

PostgreSQL Switchover and Switchback Process

100.125.105.2 172.25.4.2 main-db master

100.125.110.3 172.26.2.3 Maindb-dr slave

Switchover Process

1. Whitelist IP on DR (100.125.110.3/172.26.2.3):

Add the following entries to the pg_hba.conf file to allow replication from the old master (100.125.105.2):

```
host replication all 172.25.4.2/24 trust
host replication all 172.25.4.2/32 trust
```

Then, reload the PostgreSQL configuration with the following command:

```
SELECT pg_reload_conf();
```

2. Stop the PostgreSQL service on the old master (100.125.105.2/172.25.4.2):

```
/usr/lib/postgresql/14/bin/pg_ctl -D /data/pgsql/14/data stop
```

3. Promote the slave (100.125.110.3/172.26.2.3) as the new master:

```
/usr/lib/postgresql/14/bin/pg_ctl -D /data/pgsql/14/data promote
```

4. On the old master (100.125.105.2/172.25.4.2), configure it as a standby server:

Create the standby signal file with the following command:

```
touch /data/pgsql/14/data/standby.signal
```

5. Edit the postgresql.auto.conf file on the old master (100.125.105.2):

Update the primary_conninfo parameter to connect to the new master (100.125.110.3):

```
primary_conninfo = 'user=postgres passfile='/var/lib/postgresql/.pgpass'
channel_binding=prefer host=172.25.4.2 port=6412 sslmode=prefer sslcompression=0
sslcertmode=allow sslsni=1 ssl_min_protocol_version=TLSv1.2 gssencmode=prefer
```

```
krbsrvname=postgres gssdelegation=0 target_session_attrs=any  
load_balance_hosts=disable'
```

6. Start the PostgreSQL service on the old master (100.125.105.2):

```
/usr/lib/postgresql/14/bin/pg_ctl -D /data/pgsql/14/data start
```

7. On the new master (100.125.110.3), verify that replication is working:

Run the following command to check replication status:

```
select * from pg_stat_replication;
```

8. On the current standby (100.125.105.2), check replication delay:

Run the following commands to check replication delay and WAL receiver status:

```
select now() - pg_last_xact_replay_timestamp() as replication_delay;
```

```
select * from pg_stat_wal_receiver;
```

Switchback Process

1. Stop the PostgreSQL service on the current master (100.125.110.3/172.26.2.3):

```
/usr/lib/postgresql/14/bin/pg_ctl -D /data/pgsql/14/data stop
```

2. Promote the old master (100.125.105.2/172.25.4.2) back to the master role:

```
/usr/lib/postgresql/14/bin/pg_ctl -D /data/pgsql/14/data promote
```

3. Remove the standby.signal file on the old master (100.125.105.2):

```
rm /data/pgsql/14/data/standby.signal
```

4. On the current master (100.125.110.3), configure it as a standby server:

Create the standby signal file with the following command:

```
touch /data/pgsql/14/data/standby.signal
```

5. Edit the postgresql.auto.conf file on the current master (100.125.110.3):

Update the primary_conninfo parameter to connect to the new master (100.125.105.2):

```
primary_conninfo = 'user=postgres passfile='/var/lib/postgresql/.pgpass'  
channel_binding=prefer host=172.25.4.2 port=6412 sslmode=prefer sslcompression=0  
sslcertmode=allow sslsni=1 ssl_min_protocol_version=TLSv1.2 gssencmode=prefer  
krbsrvname=postgres gssdelegation=0 target_session_attrs=any  
load_balance_hosts=disable'
```

6. Start the PostgreSQL service on the current master (100.125.110.3):

```
/usr/lib/postgresql/14/bin/pg_ctl -D /data/pgsql/14/data start
```

7. Verify that replication is working correctly on both the master and the standby:

On the master (100.125.105.2), run:

```
select * from pg_stat_replication;
```

On the standby (100.125.110.3), run:

```
select now() - pg_last_xact_replay_timestamp() as replication_delay;
```

```
select * from pg_stat_wal_receiver;
```