PERCONA LIVE 2023

# I Dropped My Database! Now What?

## A Dive Into PostgreSQL Backup Using pgBackRest and how to use it for PITR

Charly Batista, Postgres Tech Lead @ Percona

# I am Charly Batista

I'm the Postgres Tech Lead at Percona. I'm passionate about database and algorithmic efficiency. I've been working with Database and Development for more than 20 years always trying to get the best performance of the databases, data structures and algorithms... Other than that, I'm a Brazilian who lives in China for more than 6 years but currently located in South America.

You can find me at **https://www.linkedin.com/in/charlybatista**
or more contacts info by the end of this presentation

# Agenda

Backup, HA, Disaster Recovery?

What is what?

PITR?

Ok, how does it really work?

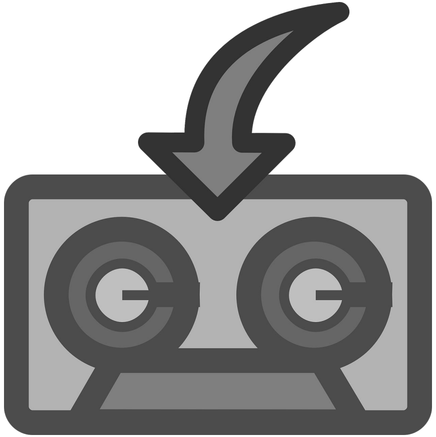
How about pgBackRest?

Questions

# Backup, HA, Disaster Recovery? What is what?

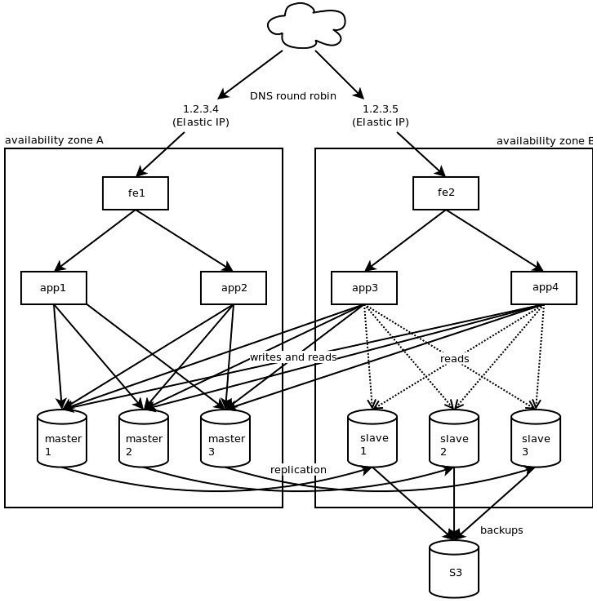
# Backup, HA, Disaster Recovery? What is what?

- Let’s start from the beginning, backups:
  - Backup is just the process of creating a copy of the database;
  - The backup allows to reconstruct the original data when needed;
  - There are different types of backups, for example:
    - Full backups;
    - Incremental backups;
    - Differential backups;
  - Can also be categorized as physical and logical backups;



# Backup, HA, Disaster Recovery? What is what?

- High Availability is the ability to always keep working, no matter what happens;
- It's all about to remove single point of failures;
- It uses concepts like:
  - Clustering
  - Network Load Balancing
  - Geographic redundancy
  - Fail Over

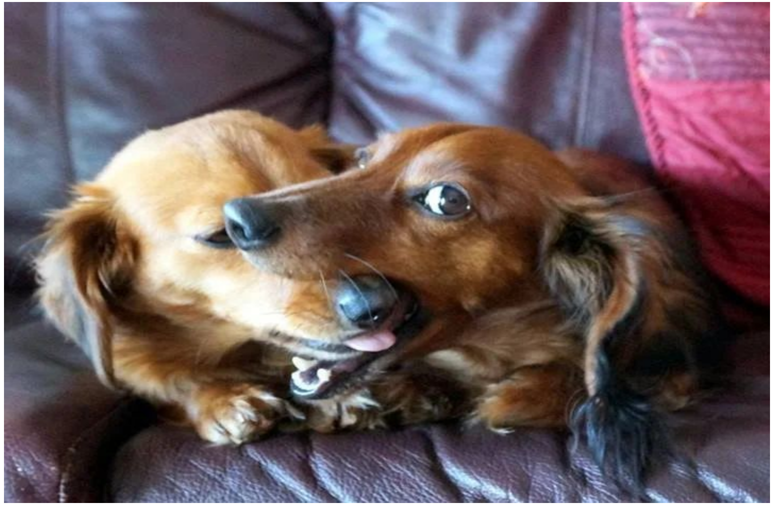


# Backup, HA, Disaster Recovery? What is what?

- Any good HA architecture has a Disaster and Recovery plan;
- We understand that replication isn't enough to prevent data loss:
  - An application bug can delete wrong data;
  - An user can make a mistake and remove and/or update wrong data;
  - Malicious software can also compromise data integrity;
  - All the above, and other scenarios, will propagate the data loss to the replicas;

# Backup, HA, Disaster Recovery? What is what?

- Replication isn’t the replacement of backups, they may bite you back;



- Only backups are the replacement for backups :D

# Backup, HA, Disaster Recovery? What is what?

- However…
- Backups are not always enough for business continuity in case of disaster;
  - What happens if we backup every day at 2am and a disaster happens at 7pm?
- A good Disaster and Recovery plan has contingency for the above and other issues;

# Backup, HA, Disaster Recovery? What is what?

- A Recovery Point Objective (RPO) is needed:
  - It will have the minimum point in time to which the database will be back dated if it was brought back online;
    - For example, with 4h RPO we should be able to recover data for at least 4h ago;
    - It also means that we can lose 4h worth of data!!!

# Point in Time Recovery (PITR)

# How frequently do I need to backup?

- If we have a 4h RPO we can lose 4h of worth of data, but…
  - Backup every hour doesn't solve the problem:
    - It's expensive to backup every hour
    - We can still lose 1h worth of data… backup every minute?

# How frequently do I need to backup?

- Can we do it differently?
- What if we can backup daily and keep the WAL files?
  - We can recover the system to the backup time;
  - Use the WAL files to redo all the transactions from backup time to now!
  - Tell PG to stop recover

    at at any point
  - Ingenious!

    We now have a

    **Point in time recovery**

# Point in time recovery

- Allows us to recovery of data changes up to a given point in time;
- We use a basebackup to restore the database;
- We use the WAL files to progressively fetch and apply the changes;
- We tell PG to stop at a specific target (time, XID, LSN);
- If everything goes well PG will then do a crash-recovery and stop where we asked it to stop;
- We restart the database and validate all went well!

# Ok, how does it really work?

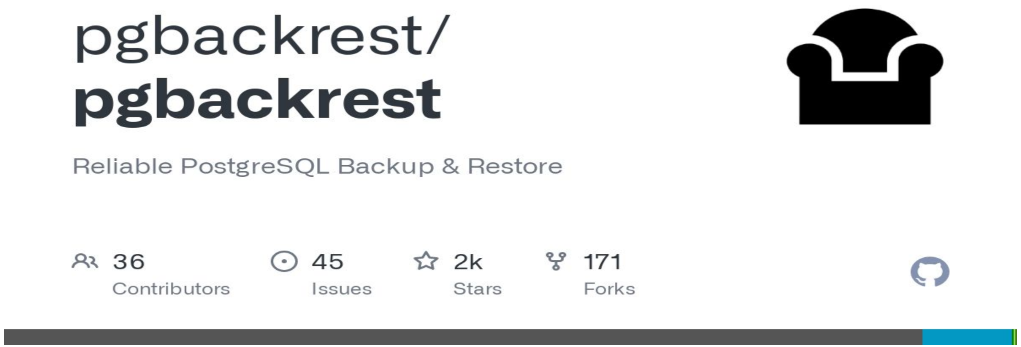
## Ok, how does it really work?

# This is the time we try and **break** things!!

# How about pgBackRest?

# How about pgBackRest?

- pgBackRest is a Postgres backup solution that aims to be:
  - Scalable;
  - Reliable;
  - Easy-to-use;



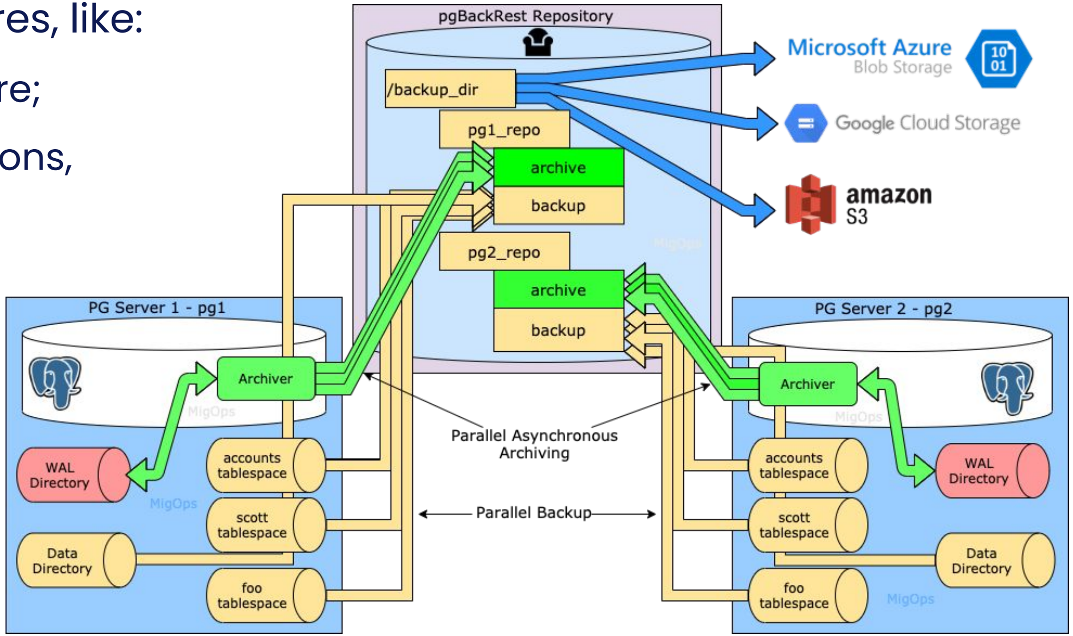
pgbackrest/
**pgbackrest**

Reliable PostgreSQL Backup & Restore

36 Contributors
45 Issues
2k Stars
171 Forks

# How about pgBackRest?

- It has many interesting features, like:
  - Parallel Backup and Restore;
  - Local and Remote Operations, with multiple repositories;
  - Ful, Incremental and Differential backups with integrity checks;
  - Backup rotation and archive expiration…



# How about pgBackRest?

- Our focus today is how to recover from a disaster;
- To use pgBackRest we need to follow the same steps of:
  - Have a base backup and
  - Properly archive the WAL files;

**Too much talk, let's see how that works!**

PERCONA LIVE 2023

# Questions?

PERCONA
LIVE
2023

# Thank you!

You can find me on :

- LinkedIn at **https://www.linkedin.com/in/charlybatista**

ttps://www.linkedin.com/in/charlybatista

- Github at **https://github.com/elchinoo**

PERCONA

# Want to know more about Percona Software For Postgres?