Setup replication using repmgr

Streaming replication is a common PostgreSQL high-availability setup, consisting of a primary server (read-write) and one or more standby replicas (read-only). However, a key limitation of streaming replication is that it doesn't automatically failover to a replica if the primary server goes down, requiring manual intervention from a DBA to promote the replica to read-write mode. This also means the setup needs constant monitoring.
This is where `repmgr` comes into play, addressing this critical issue by automating failover to a replica in the event of a primary server failure, eliminating the need for DBA intervention

Installing repmgr and PostgreSQL

The setup below will be performed on two VMs running Ubuntu 22, with PostgreSQL 16 being used for the configuration.

hostname	ip address
postgresql-db01	10.217.10.6
postgresql-db02	10.217.10.7

Installing PostgreSQL 16

Preform the below steps in both DB servers

1. 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
```

2. 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'
```

3. Update the package lists:

```
sudo apt update
```

4. Install PostgreSQL 12

```
sudo apt -y install postgresql-16
```

```
dba@postgresql-db01:~$ sudo curl -o /usr/share/postgresql-common/pgdg/apt.postgresql.org.asc --fail https://www.postgresql.org/media/keys/ACCC4CF8.asc
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  4812  100  4812    0     0   8467      0 --:--:-- --:--:-- --:--:--  8456
dba@postgresql-db01:~$ sudo sh -c 'echo "deb [signed-by=/usr/share/postgresql-common/pgdg/apt.postgresql.org.asc] https://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.li
st.d/pgdg.list'
dba@postgresql-db01:~$ sudo apt update
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:5 https://apt.postgresql.org/pub/repos/apt jammy-pgdg InRelease [129 kB]
Get:6 https://apt.postgresql.org/pub/repos/apt jammy-pgdg/main amd64 Packages [343 kB]
Fetched 472 kB in 2s (235 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
136 packages can be upgraded. Run 'apt list --upgradable' to see them.
dba@postgresql-db01:~$ sudo apt -y install postgresql-12
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcommon-sense-perl libjson-perl libjson-xs-perl libllvm15 libpq5 libsensors-config libsensors5 libtypes-serialiser-perl postgresql-client-12 postgresql-client-common postgresql-common ssl-cert sysstat
Suggested packages:
  lm-sensors postgresql-doc-12 isag
The following NEW packages will be installed:
  libcommon-sense-perl libjson-perl libjson-xs-perl libllvm15 libpq5 libsensors-config libsensors5 libtypes-serialiser-perl postgresql-12 postgresql-client-12 postgresql-client-common postgresql-common
  ssl-cert sysstat
0 upgraded, 14 newly installed, 0 to remove and 136 not upgraded.
Need to get 43.6 MB of archives.
After this operation, 177 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 libjson-perl all 4.04000-1 [81.8 kB]
Get:2 https://apt.postgresql.org/pub/repos/apt jammy-pgdg/main amd64 postgresql-client-common all 262.pgdg22.04+1 [94.8 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 ssl-cert all 1.1.2 [17.4 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/main amd64 libcommon-sense-perl amd64 3.75-2build1 [21.1 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/main amd64 libtypes-serialiser-perl all 1.01-1 [11.6 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/main amd64 libjson-xs-perl amd64 4.030-1build3 [87.2 kB]
Get:7 https://apt.postgresql.org/pub/repos/apt jammy-pgdg/main amd64 postgresql-common all 262.pgdg22.04+1 [240 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libllvm15 amd64 1:15.0.7-0ubuntu0.22.04.3 [25.4 MB]
Get:9 https://apt.postgresql.org/pub/repos/apt jammy-pgdg/main amd64 libpq5 amd64 16.4-1.pgdg22.04+1 [217 kB]
Get:10 https://apt.postgresql.org/pub/repos/apt jammy-pgdg/main amd64 postgresql-client-12 amd64 12.20-1.pgdg22.04+1 [1439 kB]
Get:11 https://apt.postgresql.org/pub/repos/apt jammy-pgdg/main amd64 postgresql-12 amd64 12.20-1.pgdg22.04+1 [15.5 MB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/main amd64 libsensors-config all 1:3.6.0-7ubuntu1 [5274 B]
Get:13 http://archive.ubuntu.com/ubuntu jammy/main amd64 libsensors5 amd64 1:3.6.0-7ubuntu1 [26.3 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 sysstat amd64 12.5.2-2ubuntu0.2 [487 kB]
Fetched 43.6 MB in 7s (6386 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libjson-perl.
```

baXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

5. Start the services

```
sudo systemctl enable --now  postgresql@16-main.service

sudo systemctl status postgresql@16-main.service
```

```
dba@postgresql-db01:/etc/systemd$ sudo systemctl enable --now  postgresql@16-main.service
Created symlink /etc/systemd/system/multi-user.target.wants/postgresql@16-main.service → /lib/systemd/system/postgresql@.service.
dba@postgresql-db01:/etc/systemd$ sudo systemctl status  --now  postgresql@16-main.service
● postgresql@16-main.service - PostgreSQL Cluster 16-main
     Loaded: loaded (/lib/systemd/system/postgresql@.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2024-09-23 13:50:50 UTC; 1min 46s ago
   Main PID: 8095 (postgres)
      Tasks: 6 (limit: 2220)
     Memory: 19.3M
        CPU: 342ms
     CGroup: /system.slice/system-postgresql.slice/postgresql@16-main.service
             ├─8095 /usr/lib/postgresql/16/bin/postgres -D /var/lib/postgresql/16/main -c config_file=/etc/postgresql/16/main/postgresql.conf
             ├─8096 "postgres: 16/main: checkpointer " "" "" "" "" "" "" "" "" "" "" "" ""
             ├─8097 "postgres: 16/main: background writer " "" "" "" "" "" "" "" "" "" ""
             ├─8099 "postgres: 16/main: walwriter " "" "" "" "" "" "" "" "" "" "" "" ""
             ├─8100 "postgres: 16/main: autovacuum launcher " "" "" "" "" "" "" "" "" ""
             └─8101 "postgres: 16/main: logical replication launcher " "" "" "" "" "" ""

Sep 23 13:50:48 postgresql-db01 systemd[1]: Starting PostgreSQL Cluster 16-main...
Sep 23 13:50:50 postgresql-db01 systemd[1]: Started PostgreSQL Cluster 16-main.
lines 1-17/17 (END)
dba@postgresql-db01:/etc/systemd$
```

installing repmgr

install `repmgr` on both primary and secondary server

```
sudo apt install repmgr
```

configuring primary server

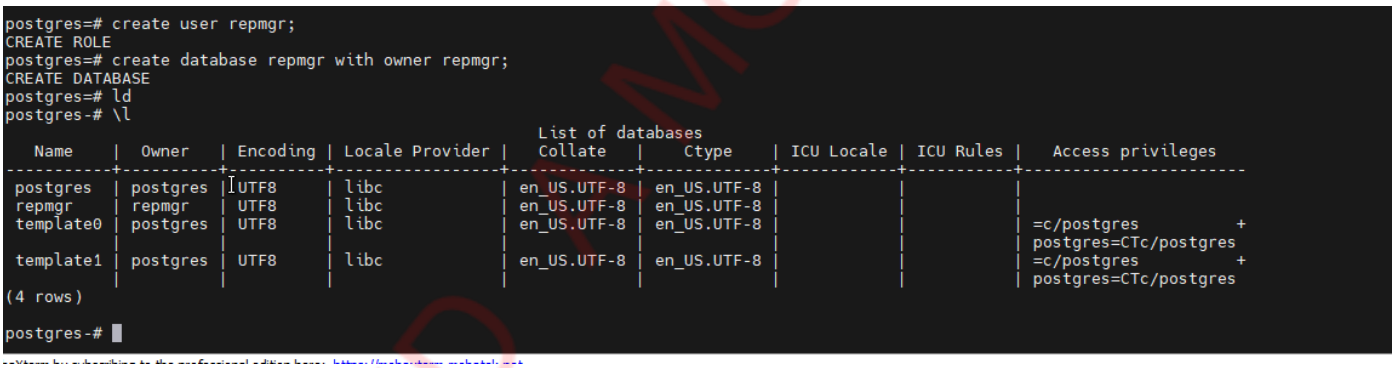
On the primary server, some parameters need to be adjusted in the `postgresql.conf` file, which can be found in `/etc/postgresql/16/main`. Below are the parameters that need to be edited

```
sudo nano /etc/postgresql/16/main/postgresql.conf
```

```
listen_addresses = '*'
max_wal_senders = 10

max_replication_slots = 10

wal_level = 'hot_standby' or 'replica' or 'logical'

hot_standby = on

archive_mode = on

archive_command = '/bin/true'

shared_preload_libraries = 'repmgr'
```

On the primary server, create another superuser called `repmgr` and create a database assigned to the `repmgr` user. This database will be used by `repmgr` for storing statistics.

```
create user repmgr;
ALTER USER repmgr WITH SUPERUSER;


create database repmgr with owner repmgr;
```



next update the `pg_hba.conf` file with connection allowed in both ipv4 and replication section as follow

```
sudo vi /etc/postgresql/16/main/pg_hba.conf
```

```
 local      replication      repmgr                                      trust

    host       replication      repmgr        127.0.0.1/32            trust

    host       replication      repmgr        10.217.10.0/24          trust


    local       repmgr           repmgr                                      trust

    host        repmgr           repmgr        127.0.0.1/32            trust
```

```
host        repmgr          repmgr        10.217.10.0/24       trust
```

```
# Database administrative login by Unix domain socket
local   all             postgres                                peer

# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
local   repmgr          repmgr                                  trust
# IPv4 local connections:
host    all             all             127.0.0.1/32            scram-sha-256
host    repmgr          repmgr          10.217.10.0/24          trust
host    repmgr          repmgr          127.0.0.1/32            trust
# IPv6 local connections:
host    all             all             ::1/128                 scram-sha-256
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     peer
host    replication     all             127.0.0.1/32            scram-sha-256
host    replication     all             ::1/128                 scram-sha-256
host    replication     repmgr          10.217.10.0/24          trust
-- INSERT --
```

Restart PostgreSQL services to get the all configuration loaded

```
sudo systemctl restart postgresql@16-main.service
```

Create a `repmgr.conf` on the master server with the following entire you can place the file in `/etc` directory

```
sudo nano /etc/repmgr.conf
```

```
cluster='failovertest'

node_id=1

node_name=node1

conninfo='host=10.217.10.6 user=repmgr dbname=repmgr connect_timeout=2'

data_directory='/var/lib/postgresql/16/main/'

failover=automatic

promote_command='/usr/bin/repmgr standby promote -f /etc/repmgr.conf --log-to-file'

follow_command='/usr/bin/repmgr standby follow -f /etc/repmgr.conf --log-to-file --upstream-node-id=%n'
```

"Now we will start by registering the primary server with `repmgr` using the following command:

```
repmgr -f /etc/repmgr.conf primary register
```

```
dba@postgresql-db01:~$ repmgr -f /etc/repmgr.conf primary register
WARNING: the following problems were found in the configuration file:
  parameter "cluster" is deprecated and will be ignored
INFO: connecting to primary database...
NOTICE: attempting to install extension "repmgr"
NOTICE: "repmgr" extension successfully installed
NOTICE: primary node record (ID: 1) registered
dba@postgresql-db01:~$
```

Check the status of the cluster by running the following command

```
repmgr -f /etc/repmgr.conf cluster  show
```

```
dba@postgresql-db01:~$ repmgr -f /etc/repmgr.conf cluster  show
WARNING: the following problems were found in the configuration file:
  parameter "cluster" is deprecated and will be ignored
 ID | Name  | Role    | Status    | Upstream | Location | Priority | Timeline | Connection string
----+-------+---------+-----------+----------+----------+----------+----------+-----------------------------------------------------------------
 1  | node1 | primary | * running |          | default  | 100      | 1        | host=10.217.10.6 user=repmgr dbname=repmgr connect_timeout=2
dba@postgresql-db01:~$
```

Configuring replica server server

1. Create the `repmgr.conf` file and populate it with the following parameters. I have placed the file in the `/etc/` directory.

```
sudo nano /etc/repmgr.conf
```

```
node_id=2

node_name=node2

conninfo='host=10.217.10.7 user=repmgr dbname=repmgr connect_timeout=2'
data_directory='/var/lib/postgresql/16/main/'
failover=automatic

promote_command='/usr/bin/repmgr  standby promote -f /etc/repmgr.conf --log-to-file'

follow_command='/usr/bin/repmgr standby follow -f /etc/repmgr.conf --log-to-file --upstream-node-id=%n
```

2 . Stop PostgreSQL services in replica server and go to the data directory and remove all the files

```
sudo systemctl stop postgresql@16-main.service
sudo -i

cd  /var/lib/postgresql/16/main
rm -rf *
```

```
dba@postgresql-db02:~$ sudo systemctl stop postgresql@16-main.service
dba@postgresql-db02:~$ sudo systemctl status  postgresql@16-main.service
○ postgresql@16-main.service - PostgreSQL Cluster 16-main
     Loaded: loaded (/lib/systemd/system/postgresql@.service; enabled; vendor preset: enabled)
     Active: inactive (dead) since Mon 2024-09-23 15:08:55 UTC; 6s ago
    Process: 5572 ExecStop=/usr/bin/pg_ctlcluster --skip-systemctl-redirect -m fast 16-main stop (code=exited, status=0/SUCCESS)
   Main PID: 5067 (code=exited, status=0/SUCCESS)
        CPU: 705ms

Sep 23 14:47:41 postgresql-db02 systemd[1]: Starting PostgreSQL Cluster 16-main...
Sep 23 14:47:43 postgresql-db02 systemd[1]: Started PostgreSQL Cluster 16-main.
Sep 23 15:08:55 postgresql-db02 systemd[1]: Stopping PostgreSQL Cluster 16-main...
Sep 23 15:08:55 postgresql-db02 systemd[1]: postgresql@16-main.service: Deactivated successfully.
Sep 23 15:08:55 postgresql-db02 systemd[1]: Stopped PostgreSQL Cluster 16-main.
dba@postgresql-db02:~$ sudo -i
root@postgresql-db02:~# cd  /var/lib/postgresql/16/main
root@postgresql-db02:/var/lib/postgresql/16/main# rm -rf *
root@postgresql-db02:/var/lib/postgresql/16/main# ls
root@postgresql-db02:/var/lib/postgresql/16/main#
```

3. now we can perform `dry run` which test our standby server configuration before we can add it t the cluster

```
sudo su - postgres

repmgr -h 10.217.10.6 -U repmgr -d repmgr -f /etc/repmgr.conf standby clone --dry-run
```

```
root@postgresql-db02:/var/lib/postgresql/16/main# sudo su - postgres
postgres@postgresql-db02:~$ repmgr -h 10.217.10.6 -U repmgr -d repmgr -f /etc/repmgr.conf standby clone --dry-run
NOTICE: destination directory "/var/lib/postgresql/16/main" provided
INFO: connecting to source node
DETAIL: connection string is: host=10.217.10.6 user=repmgr dbname=repmgr
DETAIL: current installation size is 29 MB
INFO: "repmgr" extension is installed in database "repmgr"
INFO: replication slot usage not requested;  no replication slot will be set up for this standby
INFO: parameter "max_wal_senders" set to 10
NOTICE: checking for available walsenders on the source node (2 required)
INFO: sufficient walsenders available on the source node
DETAIL: 2 required, 10 available
NOTICE: checking replication connections can be made to the source server (2 required)
INFO: required number of replication connections could be made to the source server
DETAIL: 2 replication connections required
WARNING: data checksums are not enabled and "wal_log_hints" is "off"
DETAIL: pg_rewind requires "wal_log_hints" to be enabled
NOTICE: standby will attach to upstream node 1
HINT: consider using the -c/--fast-checkpoint option
INFO: would execute:
  pg_basebackup -l "repmgr base backup"  -D /var/lib/postgresql/16/main -h 10.217.10.6 -p 5432 -U repmgr -X stream
INFO: all prerequisites for "standby clone" are met
postgres@postgresql-db02:~$
```

4. Start cloning of the data directory from primary server by running the below command

```
repmgr -h 10.217.10.6 -U repmgr -d repmgr -f /etc/repmgr.conf standby clone
```

```
postgres@postgresql-db02:~$ repmgr -h 10.217.10.6 -U repmgr -d repmgr -f /etc/repmgr.conf standby clone
NOTICE: destination directory "/var/lib/postgresql/16/main" provided
INFO: connecting to source node
DETAIL: connection string is: host=10.217.10.6 user=repmgr dbname=repmgr
DETAIL: current installation size is 29 MB
INFO: replication slot usage not requested;  no replication slot will be set up for this standby
NOTICE: checking for available walsenders on the source node (2 required)
NOTICE: checking replication connections can be made to the source server (2 required)
WARNING: data checksums are not enabled and "wal_log_hints" is "off"
DETAIL: pg_rewind requires "wal_log_hints" to be enabled
INFO: checking and correcting permissions on existing directory "/var/lib/postgresql/16/main"
NOTICE: starting backup (using pg_basebackup)...
HINT: this may take some time; consider using the -c/--fast-checkpoint option
INFO: executing:
  pg_basebackup -l "repmgr base backup"  -D /var/lib/postgresql/16/main -h 10.217.10.6 -p 5432 -U repmgr -X stream
NOTICE: standby clone (using pg_basebackup) complete
NOTICE: you can now start your PostgreSQL server
HINT: for example: pg_ctl -D /var/lib/postgresql/16/main start
HINT: after starting the server, you need to register this standby with "repmgr standby register"
postgres@postgresql-db02:~$
```

5. start PostgreSQL 16 services

```
systemctl start  postgresql@16-main.service
```

6. Update the `postgresql.conf` and `pg_hba.conf` files to match the configuration parameters used on the primary server.
7. Register the standby server to the cluster

```
repmgr -f /etc/repmgr.conf standby register
```

```
postgres@postgresql-db02:~$ repmgr -f /etc/repmgr.conf standby register
INFO: connecting to local node "node2" (ID: 2)
INFO: connecting to primary database
WARNING: --upstream-node-id not supplied, assuming upstream node is primary (node ID: 1)
INFO: standby registration complete
NOTICE: standby node "node2" (ID: 2) successfully registered
postgres@postgresql-db02:~$
```

```
postgres@postgresql-db02:~$ repmgr -f /etc/repmgr.conf cluster show
 ID | Name  | Role    | Status    | Upstream | Location | Priority | Timeline | Connection string
----+-------+---------+-----------+----------+----------+----------+----------+-----------------------------------------------------------------
 1  | node1 | primary | * running |          | default  | 100      | 1        | host=10.217.10.6 user=repmgr dbname=repmgr connect_timeout=2
 2  | node2 | standby |   running | node1    | default  | 100      | 1        | host=10.217.10.7 user=repmgr dbname=repmgr connect_timeout=2
postgres@postgresql-db02:~$
```

8. check the replication by creating testdb in primary and check weather it get replicated to the standby server

```
sudo -u postgres psql

create database testdb;
```

```
dba@postgresql-db01:~$ sudo -u postgres psql
[sudo] password for dba:
psql (16.4 (Ubuntu 16.4-1.pgdg22.04+1))
Type "help" for help.

postgres=# create database testdb;
CREATE DATABASE
postgres=# \l
                                                  List of databases
   Name    |  Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access privileges
-----------+----------+----------+-----------------+-------------+-------------+------------+-----------+-----------------------
 postgres  | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
 repmgr    | repmgr   | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
 template0 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres          +
           |          |          |                 |             |             |            |           | postgres=CTc/postgres
 template1 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres          +
           |          |          |                 |             |             |            |           | postgres=CTc/postgres
 testdb    | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
(5 rows)

postgres=#
```

```
postgres@postgresql-db02:~$ psql
psql (16.4 (Ubuntu 16.4-1.pgdg22.04+1))
Type "help" for help.

postgres=# \l
                                                  List of databases
   Name    |  Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access privileges
-----------+----------+----------+-----------------+-------------+-------------+------------+-----------+-----------------------
 postgres  | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
 repmgr    | repmgr   | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
 template0 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres          +
           |          |          |                 |             |             |            |           | postgres=CTc/postgres
 template1 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres          +
           |          |          |                 |             |             |            |           | postgres=CTc/postgres
 testdb    | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
(5 rows)

postgres=#
```

I would like to extend my thanks to EDB for their excellent guide. You can view it yourself by visiting the following URL

https://www.enterprisedb.com/postgres-tutorials/how-implement-repmgr-postgresql-automatic-failover