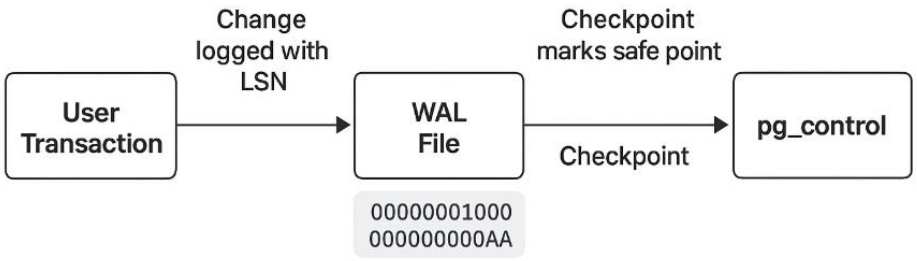
? Complete WAL Flow in PostgreSQL



1? A Change Happens

- A user executes a query that modifies data (INSERT, UPDATE, DELETE).
- PostgreSQL first writes the change to WAL (Write-Ahead Log) before changing the actual data files.
- **Purpose:** guarantee durability — even if the system crashes, WAL can replay the change.
- **? WAL directory path:**

$PGDATA/pg_wal/

2? WAL Write (LSN Assigned):-

- Each record written to WAL gets a unique LSN (Log Sequence Number).
- The LSN points to the exact byte position in the WAL stream.
- LSN format: X/Y (file number / byte offset).

? Think:
LSN = exact location of a change inside a WAL file.

3? WAL Segment (File)

- WAL records are stored sequentially in WAL segment files (default size = 16 MB).
- Each file has a unique name like:

000000010000000300000000A 00000001 → Timeline ID 00000003 → Log (WAL) file number high

0000000A → Log file number low

Location:

$PGDATA/pg_wal/

4 Timeline ID

- Identifies the history or version of the database.
- Incremented when a failover or recovery creates a new “branch” of WAL.
- Ensures replicas use the correct WAL history.

5 Checkpoint

- A checkpoint is triggered (automatically or manually).
- PostgreSQL writes a checkpoint record inside the current WAL file at the current LSN.
- Then, all dirty data pages in memory are flushed to disk (data files in $PGDATA/base/).

Purpose:

- Marks a “safe point” — all WAL changes before this LSN are already on disk.

6 pg_control Update

- After the checkpoint record is written, PostgreSQL updates pg_control with:

1. Checkpoint LSN
2. Redo LSN
3. Timeline ID
4. Corresponding WAL file name

File path:

$PGDATA/global/pg_control

7 Crash or Restart Recovery

- When PostgreSQL starts:

1. It reads pg_control to find:

- Last checkpoint LSN
- Redo LSN
- Timeline ID

2. It finds the exact WAL file from pg_wal/ containing that LSN.

3. It replays WAL records from the redo LSN until all changes are consistent.

Flow Summary

Step	Component	Description	Path
1	User Transaction	Change happens	—
2	WAL Record	Change logged with LSN	$PGDATA/pg_wal/
3	WAL File	Records written sequentially	$PGDATA/pg_wal/
4	Checkpoint	Special record added to WAL	$PGDATA/pg_wal/
5	pg_control	Saves checkpoint LSN and WAL info	$PGDATA/global/pg_control
6	Recovery	PostgreSQL reads pg_control → replays WAL	$PGDATA/pg_wal/

✅ In one line:

> Transaction → WAL (with LSN) → Checkpoint (marks safe point) → pg_control (stores checkpoint LSN) → Recovery uses these to restart safely.

Example:-

🔍 Example: Complete WAL Flow

1️⃣ User Transaction

- A user runs this SQL:

```
INSERT INTO sales VALUES (1, 'Laptop', 80000);
```

- PostgreSQL does not immediately write this to the main table file.
- Instead, it first creates a record in the WAL so that if the system crashes, the change can be replayed.

2️⃣ WAL Record (with LSN)

- The change gets logged in the current WAL file — for example:

000000010000000030000000A

- And PostgreSQL assigns a Log Sequence Number (LSN) to this record:

3/A000000

- This LSN means:

3 → WAL segment (file number 3)

A000000 → Byte offset inside that file (the exact place where this change is written)

3 WAL File in pg_wal

- This WAL file (000000010000000030000000A) lives inside:

/var/lib/pgsql/15/data/pg_wal/

- As more transactions happen, WAL keeps getting new records, and LSN keeps increasing sequentially.

4 Checkpoint

- After some time (based on settings like checkpoint_timeout or max_wal_size), PostgreSQL triggers a checkpoint.
- When the checkpoint occurs:

1. It writes a special checkpoint record inside the same current WAL file.
2. Let's say the checkpoint record is at LSN = 3/A800000.

- This means:

"All changes up to LSN 3/A800000 are safely written to the main data files."

5 pg_control Update

- Once the checkpoint record is written, PostgreSQL updates the file:

/var/lib/pgsql/15/data/global/pg_control

- Inside this file, PostgreSQL stores:

Checkpoint LSN: 3/A800000

Redo LSN: Starting point for recovery

Timeline ID: 1

WAL file name: 000000010000000030000000A

- This tells PostgreSQL where to start replaying WAL if the database restarts.

6️⃣ Crash or Restart (Recovery)

- If PostgreSQL crashes or restarts:
- On startup, it reads pg_control.
- Finds the checkpoint LSN = 3/A800000.
- Finds the WAL file 000000010000000030000000A in pg_wal/.
- Replays WAL records after the checkpoint LSN (if any) to restore full consistency.

✅ Summary in Words

Step	What Happens	Example
1	Transaction occurs	INSERT INTO sales ...
2	WAL record written with LSN	LSN 3/A000000
3	WAL stored in pg_wal/	000000010000000030000000A
4	Checkpoint written in same WAL file	Checkpoint LSN 3/A800000
5	pg_control updated	Stores LSN & WAL file info
6	On crash, PostgreSQL reads pg_control	Starts recovery from checkpoint LSN