------+------------+--------------------------------
 pglogical | 2.4.4   | pglogical  | PostgreSQL Logical Replication
 plpgsql   | 1.0     | pg_catalog | PL/pgSQL procedural language
(2 rows)

test_db=#
```

#Create subscriber node

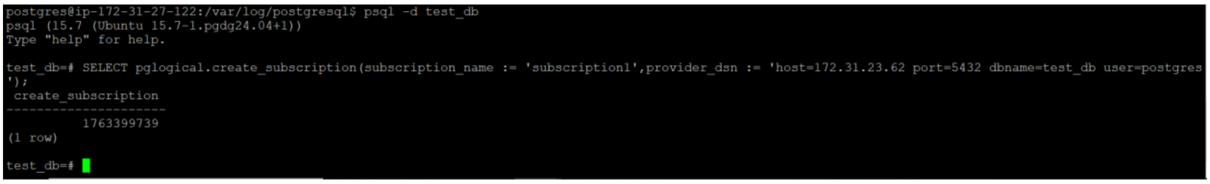
SELECT pglogical.create_node(node_name := 'subscriber1',dsn := 'host=localhost port=5432 dbname=test_db user=postgres');



```
postgres@ip-172-31-27-122:/var/log/postgresql$ psql -d test_db
psql (15.7 (Ubuntu 15.7-1.pgdg24.04+1))
Type "help" for help.

test_db=# SELECT pglogical.create_node(node_name := 'subscriber1',dsn := 'host=localhost port=5432 dbname=test_db user=postgres');
 create_node
-------------
   330520249
(1 row)

test_db=#
```

#And finally on the subscriber node you can create the subscription which will start synchronization and replication process in the background

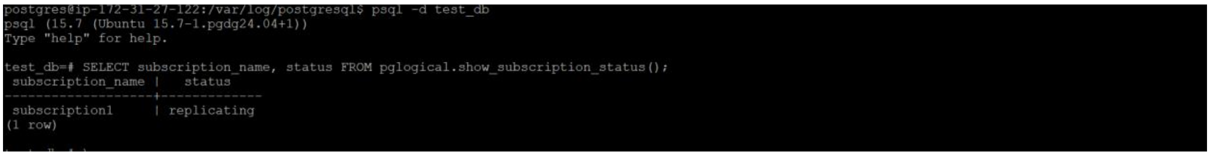
SELECT pglogical.create_subscription(subscription_name := 'subscription1',provider_dsn := 'host=172.31.23.62 port=5432 dbname=test_db user=postgres');



```
postgres@ip-172-31-27-122:/var/log/postgresql$ psql -d test_db
psql (15.7 (Ubuntu 15.7-1.pgdg24.04+1))
Type "help" for help.

test_db=# SELECT pglogical.create_subscription(subscription_name := 'subscription1',provider_dsn := 'host=172.31.23.62 port=5432 dbname=test_db user=postgres
');
 create_subscription
---------------------
          1763399739
(1 row)

test_db=#
```

#Check replication status

SELECT subscription_name, status FROM pglogical.show_subscription_status();

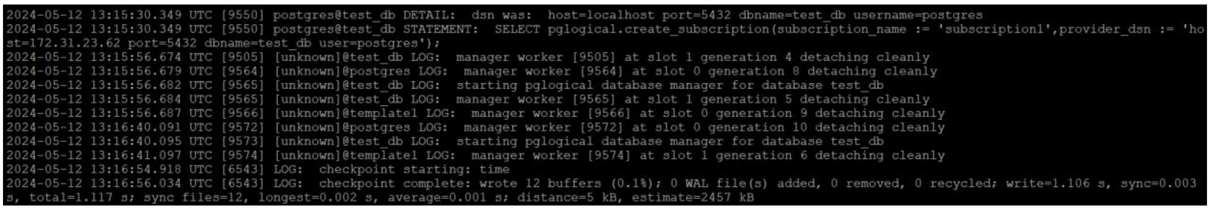


```
postgres@ip-172-31-27-122:/var/log/postgresql$ psql -d test_db
psql (15.7 (Ubuntu 15.7-1.pgdg24.04+1))
Type "help" for help.

test_db=# SELECT subscription_name, status FROM pglogical.show_subscription_status();
 subscription_name |   status
-------------------+-------------
 subscription1     | replicating
(1 row)
```

#Check logs

cd /var/log/postgresql/

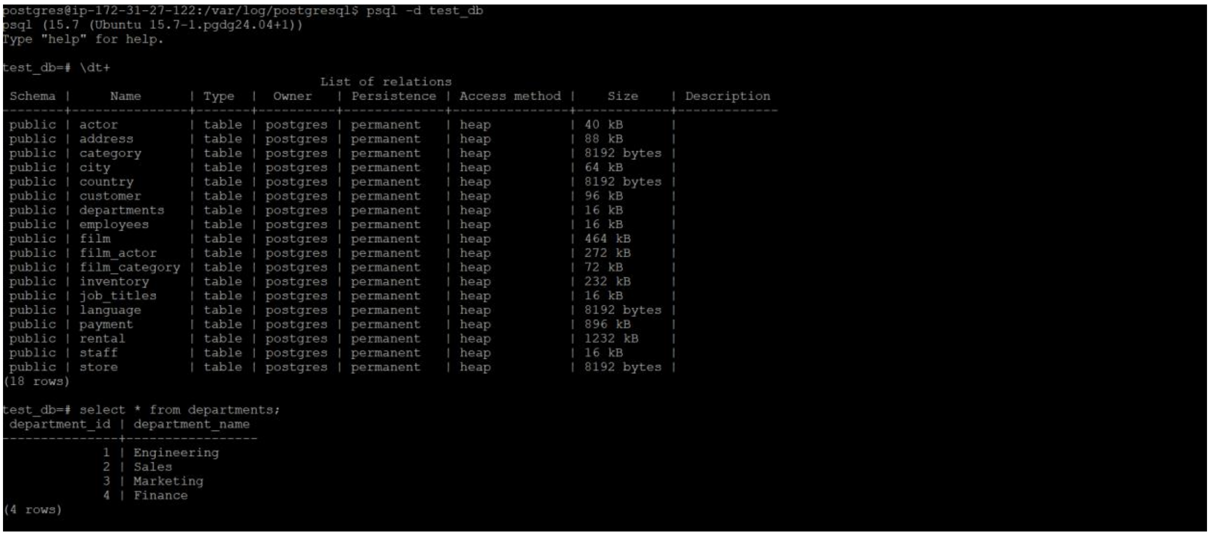
tail -f postgresql-15-main.log



```
2024-05-12 13:15:30.349 UTC [9550] postgres@test_db DETAIL:  dsn was:  host=localhost port=5432 dbname=test_db username=postgres
2024-05-12 13:15:30.349 UTC [9550] postgres@test_db STATEMENT:  SELECT pglogical.create_subscription(subscription_name := 'subscription1',provider_dsn := 'ho
st=172.31.23.62 port=5432 dbname=test_db user=postgres');
2024-05-12 13:15:56.674 UTC [9505] [unknown]@test_db LOG:  manager worker [9505] at slot 1 generation 4 detaching cleanly
2024-05-12 13:15:56.679 UTC [9564] [unknown]@postgres LOG:  manager worker [9564] at slot 0 generation 8 detaching cleanly
2024-05-12 13:15:56.682 UTC [9565] [unknown]@test_db LOG:  starting pglogical database manager for database test_db
2024-05-12 13:15:56.684 UTC [9565] [unknown]@test_db LOG:  manager worker [9565] at slot 1 generation 5 detaching cleanly
2024-05-12 13:15:56.687 UTC [9566] [unknown]@template1 LOG:  manager worker [9566] at slot 0 generation 9 detaching cleanly
2024-05-12 13:16:40.091 UTC [9572] [unknown]@postgres LOG:  manager worker [9572] at slot 0 generation 10 detaching cleanly
2024-05-12 13:16:40.095 UTC [9573] [unknown]@test_db LOG:  starting pglogical database manager for database test_db
2024-05-12 13:16:41.097 UTC [9574] [unknown]@template1 LOG:  manager worker [9574] at slot 1 generation 6 detaching cleanly
2024-05-12 13:16:54.918 UTC [6543] LOG:  checkpoint starting: time
2024-05-12 13:16:56.034 UTC [6543] LOG:  checkpoint complete: wrote 12 buffers (0.1%); 0 WAL file(s) added, 0 removed, 0 recycled; write=1.106 s, sync=0.003
s, total=1.117 s; sync files=12, longest=0.002 s, average=0.001 s; distance=5 kB, estimate=2457 kB
```

#Verify tables data

\dt+

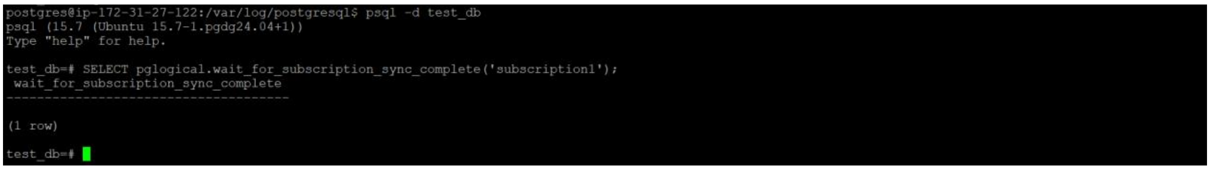


```
postgres@ip-172-31-27-122:/var/log/postgresql$ psql -d test_db
psql (15.7 (Ubuntu 15.7-1.pgdg24.04+1))
Type "help" for help.

test_db=# \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 | departments   | table | postgres | permanent   | heap          | 16 kB      |
 public | employees     | table | postgres | permanent   | heap          | 16 kB      |
 public | film          | table | postgres | permanent   | heap          | 464 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 | job_titles    | table | postgres | permanent   | heap          | 16 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 |
(18 rows)

test_db=# select * from departments;
 department_id | department_name
---------------+-----------------
             1 | Engineering
             2 | Sales
             3 | Marketing
             4 | Finance
(4 rows)
```

#Check replication completed

SELECT pglogical.wait_for_subscription_sync_complete('subscription1');

Note: When you execute this function and it returns empty, it means that the subscription synchronization has already completed or there are no pending synchronization tasks to wait for.



```
postgres@ip-172-31-27-122:/var/log/postgresql$ psql -d test_db
psql (15.7 (Ubuntu 15.7-1.pgdg24.04+1))
Type "help" for help.

test_db=# SELECT pglogical.wait_for_subscription_sync_complete('subscription1');
 wait_for_subscription_sync_complete
-------------------------------------

(1 row)

test_db=#
```