Belisama

Configuring PowerSC MFA High Availability.

I was thrown an interesting challenge a short while ago, and thought I would share the solution. While the IBM PowerSC MFA 1.3 installation and configuration guide covers setting up PowerSC MFA for high availability, it requires shared storage to be configured between the two MFA Servers. Unfortunately my customer only had IP connectivity between their two data centres, and I don't think that they would be impressed if I had added complexity by including GLVM replication in the mix.
As the only component that needs to be replicated is the Postgres database, I thought I would configure and test Postgres replication (warm standby) and confirm that it will work with PowerSC. The steps to configure Postgres replication for PowerSC MFA and switch sites is provided below. Please feel free to contract me if you have further questions.

Antony Steel
Belisama Pte. Ltd.
antony.steel@belisama.com.sg

Postgres Write Ahead Logs (WAL)

Postgres replication was configured using Postgres WAL (write ahead logging). Replication was configured using the Private network:

Server	Private
server_prod	192.168.1.1
server_dr	192.169.1.2

The first step was to create a postgres replication user as root:

```
sudo -u postgres createuser -U postgres -P -c 5 --replication replicamfa
```

and then set the password.

Configuring replication

As the postgres user, configure replication on the primary node (Stop the MFA daemon before making changes).

On the Primary node:
Stop the database:

```
pg_ctl -D /opt/IBM/powersc/MFA/mfadb stop
```

Modify the `/opt/IBM/powersc/MFA/mfadb/pg_hba.conf` file, as in Text 1

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     trust
# IPv4 local connections:
host    all             all             127.0.0.1/32            trust
host    all             all             192.169.1.2/32          trust
host    all             all             192.168.1.1/32          trust
# IPv6 local connections:
host    all             all             ::1/128                 trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     trust
host    replication     all             127.0.0.1/32            trust
host    replication     all             ::1/128                 trust
host    replication    replicamfa     192.168.1.1/32 md5
host    replication    replicamfa     192.169.1.2/32 md5
```

Text 1: pg_hba.conf

Modify `/opt/IBM/powersc/MFA/mfadb/postgresql.conf` file, as in Text 2

```
max_connections = 100                # (change requires restart)
shared_buffers = 128MB               # min 128kB
dynamic_shared_memory_type = posix   # the default is the first option
wal_level = replica                  # minimal, replica, or logical
wal_log_hints = on                   # also do full page writes of non-
critical updates
max_wal_senders = 10          # max number of walsender processes
wal_keep_segments = 10        # in logfile segments, 16MB each; 0 disables
wal_receiver_status_interval = 5s    # send replies at least this often
hot_standby_feedback = on            # send info from standby to prevent
log_timezone = 'Asia/Singapore'
datestyle = 'iso, mdy'
timezone = '[your TZ]'
lc_messages = 'en_US'                # locale for system error message
lc_monetary = 'en_US'                # locale for monetary formatting
lc_numeric = 'en_US'                 # locale for number formatting
lc_time = 'en_US'                           # locale for time formatting
default_text_search_config = 'pg_catalog.english'
```

Text 2: postgresql.conf

Create the replication slots

```
select * from pg_create_physical_replication_slot('prod_slot');
select * from pg_create_physical_replication_slot('dr_slot');
```

Start the database

```
pg_ctl -D /opt/IBM/powersc/MFA/mfadb start
```

On the Standby Node:
Make sure that both the pmfad daemon and database are not running then remove all the files in the DB directory (`/opt/IBM/powersc/MFA/mfadb`)

Copy the current DB from the primary Server, see Text 3

```
pg_basebackup -Xs -d "hostaddr= 192.168.1.1 port= 5432 user= replicamfa
password= [replica user password]" -D /opt/IBM/powersc/MFA/mfadb -v -Fp

pg_basebackup: initiating base backup, waiting for checkpoint to complete
pg_basebackup: checkpoint completed
pg_basebackup: write-ahead log start point: 0/2000028 on timeline 1
pg_basebackup: starting background WAL receiver
pg_basebackup: write-ahead log end point: 0/20000F8
pg_basebackup: waiting for background process to finish streaming ...
pg_basebackup: base backup completed
```

Text 3: Replicating initial configuration to standby

Create a recovery configuration file in `/opt/IBM/powersc/MFA/mfadb as in Text 4`

```
standby_mode = on
recovery_target_timeline = 'latest'
primary_conninfo = 'hostaddr= 192.168.1.1 port= 5432 user= replicamfa
password = [replica user password]'
primary_slot_name = 'prod_slot'
trigger_file = pg_replication_trigger
```

Text 4: recovery.conf

Now start the database and confirm that there are no errors

```
pg_ctl -D /opt/IBM/powersc/MFA/mfadb start
```

Start the MFA daemon on the Primary Server and check using the GUI.

Switching active Postgres DB between Servers.

In this example the MFA Primary instance is active on the Production Server and we will perform a smooth switch to the standby server.

In the scenario below, ServerA is the currently active Server and ServerB is the standby

On the ServerA (active) Server

Run the following steps:

1. As root stop the MFA Server:
   `stopsrc -s pmfad`

2. As postgres stop the database:
   `pg_ctl -D /opt/IBM/powersc/MFA/mfadb stop`

3. Check the replica slots exist, see Text 5

```
postgres=#  select slot_name, slot_type, active from
pg_replication_slots;
  slot_name   | slot_type | active
--------------+-----------+--------
 ServerA_slot | physical  | f
 ServerB_slot | physical  | f
(2 rows)
```

Text 5: List the replica slots

On ServerB (the new active) Server

Run the following steps:

1. As the postgres user, promote the database to primary:
   `pg_ctl -D /opt/IBM/powersc/MFA/mfadb promote`

2. As root start the PowerSC MFA daemon
   `startsrc -s pmfad`

3. Check the operation of the MFA GUI

On ServerA (the old active) Server

As the postgres user, run the following steps

1. Perform a recovery against the new Primary server as postgres user:
   `pg_rewind --target-pgdata /opt/IBM/powersc/MFA/mfadb --source-server='host= ServerB port=5432 user=postgres dbname=postgres' -P`

2. Create the recovery configuration file in `/opt/IBM/powersc/MFA/mfadb` as in Text 6:

```
recovery_target_timeline = ‘latest’
standby_mode = on
primary_conninfo = ‘hostaddr= SErverB port=5432 user=replicamfa
password = [replica user password]’
primary_slot_name = ‘ServerB_slot’
```

Text 6: recovery.conf

3. Start the database:
   `pg_ctl -D /opt/IBM/powersc/MFA/mfadb start`

References

IBM PowerSC Standard Edition Version 1.3.0
IBM PowerSC Multi-Factor Authentication Version 1.3.0 Installation and Configuration
IBM PowerSC Multi-Factor Authentication Version 1.3.0 User's Guide
PostgreSQL 10.17 Documentation The PostgreSQL Global Development Group
Postgres warm standby configuration: https://www.postgresql.org/docs/13/warm-standby.html