Difference between wal file & archival file

Difference Between WAL File & Archival File

Feature	WAL File (pg_wal)	Archival File (Archived WAL)
Location	$PGDATA/pg_wal/	Archive directory (configured via archive_command)
Purpose	Stores recent WAL segments for crash recovery & replication	Used for Point-in-Time Recovery (PITR) and backup retention
Lifespan	Automatically removed after checkpoint_completion_target or max_wal_size is reached	Retained based on archive retention policy
Deletion	PostgreSQL removes old WAL files automatically	Retained in archive until manually cleaned up
Use Case	Replication, crash recovery, streaming replication	PITR, backup restoration, historical recovery

---

Which Files Are Used for Replication, PITR, and Instance Recovery?

Scenario	File Used
Streaming Replication (Primary to Standby)	WAL files (pg_wal)
Logical Replication	WAL records but decoded into logical changes
Point-in-Time Recovery (PITR)	Archived WAL files (archive_command)
Crash Recovery (Instance Restart)	WAL files (pg_wal)
Disaster Recovery (Full Restore)	Base backup + Archived WAL files

---

Detailed Explanation of Usage:

1. Replication (Streaming & Logical)

1. Uses WAL files in pg_wal/ directory.
2. Standby server fetches WAL from pg_wal or uses restore_command for missing WAL.
3. Logical replication decodes WAL to SQL statements.

2. Point-in-Time Recovery (PITR)

1. Uses Archived WAL Files from the archive directory.
2. Restores a base backup and replays archived WAL files until a
3. specific timestamp or LSN.

4. Instance Recovery (Crash Recovery)

1. Uses WAL files in pg_wal/.
2. PostgreSQL replays WAL from the last checkpoint to bring the database to a consistent state.

Summary
Replication uses WAL files (pg_wal/).
PITR uses archived WAL files from the archive directory.
Instance recovery uses WAL files (pg_wal/) from the latest checkpoint.
Let me know if you need more details!

Relationship Between WAL Files and Archive Files

Both WAL files and archive files contain the same transaction log data, but they serve different roles and are stored in different locations:

WAL Files (pg_wal):
These files are actively written by PostgreSQL to record every change made to the database. They reside in the `pg_wal` directory and are critical for crash

recovery and replication. As new transactions occur, WAL files grow until they're full, at which point PostgreSQL switches to a new WAL file.

Archive Files:
Once a WAL file is complete (i.e., it is no longer actively written to), PostgreSQL can archive it. Archiving involves copying the finished WAL file from the `pg_wal` directory to a designated archive location. This process is controlled by the `archive_command` parameter in the PostgreSQL configuration. Archived WAL files are primarily used for **Point-in-Time Recovery (PITR)** and long-term backup, ensuring that you can restore the database to a specific moment even if the current WAL files are no longer available.

How Archive Files Are Generated: Step-by-Step Example

Enable Archiving in PostgreSQL Configuration:

In your `postgresql.conf` file, you enable WAL archiving by setting

```
archive_mode = on
archive_command = 'cp %p /mnt/wal_archive/%f'
```

- `archive_mode = on` turns on the archiving process.
- `archive_command` specifies the shell command to execute when a WAL file is ready to be archived.
  - `%p` is replaced with the full path of the WAL file in the `pg_wal` directory.
  - `%f` is replaced with the WAL file's name.

WAL File Completion:

As the database processes transactions, it writes to WAL files in the `pg_wal` directory. For example, assume PostgreSQL writes a WAL file named:

```
000000010000000000000016
```

When this file reaches its size limit (typically 16MB), it is considered complete.

Archiving Triggered:

Once the WAL file is complete, PostgreSQL automatically executes the `archive_command`. Using our example:

```
cp /var/lib/postgresql/data/pg_wal/000000010000000000000016
/mnt/wal_archive/000000010000000000000016
```

This command copies the finished WAL file from the active `pg_wal` directory to the archive directory `/mnt/wal_archive/`.

Usage of Archived WAL Files:

- **Point-in-Time Recovery (PITR):**
  When you need to restore your database to a specific point in time, you restore a base backup and then replay the archived WAL files. These archived files ensure that all the transactions that occurred after the backup are applied.
- **Disaster Recovery:**
  They provide a historical record of all transactions, which can be invaluable if you need to perform a full database recovery.

Summary

WAL Files in `pg_wal`:
Used for **crash recovery** and **streaming replication**; actively written and maintained by PostgreSQL.

Archived WAL Files:
Created by copying complete WAL files from `pg_wal` to an archive location using the `archive_command`. These are used for **PITR** and **long-term backup/recovery**.