**PostgreSQL Replication-** The process of copying data from a PostgreSQL database server to another server is called PostgreSQL Replication. The source database server is usually called the Master server, whereas the database server receiving the copied data is called the Replica server.

**Automatic Failover in PostgreSQL-** If the primary server  for the database fails.  there are tools available that allow for Automatic Failover, which can help detect failures and automatically switch to the standby, minimizing database downtime. Enterprise DB’s EDB Postgres Failover Manager lets you automatically detect database failures and promotes the most current standby server as the new master, helping to avoid costly database downtime.

**High Availability and Failover Replication-** High Availability refers to database systems that are set up so that standby servers can take over quickly when the master or primary server fails. To achieve high availability, a database system should meet some key requirements it should have redundancy to prevent single points of failure, reliable switchover mechanisms, and active monitoring to detect any failures that may occur.

**Use PostgreSQL Replication**

* Replication of data can have many uses:
* OLTP Performance
* Data Migration
* Testing Systems in Parallel

**The Models of PostgreSQL Database Replication**

* Single-Master Replication (SMR), changes to table rows in a designated master database server are replicated to one or more replica servers.
* Multi-Master Replication (MMR), changes to table rows in more than one designated master database are replicated to their counterpart tables in every other master database.

**Classes of PostgreSQL Replication**

* Unidirectional Replication
* Bidirectional Replication

**Replication Modes in PostgreSQL Database**

* Asynchronous Mode of Replication
* Synchronous Mode of Replication

**Types of PostgreSQL Database Replication**

* Physical Replication of PostgreSQL Database
* Logical Replication of PostgreSQL Database

**Write-Ahead Logging (WAL) in PostgreSQL Database-** A standard mechanism to prevent data loss in case of circumstances like operating system crash, hardware failure, or PostgreSQL crash. This mechanism is called Write Ahead Logging (WAL), and the log file is called Write Ahead Log.

**Transaction Log and WAL Segment Files in PostgreSQL**

The PostgreSQL transaction log is a virtual file with a capacity of 8-byte length. Physically, the log is divided into 16-MB files, each of which is called a WAL segment.

**WAL Writer in PostgreSQL**- It is a background process to check WAL buffer. It avoids burst of IO activity & extends the process over time.

**WAL Segment File Management**- They are stored in the pg\_wal subdirectory. PostgreSQL switches to a new WAL segment file under the following conditions:

* The WAL segment has been filled up.
* The function pg\_switch\_wal has been issued.

archive\_mode is enabled and the time set to archive\_timeout has been exceeded.

After switching, the files can be re used again.

**Replication Options Based on WAL in PostgreSQL**

* Continuous WAL Archiving- The script can use the scp command to copy the file to one or more locations (NFS mount) for recovery.
* Log Shipping Based Replication (File Level)- Copying log files to another PostgreSQL server for the purpose of creating another standby server by replaying WAL files.
* Log Shipping Based Replication (Block Level)- Streaming replication improves the log shipping process. Instead of waiting for the WAL switch, the records are sent as they are generated, thus reducing replication delay.

**Time required for primary server to retain WAL segment files**- the server can discard/recycle the WAL segment file once the archive script reports success, if the clients are active, an error will occur.

**Log Shipping Based Replication**- Physically Active Replication in PostgreSQL database is the data replication based on WAL. In active replication , the standby server connects to the primary server and receives WAL records using a replication protocol.

**WAL Sender and WAL Receiver in PostgreSQL**

* WAL receiver-uses the connection details provided in the primary\_conninfo parameter of recovery.conf and connects to the primary server using a TCP/IP connection.
* WAL sender - saves the WAL records in WAL as if they were generated by client activity of locally connected clients. Once the WAL records reach the WAL segment files the standby server constantly keeps replaying the WAL so that standby and primary are up to date.

**PostgreSQL Replication and Failover Over Setup**

Step 1: Disable and stop firewall on both the machines:

Step 2: On the primary server, allow replication connections and connections from the same network. Modify pg\_hba.conf

Step 3: On the primary server, edit postgresql.conf to modify the parameters

Step 4: Start the primary server

Step 5: Take base backup to bootstrap the standby server

Step 6: Check the base backup label file

Step 7: In the base backup, add the line in the recovery.conf

Step 8: Check the /tmp/sb\_data/recovery.conf file

Step 9: Connect to the primary server and issue this command

Step 10: Transfer the base backup to the standby server

Step 11: Start the standby server

Step 12: Connect to the primary server and issue some simple commands

Step 13: Check the data on the replica

**PostgreSQL Manual Failover Steps**

Step 1: Crash the primary server.

Step 2: Promote the standby server by running the following command on the standby server

Step 3: Connect to the promoted standby server and insert a row

**Automate Failover and Replication in PostgreSQL**

EDB Postgres Failover Manager (EFM) continuously monitors system health and sends email alerts based on system events. During failure,it switches to recent standby

**repmgr for PostgreSQL.**

repmgr (Replication Manager) to manage replication & failover for postgresql