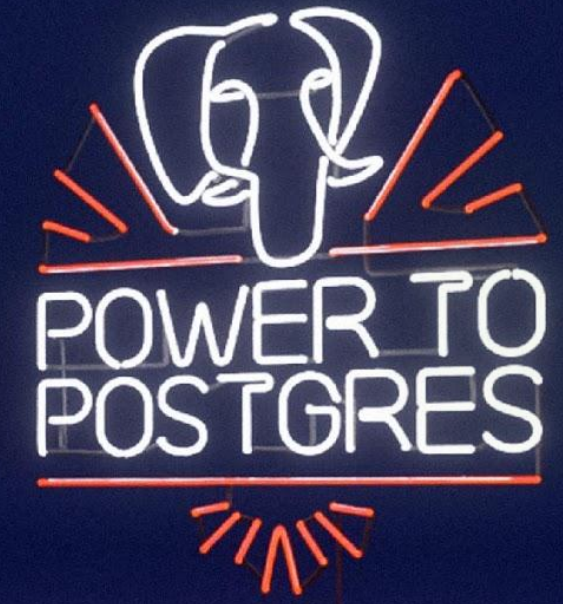
How to properly backup your data



Lætitia Avrot

March 2021 - Version 1.0

EDB

Who am I ?

Lætitia Avrot

- Senior Database Consultant at EDB
- Postgres Contributor
- PostgreSQL Europe treasurer

Agenda

- Data loss
- Logical exports
- Standbys
- WALs and Recovery
- VM/Storage snapshots
- Physical backup
- Conclusion

Data loss

EDB

If the Death Star shows up and does to Earth what it did to Alderaan, practically everybody is going to lose data.

Robert Haas, VP, Chief Database Scientist at EDB, 2019/09/23

About zero data loss...

The story of a unicorn hunt

- **Not possible**
- **Let's try another approach**
    - Can you recreate lost data/transactions?
    - How much costs 1 second/1 minute/1 hour/1 day/1 week of data loss?
    - How much time can the application be down?
    - How much time can you afford to recover?

RTO/RPO

Defining constraints

Recovery Time Objective

"It is the **targeted duration** of time and a service level within which a business process must be **restored** after a **disaster** (or **disruption**) in order to avoid unacceptable consequences associated with a break in business continuity."

Recovery Point Objective

"It is defined by **business continuity planning**. It is the maximum **targeted period** in which data (transactions) might be **lost** from an IT service due to a major incident."

Quotes from Wikipedia

Scenarios of data loss

- Data corruption
- Host failing
- Network failing
- Any massive disaster (Fire, Earthquake, Nuclear bomb, tsunami, tornado…)
- Human destroying data (deliberately or by mistake)

Logical exports

What is a logical export?

- It is an export of the data in a format that can be portable. It's a snapshot of your data.
- Examples:
    - csv files
    - SQL files
- The Postgres core recommended tools are `pg_dump` and `pg_dumpall`

When to rely on logical export only?

All the following conditions needs to be fulfilled

- Losing data between your export time and the stopping point is not a problem
- Having downtime during restore and post-restore operations (`vacuum full analyze`) is not a problem
- You solemnly swear you'll test the restore and post-restore operations frequently

When to rely on logical export only?

Example:

- You export all your data every day at 3 AM. Import time takes 2 hours. Post-restore operations take 1 hour to complete. An incident occurs at 3 PM.
- How many transactions will be lost ? When will the database be available again ?

➔ All transactions between **3 AM** and **3 PM** will be lost.

➔ The database should be available again around **6 PM**. (3 hours of downtime)

What do I need in order to rely on logical export only ?

- Check regularly that data loss stays within acceptable bounds
- Try regularly to import the generated export to check that:
  - it's not corrupted;
  - the restore duration is still ok;
  - the post-restore operations duration is still ok.

How to use pg_dump/pg_restore

Exporting

- A whole database

```
pg_dump -h <host> -p <port> -U <user> <dbname>
```

- Plain SQL/Custom format

```
pg_dump <connection options> -Fp/-Fc > export.sql/.dmp
```

- A single table/schema

```
pg_dump <connection options> -t/-n <name>
```

- DDL only

```
pg_dump <connection options> -s
```

Importing

- Plain SQL

```
psql <connection options> -f export.sql
```

- Custom format

```
pg_restore <connection options> export.dmp
pg_restore export.dmp > export.sql
```

- A single table/schema

```
pg_restore <connection options> -t/-n <name> export.dmp
```

- DDL only

```
pg_restore <connection options> -s
```

Standby

What is a standby?

Also called physical replica, princess/worker, secondary, tertiary...

- Another instance identical to your queen node
    - Physical standby
    - Logical standby
- In recovery forever (applying WAL files)
- Open to read-only queries or not

When to rely on standby only ?

All of the following conditions need to be fulfilled

✓ **Losing** some or **all** of your data is not a problem

Example:

- A former employee connects to the database and drops it.
- A hacker enters your system and drops the database.
- The DBA removes the PGDATA directory on the primary by mistake while trying to rebuild the secondary

How much data will you lose?

→ **Everything**

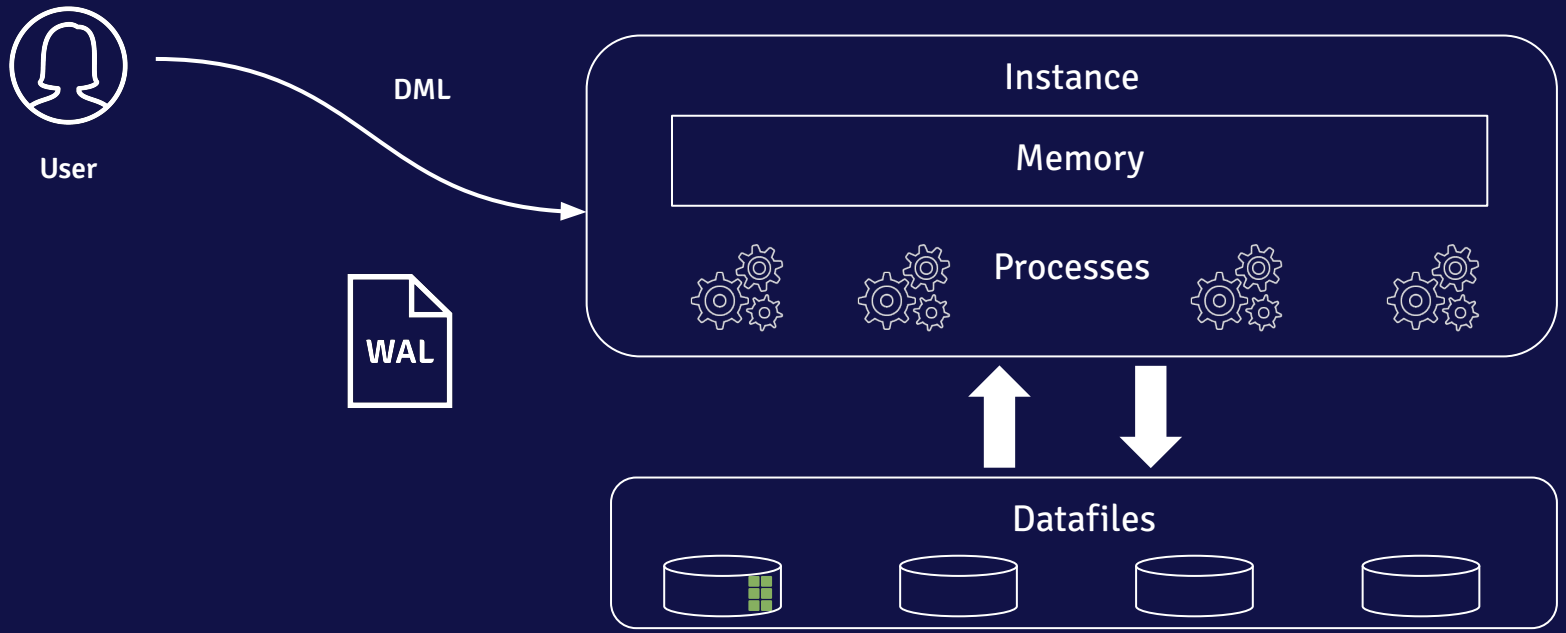
How to set up a standby?

High level steps

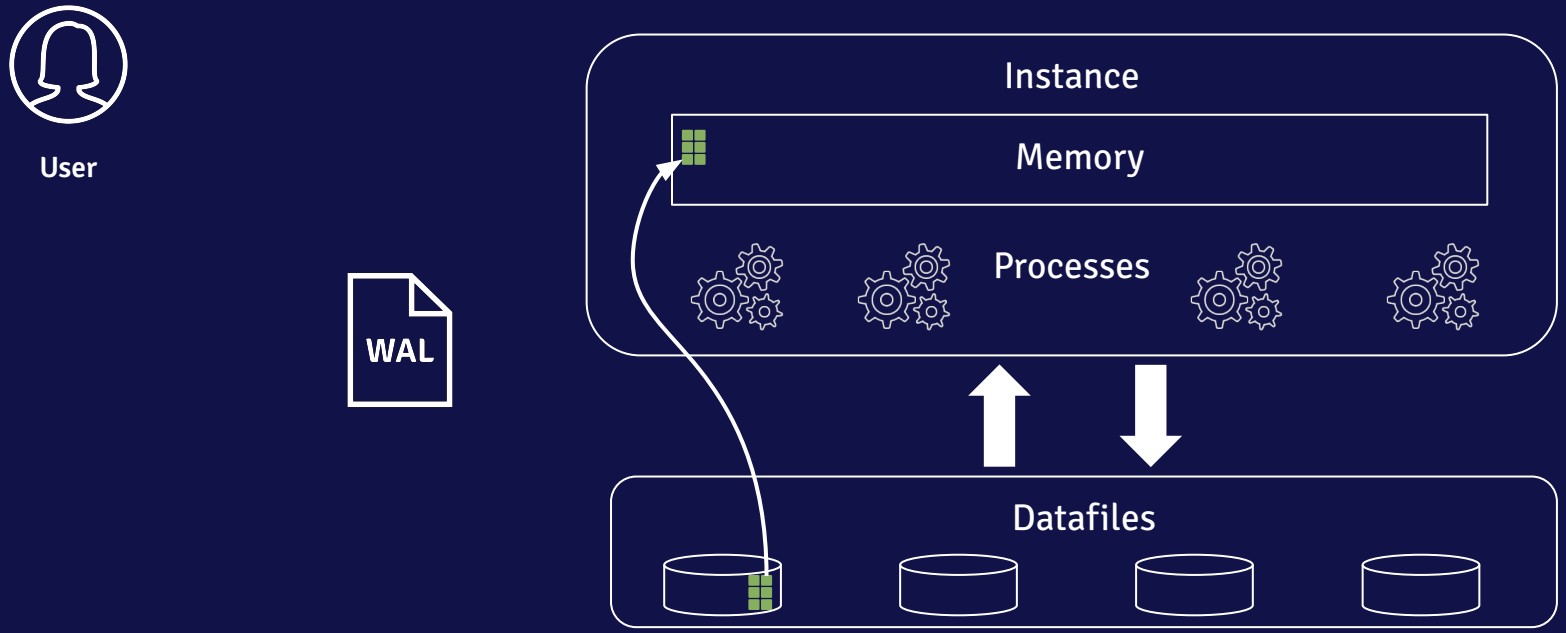
- Make the queen node ready for replication
- Take a physical backup of the queen node
- Restore the physical backup on the princess/worker node
- Change some settings so the princess/worker node knows it’s not a queen
- Start the princess/worker node
- Check that replication is working fine

WALs and Recovery

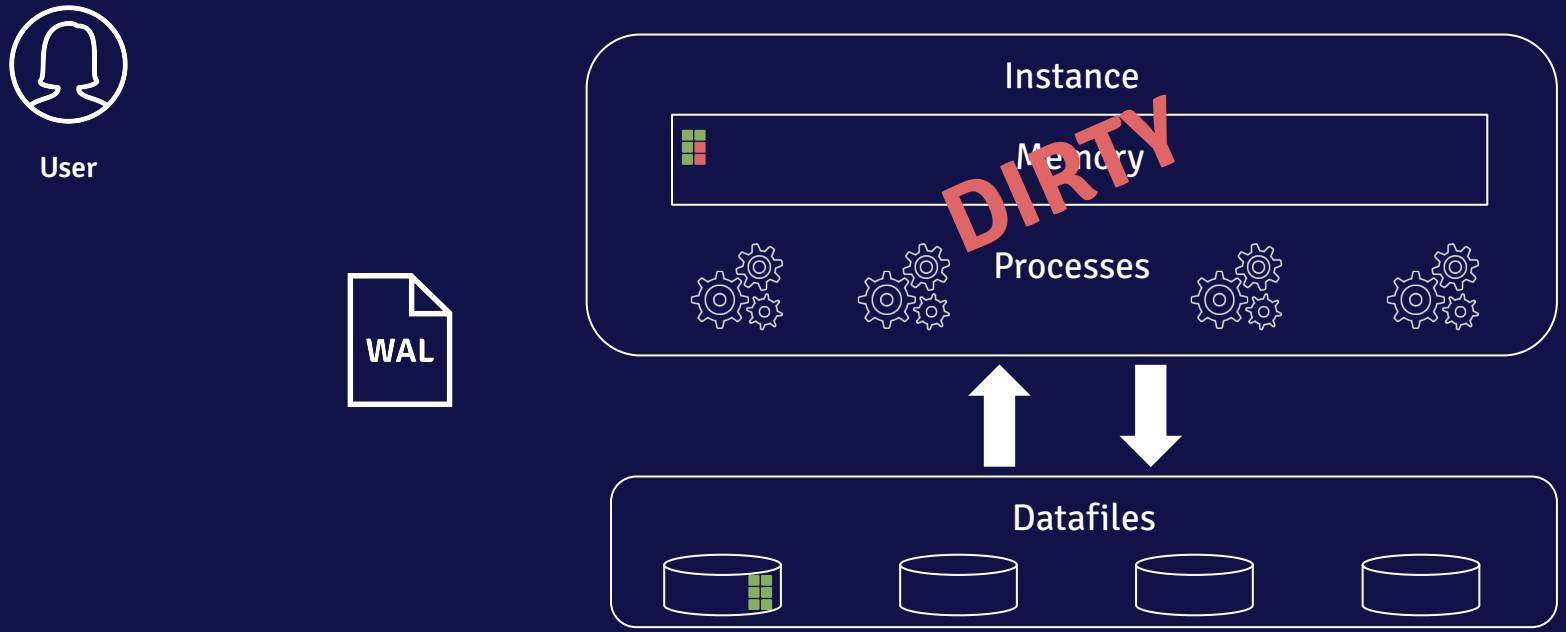
WALs on a normal day



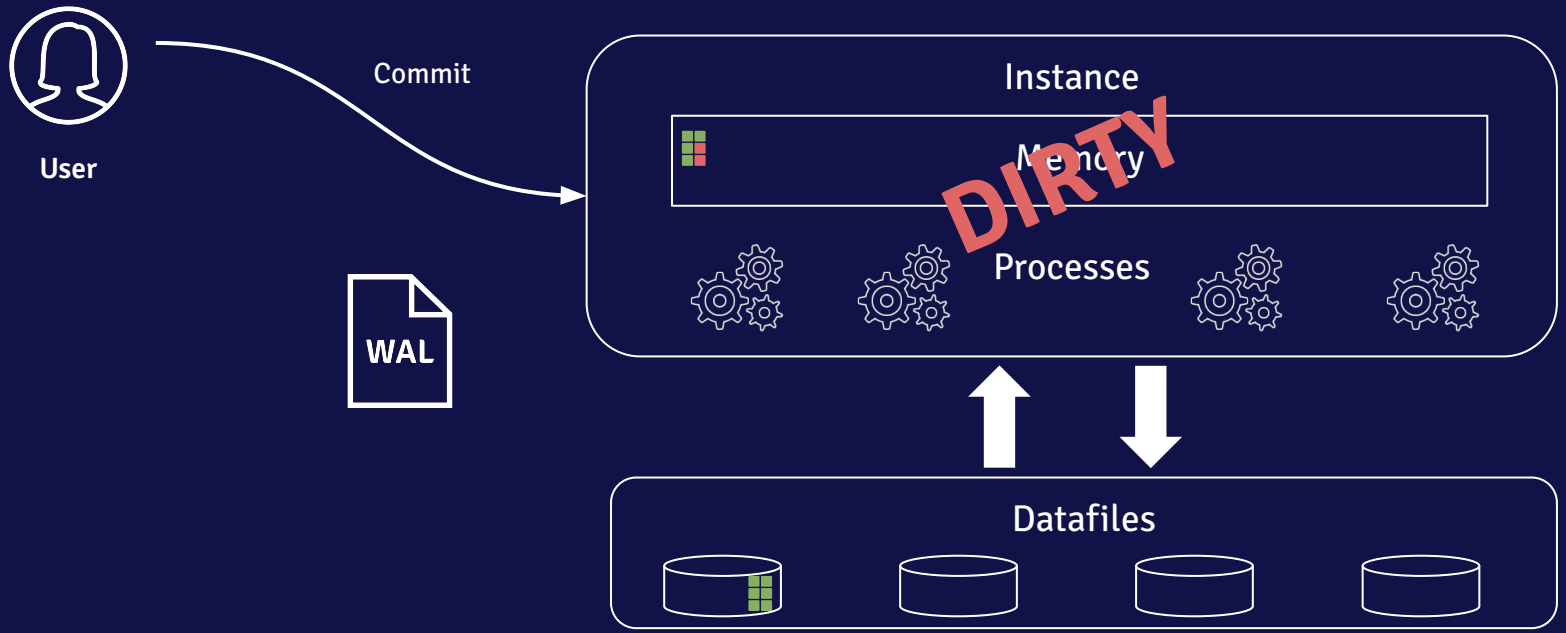
WALs on a normal day



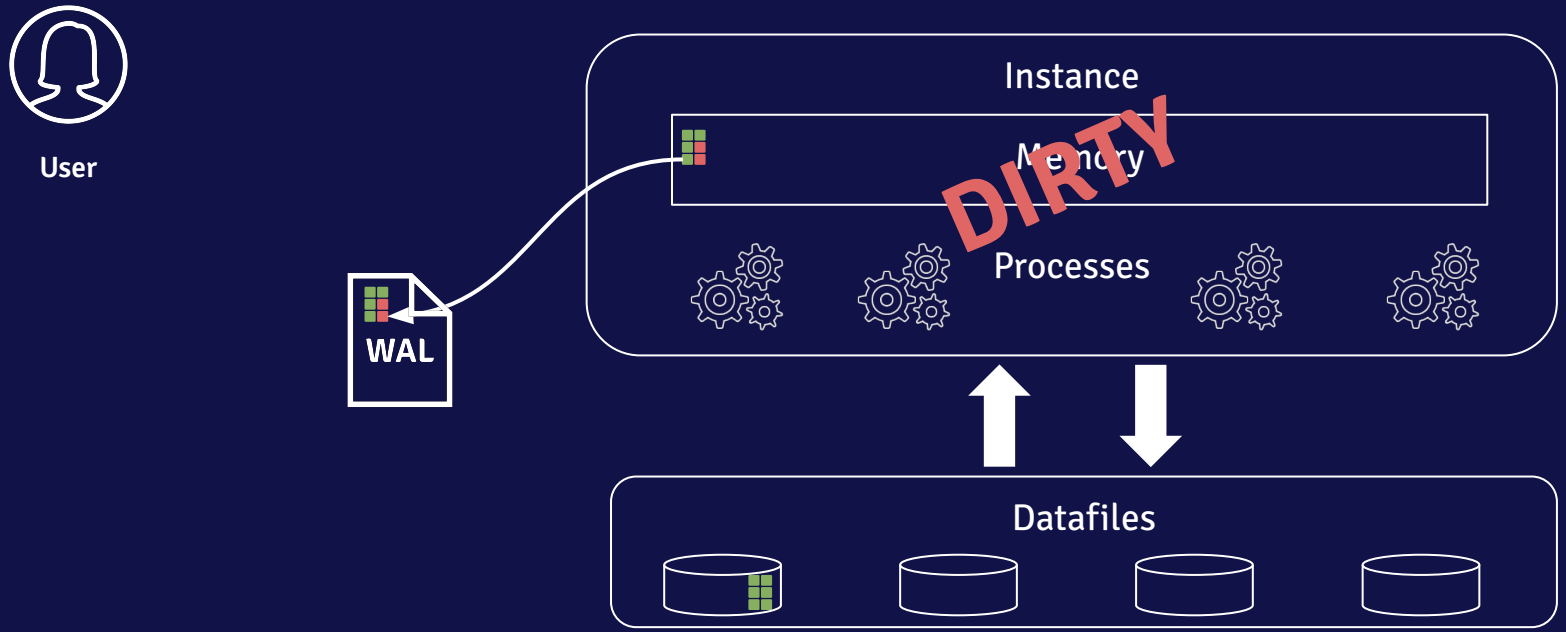
WALs on a normal day



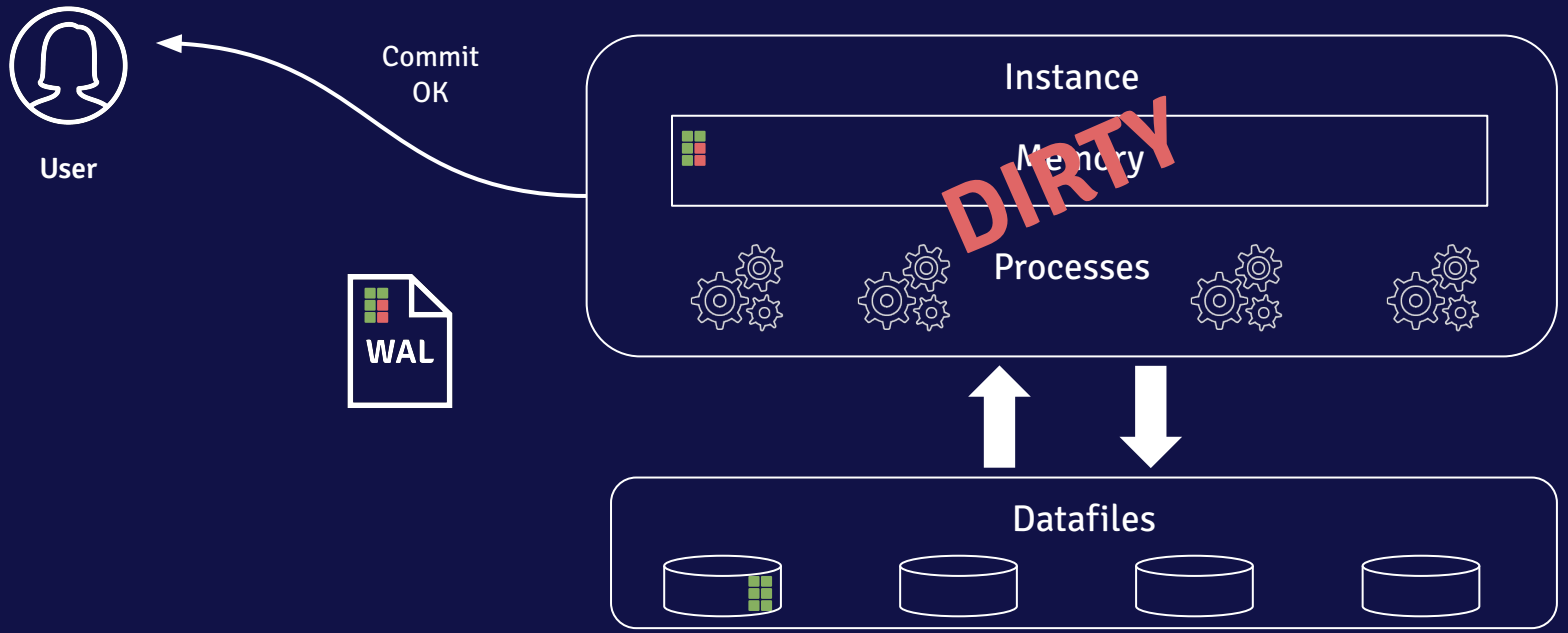
WALs on a normal day



WALs on a normal day

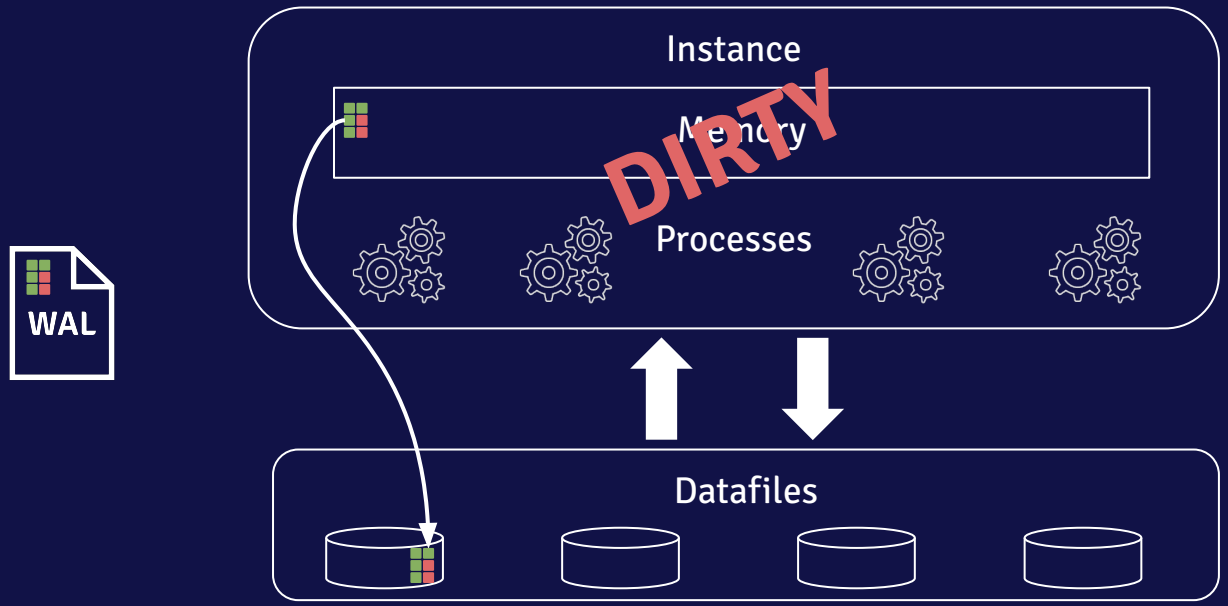


WALs on a normal day

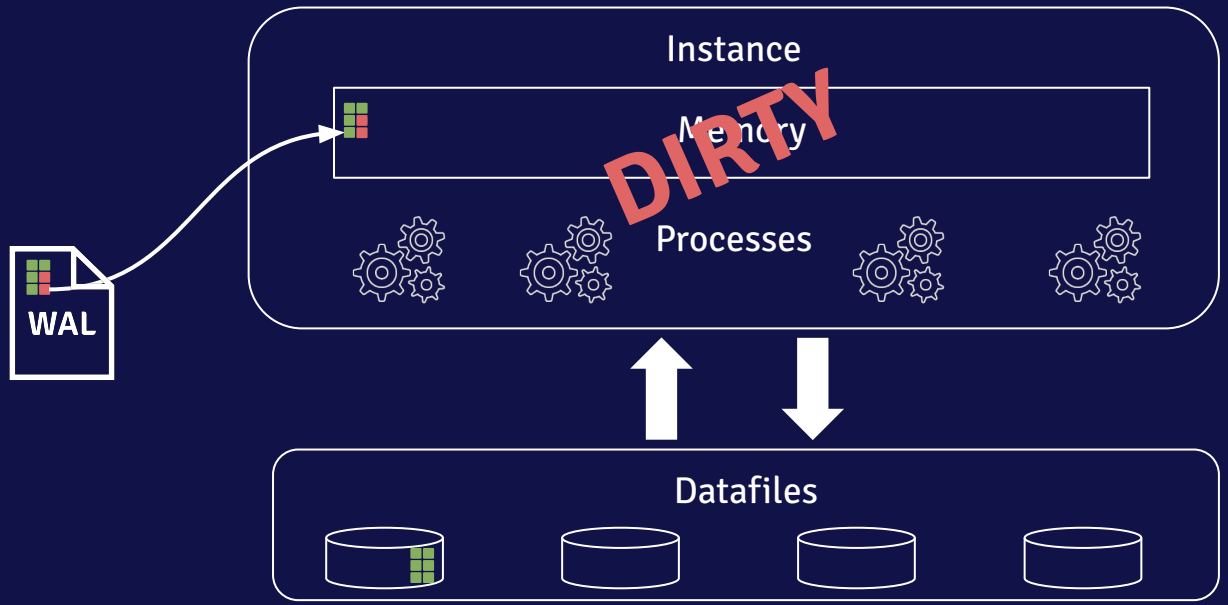


WALs on a normal day

Checkpoint

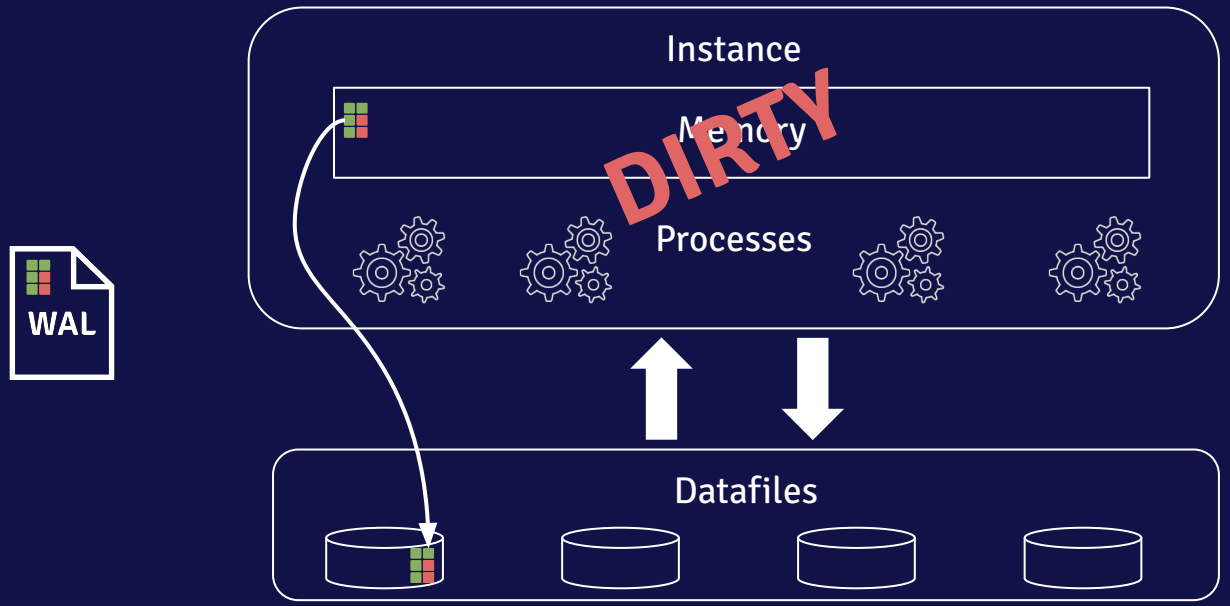


Recovery after a crash



Recovery after a crash

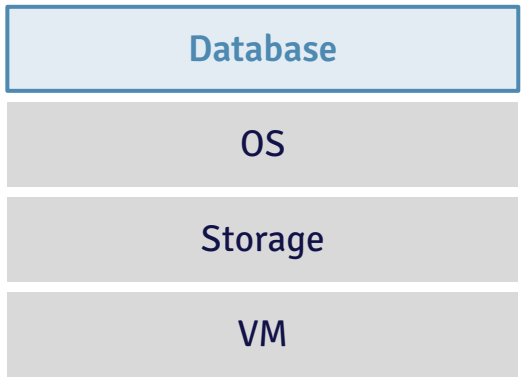
Checkpoint



VM or storage snapshots

What is a VM/storage snapshot ?

- Set of files at one point in time
  - Offline snapshot
  - Online snapshots
- It operates on layer **below** the database



An RDBMS can at one point have **inconsistent files** on disc. You need to remember to put your database in **backup mode** to not rely on luck for your recovery and to **archive the WAL files**.

When to rely on VM/Storage snapshots only ?

All the following conditions needs to be fulfilled

- You really understand what you're doing
- You don't want another option
- You solemnly swear you'll test the restore and recovery operations frequently
- You'll also archive the WAL files between the beginning of a snapshot and now

How to perform restorable VM/Storage snapshots ?

- Taking the snapshot offline
- Taking the snapshot online
  - Use `pg_start_backup('label', false, false)` beforehand
  - Use `pg_stop_backup(false, true)` afterwards
  - Don't forget the `backup_label` and the `tablespace_map` files

When the database is in backup mode, you might generate more WAL files

Physical backups

What is a physical backup?

A set of consistent or inconsistent files that will allow recreating the cluster from nothing

- Offline physical backups
- Online physical backups
- Several tools
    - pg_basebackup
    - Barman
    - pgBackRest
    - ...

Creating a backup that the database **can recover from** is complex.
Don't use your own scripts!

When to rely on physical backup only ?

All the following conditions needs to be fulfilled

- You can afford the additional backup storage
- You can afford the restore and recovery time
- You solemnly swear you'll test the restore and recovery operations frequently
- You'll also archive all of the WAL files between the earliest backup and now

Comparing solutions

Let’s compare

	Logical exports	Standbys	VM/Storage snapshots	Physical backups
✓	• Portable • Smaller granularity	• Quickly available	• Quickly available • Reduced data loss	• Safe tools available • Reduced data loss
✗	• Slow to restore • Huge Data loss	• Whole cluster only • Huge Data loss (possibly)	• Whole cluster only • No WAL management • Still needs to recover • Can go very wrong	• Whole cluster only • Still needs to recover

Questions?