POSTGRESQL MIGRATION

BY USING TERMINAL

1. Stabilization and Installation of Postgresql
2. Copying and Modification of the data directory
3. Install Postgresql contrib and the extension of the old Postgresql
4. Stabilization and Installation of Postgresql
5. Ensure your database is up to date and replication is going well
6. Old server 172.19.6.17 New Server 172.19.6.147
7. Check the version of the postgresql in the old server: psql –V
8. Install the same version of the postgresql in the new server
9. Initialize the new postgres in the new server: /usr/pgsql-11/bin/pg\_ctl -D /home/pg\_data/pgdata/ initdb
10. Enable & Start the services: systemctl enable/start postgresql-11.service
11. Ensure everything is working perfectly
12. Finally Wipe the data directory: rm –rf /home/pg\_data/pgdata
13. Copying and Migration of Postgresql
14. 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**](about:blank)
15. 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](about:blank)) pls skip this
16. Do everything now at the new server 6.147 change ownership using chown –R postgres:postgres /pg\_data/pgdata/
17. Grant privilege chmod 700 /pg\_data/pgdata/
18. Navigate to the postgres service: vi or nano /usr/lib/systemd/system/postgresql-11.service
19. Set environment to /pg\_data/pgdata/
20. Goto postgres.conf and edit IP (located at listen\_address) put the new server IP there and port(not all that necessary)
21. Reload system daemon – systemctl daemon-reload
22. Install Postgresql contrib and the extension of the old postgresql
23. List postgrersql and select the version: Yum install list module postgresql\*
24. Install the postgresql contrib : yum install postgresql11-contrib.x86\_64
25. 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](http://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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Stop the two service on v11 and 15: systemctl stop postgresql-11.service
2. Enter the directory of v11: cd /home/pg\_data/11/pgdata
3. Login as postgres user: su postgres. I used this instead of su – postgres so that I maintained current directory
4. 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
5. Any error encountered here try to solve it and rerun the check compactible script
6. 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 :

1. CREATE SCHEMA statsinfo;
2. 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;