# POSTGRESQL MIGRATION

BY USING TERMINAL

A.) Stabilization and Installation of Postgresql

B.) Copying and Modification of the data directory

C.) Install Postgresql contrib and the extension of the old Postgresql

## A.) Stabilization and Installation of Postgresql

i. Ensure your database is up to date and replication is going well

ii. Old server 172.19.6.17 New Server 172.19.6.147

iii. Check the version of the postgresql in the old server: psql –V

iv. Install the same version of the postgresql in the new server

v. Initialize the new postgres in the new server: /usr/pgsql-11/bin/pg_ctl -D /home/pg_data/pgdata/ initdb

vi. Enable & Start the services: systemctl enable/start postgresql-11.service

vii. Ensure everything is working perfectly

viii. Finally Wipe the data directory: rm –rf /home/pg_data/pgdata

## B.) Copying and Migration of Postgresql

i. From the old server 6.17 copy the data directory to the new server 6.147 : **scp -r /data/pgdata/ root@172.19.6.147:/home/pg_data/11**

ii. This is for moving my backup file from production to my testing server. (scp -r /data/backup1/PayArenaV2_databases.sql.gz root@172.19.6.147:/home/pg_restore) pls skip this

iii. Do everything now at the new server 6.147 change ownership using chown –R postgres:postgres /pg_data/pgdata/

iv. Grant privilege chmod 700 /pg_data/pgdata/

v. Navigate to the postgres service: vi or nano /usr/lib/systemd/system/postgresql-11.service

vi. Set environment to /pg_data/pgdata/

vii. Goto postgres.conf and edit IP (located at listen_address) put the new server IP there and port(not all that necessary)

viii. Reload system daemon – systemctl daemon-reload

## C.) Install Postgresql contrib and the extension of the old postgresql

i. List postgrersql and select the version: Yum install list module postgresql*

ii. Install the postgresql contrib : yum install postgresql11-contrib.x86_64

iii. Go to postgres.conf at share_preload_libraries and check for the extension file might be missing which are: pgaudit, pg_statsinfo, pg_qualstats so go to old server and run this one by one rpm –qa | grep pgaudit, rpm –qa | grep pg_statsinfo, rpm –qa | grep pg_qualstats.

After installation start the server: systemctl start postgresql-11.service. Incase if the server didn’t start change user to postgres:  su – postgres and start the server via : /usr/pgsql-11/bin/pg_ctl -D /home/pg_data/pgdata/ start

Ubuntu : /usr/lib/postgresql/16/bin/pg_ctl -D /directory/ start

**************************************************************************

MY NONSENSE JOTTING PLEASE IGNORE THIS

Or st check missing extension : yum list available | ‘grep extentionname’

/usr/pgsql-11/bin/pg_ctl -D /home/pg_data/11/pgdata/ status

Your installation contains user-defined objects that refer to internal
polymorphic functions with arguments of type "anyarray" or "anyelement".
These user-defined objects must be dropped before upgrading and restored
afterwards, changing them to refer to the new corresponding functions with
arguments of type "anycompatiblearray" and "anycompatible".
A list of the problematic objects is in the file:

THE NONSENSE STOPPED NOW

****************************************************************************

## POSTGRESQL UPGRADE

Using PayareanaV2 (6.147) as case study from v11 to v15

*************************

BEFORE THE UPGRADE

*************************

Download the latest postgres version at www.postgres.org

Install the postgres: yum install postgresql15.x86_64

Create the directory you wish your data file to be:  mkdir /home/pg_data/pgdata15

Give postgres ownership to the directory: chown postgres:postgres –R /home/pg_data/pgdata15

Give permission to the directory: chmod 700 –R /home/pg_data/pgdata15

NB: -R means recursive to give entire directory ownership or permission

Go to the environmental directory to put the directory there: nano /usr/lib/system/system/postgresql-15.service

Look for Environment=PGDATA. Remove the directory there and place yours i.e. /home/pg_data/pgdata15

Initialize the DB via /usr/lib/pgsql-15/bin/postgresql-15-setup initdb

Change the port so that it will be different from V11. nano /home/pg_data/pgdata15/postgresql.conf. Look for port, remove the # in front of it to make it commentable and input 1701. This is the port of my choice

Enable the service: systemctl enable postgresql-15.service

Start the service:  systemctl start postgresql-15.service

Go to postgres.conf on V11 to know the extension files you need to install: nano /home/pg_data/11/pgdata/postgres.conf. Look for shared_preload_libraries. In payarena I saw pgaudit, pg_statsinfo, pg_stat_statements, pg_qualstats.

Search this extensions via: yum list available | grep pgaudit Ensure to get the same version with your postgres and install it yum install pgaudit15_13.x86_64. Do same to the rest of the extension. When searching and you didn’t see it just know it is not there.

*********************

START THE UPGRADE

*********************

i. Stop the two service on v11 and 15: systemctl stop postgresql-11.service
ii. Enter the directory of v11: cd /home/pg_data/11/pgdata
iii. Login as postgres user: su postgres. I used this instead of su – postgres so that I maintained current directory
iv. Run compatibility check of the postgres: /usr/pgsql-15/bin/pg_upgrade -b /usr/pgsql-11/bin -B /usr/pgsql-15/bin -d /home/pg_data/11/pgdata -D /home/pg_data/pgdata15 -c
v. Any error encountered here try to solve it and rerun the check compactible script
vi. If no error Run this to upgrade: /usr/pgsql-15/bin/pg_upgrade -b /usr/pgsql-11/bin -B /usr/pgsql-15/bin -d /home/pg_data/11/pgdata -D /home/pg_data/pgdata15

SOLVED ERROR I ENCOUNTERED ON IV

tablespace directory “pg_tblspc/19888” does not exist

To solve this delete the directory: rm -rf /home/pg_data/11/pgdata/pg_tblspc

Create an empty directory of it: mkdir /home/pg_data/11/pgdata/pg_tblspc/19888

Login to v11 database and drop these: drop schema statsrepo cascade
drop schema statsinfo cascade
drop schema statsinfo

Go to shared_preload_libraries on postgresql.con v15 : nano /home/pg_data/pgdata15/postgresql.conf

Uncomment shared_preload _libraries and add only pgaudit save and quit

Then rerun this script: /usr/pgsql-15/bin/pg_upgrade -b /usr/pgsql-11/bin -B /usr/pgsql-15/bin -d /home/pg_data/11/pgdata -D /home/pg_data/pgdata15 -c. if no error run this script to upgrade: /usr/pgsql-15/bin/pg_upgrade -b /usr/pgsql-11/bin -B /usr/pgsql-15/bin -d /home/pg_data/11/pgdata -D /home/pg_data/pgdata15 –

AFTER THE UPGRADE THE SCHEMA I DROP YOU HAVE TO RECREATE IT BACK

TO CREATE THE STATSREPO CASCADE:

1.) Create statsrepo: CREATE SCHEMA statsrepo;

2.) Grant privileges to the user who will be accessing the statsrepo schema:

```
GRANT USAGE, CREATE ON SCHEMA statsrepo TO postgres;
GRANT USAGE, SELECT ON SCHEMA statsrepo TO postgres;
GRANT USAGE, SELECT ON ALL TABLES IN SCHEMA statsrepo TO postgres;
```

3.) Create the statsrepo tables using the following commands:

```
CREATE TABLE statsrepo.statistics (
    schemaname text,
    relname text,
    seq_scan bigint,
    seq_tup_read bigint,
    idx_scan bigint,
    idx_tup_fetch bigint,
    n_tup_ins bigint,
    n_tup_upd bigint,
    n_tup_del bigint,
    n_tup_hot_upd bigint,
    n_live_tup bigint,
    n_dead_tup bigint,
    last_vacuum timestamp without time zone,
    last_autovacuum timestamp without time zone,
    last_analyze timestamp without time zone,
    last_autoanalyze timestamp without time zone,
    vacuum_count bigint,
    autovacuum_count bigint,
    analyze_count bigint,
    autoanalyze_count bigint
);
```

4.)

```
CREATE TABLE statsrepo.pg_stat_activity (
    datid oid,
    datname name,
    pid integer,
    usesysid oid,
    usename name,
    application_name text,
    client_addr inet,
    client_hostname text,
    client_port integer,
    backend_start timestamp without time zone,
    xact_start timestamp without time zone,
    query_start timestamp without time zone,
    state_change timestamp without time zone,
    wait_event_type text,
    wait_event text,
    state text,
    backend_xid integer,
    backend_xmin integer,
    query text,
    backend_type text
);
```

5.) Create the appropriate indexes and constraints as needed:

```
CREATE UNIQUE INDEX statsrepo_statistics_pkey ON statsrepo.statistics(schemaname, relname);
CREATE UNIQUE INDEX statsrepo_pg_stat_activity_pkey ON statsrepo.pg_stat_activity(pid);
```

To create schema statsinfo cascade :

i) CREATE SCHEMA statsinfo;

ii) Create the table

```
CREATE TABLE statsinfo.statistics (
    tablename text,
    num_rows bigint,
    avg_row_length double precision,
    data_length bigint,
    max_data_length bigint,
    index_length bigint,
    data_free bigint,
    auto_increment bigint,
    create_time timestamp without time zone,
    update_time timestamp without time zone,
    check_time timestamp without time zone,
    table_collation text,
    checksum bigint,
    create_options text,
    table_comment text
);
```

```
GRANT USAGE ON SCHEMA statsinfo TO postgres;
GRANT SELECT ON TABLE statsinfo.statistics TO postgres;
```