

# You forgot to put the WHERE in the DELETE?

Martín Marqués

2ndQuadrant

October 18, 2019

2ndQuadrant<sup>®</sup>   
PostgreSQL

# Content

- 1 Introduction
- 2 What are backups for
- 3 Backup types
- 4 Backups considerations
- 5 Recovering
- 6 Finally
- 7 Question

# What are we going to talk about

- What backups are for
- Different ways to backup postgres databases
- Disaster scenarios
- Recovery options
- Final notes

# Content

- 1 Introduction
- 2 What are backups for**
- 3 Backup types
- 4 Backups considerations
- 5 Recovering
- 6 Finally
- 7 Question

# What are backups for?

- Recover from disasters
- Spin-up replicas
- Create development environments
- Archive can serve as a WAL hub

# Types of disasters to recover from

- Server irrecoverable
- Table gone
- Data gone
- Corruption

# Types of disasters to recover from

- Server irrecoverable → DC catches fire/gets flooded
- Table gone → Devops drops/truncates a table
- Data gone → Yesterdays deploy has a bug
- Corruption → HW corrupts data

# Content

- 1 Introduction
- 2 What are backups for
- 3 Backup types**
- 4 Backups considerations
- 5 Recovering
- 6 Finally
- 7 Question



# Backup types

- Logical backups
  - pg\_dump
  - pglogical
  - in-core logical replication

# Backup types

- Logical backups
  - pg\_dump
  - pglogical
  - in-core logical replication
- Physical backups
  - Basebackup + WAL files
    - External OS apps: rsync, tar, etc.
    - pg\_basebackup
    - Specific postgres tools
  - Standby: sync, async, delayed

# Logical Backups

- Use `pg_dump` for taking backups
- Use `pg_restore` to restore the backup

# Logical Backups

- Pros
  - Extremely robust
  - Backups are forward compatible
  - Partial recovery is an option
  - Flexible backup and recovery options

# Logical Backups

- Pros
  - Extremely robust
  - Backups are forward compatible
  - Partial recovery is an option
  - Flexible backup and recovery options
- Cons
  - Holds a snapshot for the whole time `pg_dump` runs
  - Recovers to snapshot taken, no PITR

# Logical Backups

```
pg_dumpall -g | psql -h remote_host -p 5433
SET
SET
SET
CREATE ROLE
ALTER ROLE
CREATE ROLE
ALTER ROLE
ERROR:  role "postgres" already exists
ALTER ROLE
```

# Logical Backups

```
$ pg_dump -Fc martin -f martin.dmp -Z 7

$ ls -l martin.dmp
-rw-r--r--. 1 postgres postgres 5207257 oct 17 11:14 martin.dmp

$ pg_restore -C -d postgres -j 4 -p 5433 martin.dmp
```

# Physical backups

- Using OS/Storage tools
  - Configure WAL archiving
  - Starts with `pg_start_backup()`
  - Do a filesystem backup with: `rsync`, `tar`, storage snapshot, etc.
  - Ends with `pg_stop_backup()`



# Physical Backups

- Using OS/Storage tools
  - Pros
    - Doesn't hold a postgres snapshot
    - Point in time recovery
    - Incremental backups are possible
    - Versatility of compression options
    - Possible to do work in parallel
  - Cons
    - Need to restore the whole cluster
    - Requires manual scripting or tool

# Physical backups

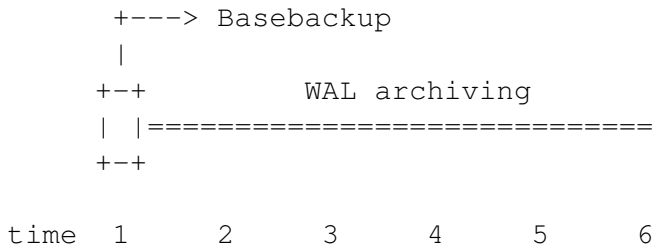
- Using `pg_basebackup`

```
$ pg_basebackup -h hoth -p 5432 -X stream -R -c fast -P \  
-D ~/11/data  
735619/735619 kB (100%), 1/1 tablespace
```

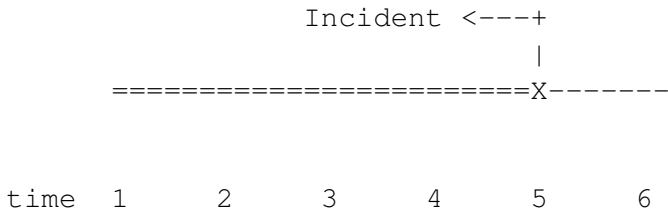
# Physical Backups

- Using `pg_basebackup`
  - Pros
    - Doesn't hold a postgres snapshot
    - Can do point in time recovery
    - Doesn't require operating system access to server
  - Cons
    - Need to restore the whole cluster
    - Can't do incremental backups
    - Backup work can not be done in parallel (WIP patch)
    - Doesn't backup configuration files outside PGDATA

# How does it work



# How does it work





# Standby: sync, async, delayed

- Create a replica from a physical backup
- Configure `primary_conninfo`,  
`recovery_min_apply_delay`, `synchronous_commit` and  
`synchronous_standby_names`

# Content

- 1 Introduction
- 2 What are backups for
- 3 Backup types
- 4 Backups considerations**
- 5 Recovering
- 6 Finally
- 7 Question



# Conditions a backup needs

- Finish successfully

# Conditions a backup needs

- Finish successfully
- Located in a remote location

# Conditions a backup needs

- Finish successfully
- Located in a remote location
- Backup and restore process is documented/automated

# Conditions a backup needs

- Finish successfully
- Located in a remote location
- Backup and restore process is documented/automated
- A restore was attempted successfully

# Schrodinger's Backup

“The condition of any backup is unknown until a restore is attempted.”

@nixcraft

# Content

- 1 Introduction
- 2 What are backups for
- 3 Backup types
- 4 Backups considerations
- 5 Recovering**
- 6 Finally
- 7 Question

# Types of disasters to recover from

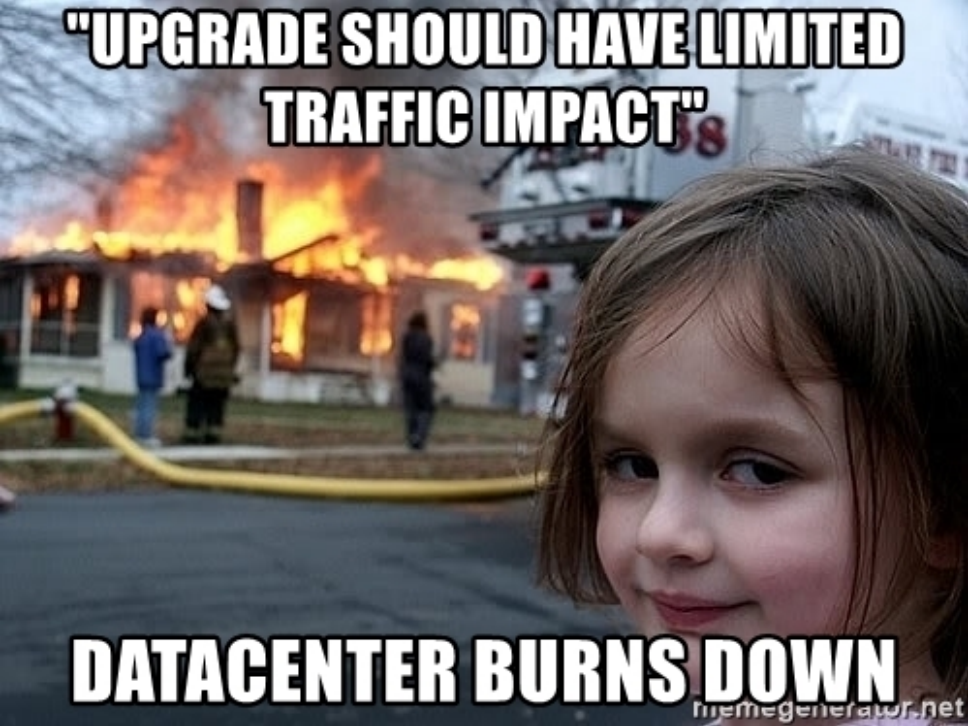
- Server irrecoverable
- Table gone / Data gone
- Corruption

# Server gone

## Server irrecoverable



**"UPGRADE SHOULD HAVE LIMITED  
TRAFFIC IMPACT"**



**DATACENTER BURNS DOWN**

# Server gone - Option 1

- Pick the standby with highest LSN and promote it
- Re-configure other standbys to connect to new primary
- Recover, if possible, the old primary as a new standby

# Server gone - Option 2

- Bring up a new server
- Recover latest base backup
- Replay all available archived WALs

# Server gone

## Table gone / Data gone

# Table/Data gone - Option 1, delayed standby

- Check if the data is still in the delayed standby
- If it is, pause replay
- Extract the missing data from this standby
- Resume replay
- Apply/merge recovered data on primary node

# Table/Data gone - Option 2, delayed standby

- Check if the data is still in the delayed standby
- If it is, pause replay
- Set `recovery_target_time` to a time before the incident
- Set `recovery_target_action` to promote
- Resume replay
- Provides less downtime
- Simpler
- There could be more data loss

# Table/Data gone - Option 3, using logical backup

- Extract missing data from a `pg_dump` backup
- Apply/merge recovered data on primary node
- There may be data loss

# Table/Data gone - Option 4

- Recover a physical base backup on new server
- Use `recovery_target_time` to recover up to before the disaster
  - Extract missing data and apply on primary node
  - Promote the recovered backup as the new primary



# Server gone

## Data corruption

# Data Corruption, FS corruption, missing files

- Almost always only option is a physical backup recovery
- Recover base backup
- Replay WALs to a consistent state, or an LSN at which the missing files are available
- Verify if a good version of the corrupt or missing file is present
- Keep replaying WALs if needed and repeat previous step

# Content

- 1 Introduction
- 2 What are backups for
- 3 Backup types
- 4 Backups considerations
- 5 Recovering
- 6 Finally**
- 7 Question



# Conclusions

# Conclusions

- Take backups!

# Conclusions

- Take backups!
- Use some automated tool

# Conclusions

- Take backups!
- Use some automated tool
- Standbys can serve as backups

# Conclusions

- Take backups!
- Use some automated tool
- Standbys can serve as backups
- Always test your backs by recovering



# Conclusions

- Take backups!
- Use some automated tool
- Standbys can serve as backups
- Always test your backs by recovering
- A delayed standby could provide lower RTO in case of disaster

# Conclusions

- Take backups!
- Use some automated tool
- Standbys can serve as backups
- Always test your backs by recovering
- A delayed standby could provide lower RTO in case of disaster
- You can achieve RPO zero by streaming WALs instead of using `archive_command`

# Conclusions

- Take backups!
- Use some automated tool
- Standbys can serve as backups
- Always test your backs by recovering
- A delayed standby could provide lower RTO in case of disaster
- You can achieve RPO zero by streaming WALs instead of using `archive_command`
- Security first: encrypt your backups

# Content

- 1 Introduction
- 2 What are backups for
- 3 Backup types
- 4 Backups considerations
- 5 Recovering
- 6 Finally
- 7 Question

