************ How Replication works in Postgresql

Streaming replication in PostgreSQL works on log shipping. Every transaction in postgres is written to a transaction log called WAL (write-ahead log) to achieve durability. A slave uses these WAL segments to continuously replicate changes from its master.
There exists three mandatory processes � wal sender , wal receiver and startup process, these play a major role in achieving streaming replication in postgres.
A wal sender process runs on a master, whereas the wal receiver and startup processes runs on its slave. When you start the replication, a wal receiver process sends the LSN (Log Sequence Number) up until when the WAL data has been replayed on a slave, to the master. And then the wal sender process on master sends the WAL data until the latest LSN starting from the LSN sent by the wal receiver, to the slave. Wal receiver writes the WAL data sent by wal sender to WAL segments. It is the startup process on slave that replays the data written to WAL segment. And then the streaming replication begins.
Note: Log Sequence Number, or LSN, is a pointer to a location in the WAL.

Replication Steps********

Master********

```
1: su - postgres

2: mkdir master
3: chmod 700 master
4: initdb -D master/
5: vi /var/lib/pgsql/master/postgresql.conf
listen_addresses='*'
wal_level='replica'
max_wal_senders=3
wal_keep_segments=32
hot_standby=on

6: vi /var/lib/pgsql/master/pg_hba.conf
```

******* under replication section-

```
host  replication postgres    192.168.1.2/32    trust

7: pg_ctl -D /var/lib/pgsql/master/ start
```

****** Slave Machine *****

```
1: su - postgres
2: mkdir standby
3: chmod 700 standby/
4:pg_basebackup -P -R -X stream -c fast -h 127.0.0.1   -U postgres -D
/var/lib/pgsql/12/standby
```

```
5: pg_ctl -D /var/lib/pgsql/standby/ start


*** Master
$ psql
select * from pg_stat_replication;


###### to prompte a slave as Master
go to slave and write following command:

./pg_ctl -D /var/lib/pgsql/10/standby/ promote



**** slave
select pg_is_in_recovery();
select pg_last_xact_replay_timestamp();
Check how far off is the Standby from Master.
pg_xlog_location_diff(pg_stat_replication.sent_location,
pg_stat_replication.replay_location)


Calculating lags in Seconds. The following is SQL, which most people uses to find
the lag in seconds:


SELECT CASE WHEN pg_last_xlog_receive_location() = pg_last_xlog_replay_location()
                 THEN 0
               ELSE EXTRACT (EPOCH FROM now() - pg_last_xact_replay_timestamp())
          END AS log_delay;
```