PERCONA

PostgreSQL High Availability:
The Considerations and the Candidates

Dave Stokes
@Stoker
David.Stokes@Percona.com

Talk Proposal

Almost every organization wants a high availability system for PostgreSQL.

This clearly depicts an active trend toward an increase in utilizing PostgreSQL for critical business applications.

In some cases, it is a move away from other major database systems like Oracle or even Teradata.

We are going to define the basic criteria for a high availability system, and have a quick glance at the current open source solutions available in the market.

20 Minutes!

Some Definitions

On What is High Availability

Defining Database High Availability …

…refers to the **continuous operation of a system so that services to end users are largely uninterrupted.** A basic high availability database system provides failover (preferably automatic) from a primary database node to redundant nodes within a cluster. - **Percona**

… is a database system **designed to operate continuously with no interruptions in service**. So database errors and failures must be handled by automatically failing over to redundant nodes when problems occur. -**ScyllaDB**

.. for Cloud Databases means that Cloud Databases users **can run their critical production workloads without worrying about a database becoming unavailable due to the failure of a database component**. It improves the reliability of running a database in the cloud environment by minimizing downtime and ensuring that the application is never down for more than a few seconds in the event of a failure. - **Rackspace**

…is a characteristic of a system which **aims to ensure an agreed level of operational performance**, usually uptime, **for a higher than normal period**. - **Wikipedia**

How Much Data *Can* You Lose?!

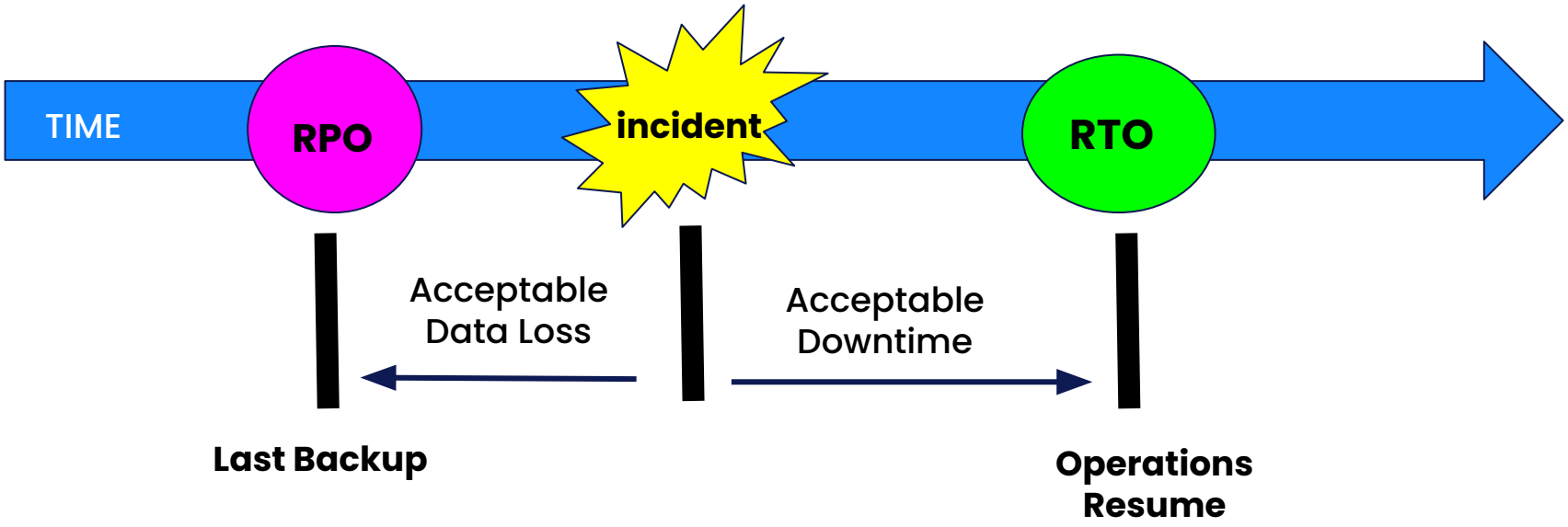
None

1. Design around temporary loss of database or other component at a diminished capacity.
2. Have a huge budget.

Some

1. Negotiate ‘target’ times
   a. RTO - Recovery Time Objective
      i. How long to recover from a failure
   b. RPO - Recovery Point Objective
      i. How much data can be lost when that failure occurs.
2. Have a large budget.

What is *your* allowable data loss?



Types of Failure

High Availability

- Network Partition
- Server Failure

Disaster Recovery

- Full Region/Network Failure

Human Error

- Little Bobby Tables

The plane still managed to land safely. One Stewardess was sucked out of the plane.

High Availability

Wasn’t that a 1970’s Cheech and Chong movie?

Simplified HA Steps

1. **No SPOFs**
   a. No Single Points of Failure
2. **Redundancy**
   a. redundant systems must be supported by failure detectability mechanisms and, in the event of an issue, failover must be handled automatically and seamlessly.
3. **Engineered Solution**
   a. When X fails, transfer to Z in a well ordered fashion
4. **Reconciliation**
   a. How X is recovered back into cluster

Your high availability needs will vary and depend on numerous factors, including:

- criticality of the applications - *Not all animals are equal*
- customers and users impacted -*Tiers of user levels*
- how quickly does a database of application need to failover - *'limp mode'*
- acceptable limits on data loss

For mission-critical applications, such as those used in e-commerce, **four nines** (99.99% or ~52 minutes/year) **is generally considered the industry standard**.

These factors may also depend on pieces of the application. I.e. order taking never down, tracking shipment does not need to be anywhere near real time.

Reality Checks

- **Maintenance is a requirement – hardware, software, people.**
  - Plan to work around maintenance periods.
- **You will lose data**
  - Capture issues, corruption, malfeasance, technological issues.

Reality Checks

- Having near zero downtime is expensive
  - Just because open-source is ‘free’ of purchase cost does not mean you can be cheap.
  - Proper staffing levels, training, and practice are all basics.
- Be aware that shifting costs or technology does not absolve you of basic responsibilities like backups, data retention rules, and good operational practices.
- Treat High Availability as an ***on-going***, evolving practice.

So what do you need?

There are a few methods for achieving high availability with PostgreSQL:

- shared disk failover
- file system replication
- trigger-based replication
- statement-based replication
- logical replication
- Write-Ahead Log (WAL) shipping via streaming replication

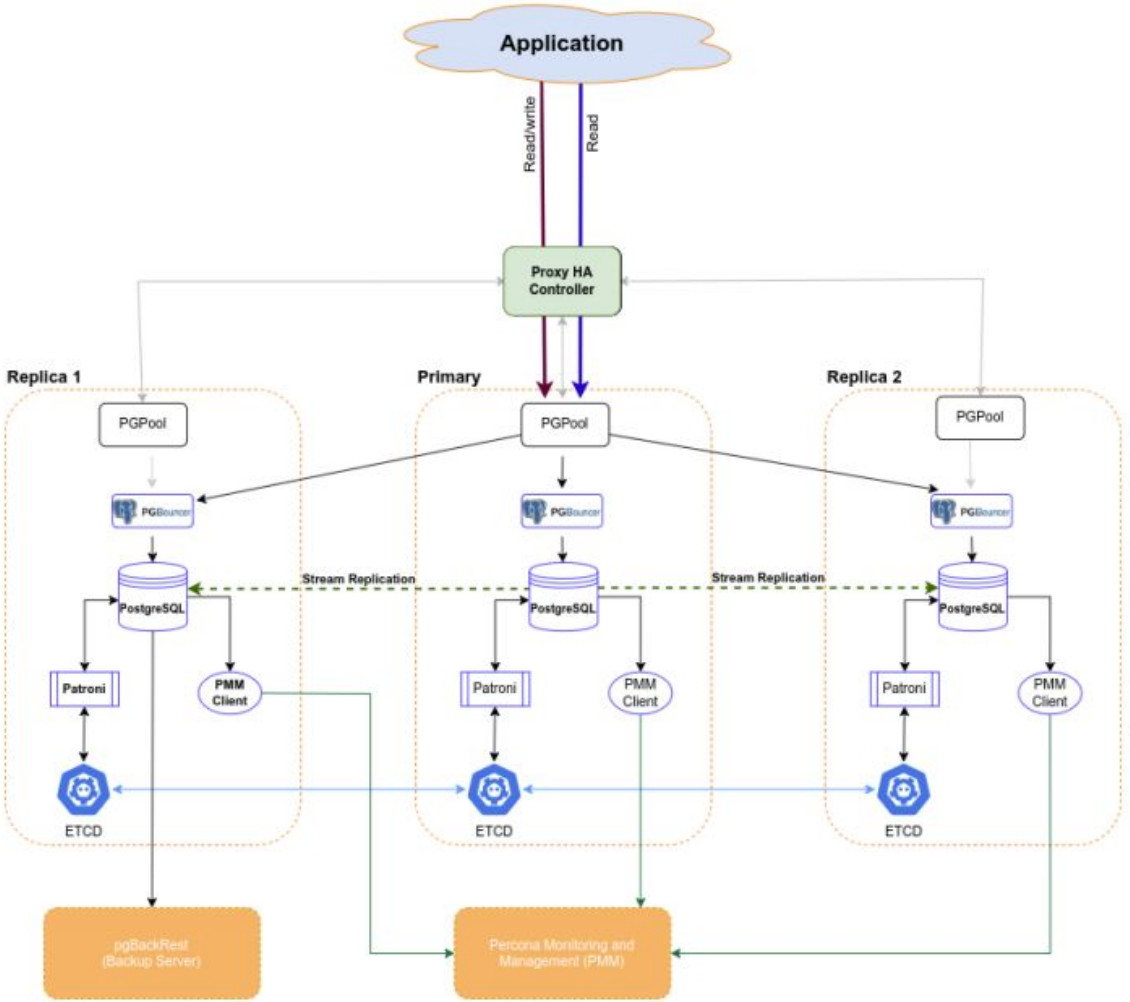
Streaming replication is part of Write-Ahead Log shipping, where changes to the WALs are immediately made available to standby replicas. With this approach, a standby instance is always up-to-date with changes from the primary node and can assume the role of primary in case of a failover.

Why native streaming replication is not enough

Although the native streaming replication in PostgreSQL supports failing over to the primary node, it lacks some key features expected from a truly highly-available solution. These include:

- No consensus-based promotion of a “leader” node during a failover -RAFT/PAXOS
- Capability for monitoring cluster status
- No automated way to bring back the failed primary node to the cluster
- A manual or scheduled switchover is not easy to manage

To address these shortcomings, there are a multitude of third-party, open-source extensions for PostgreSQL. The challenge for a database administrator here is to select the right utility for the current scenario.

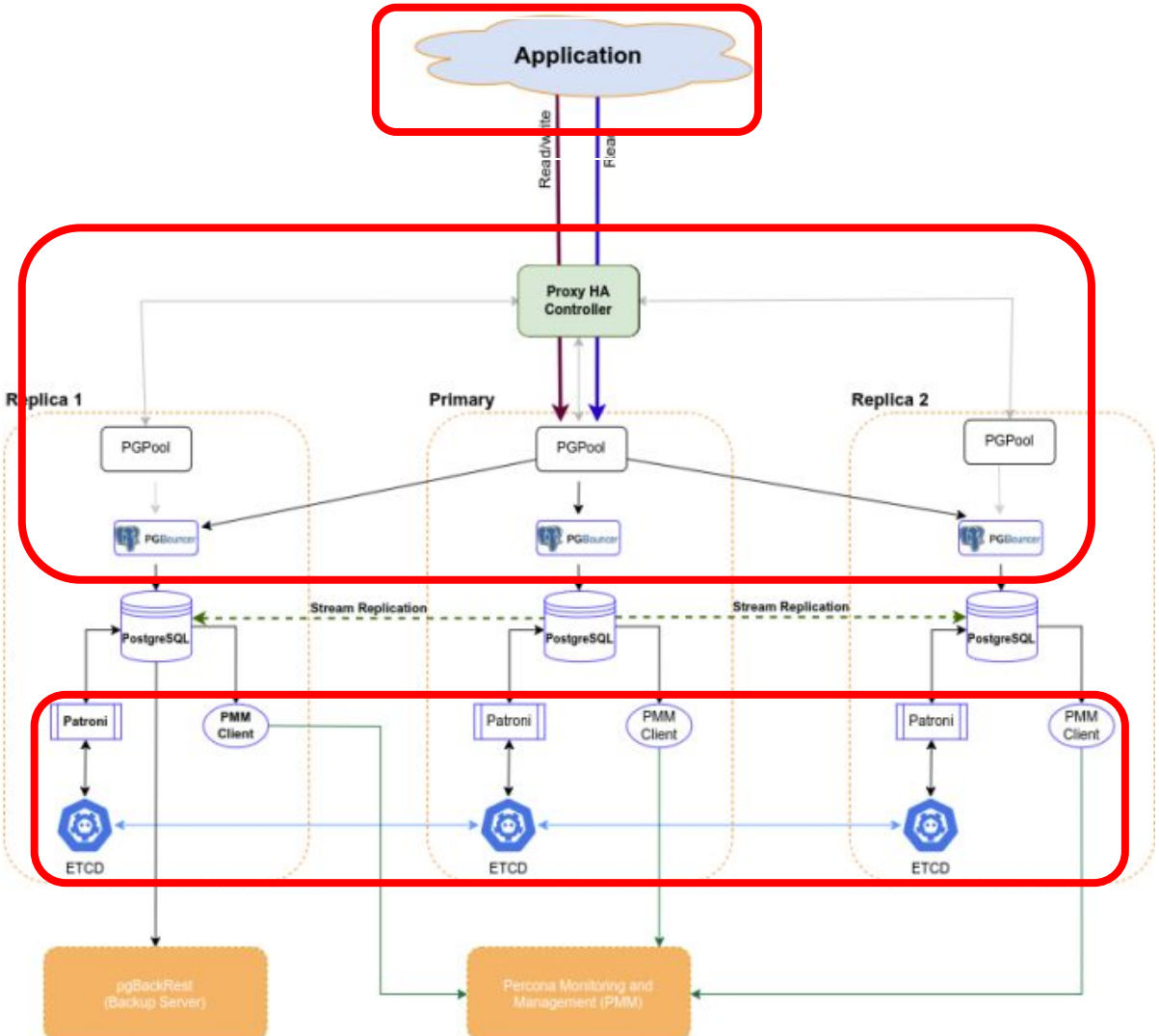


Assumptions: You want four or five nines availability, you have trained staff that are paying attention, and there will be no *force majeures* allowed.

Note:

1. Five nines availability 99.999% uptime is 5.256 minutes a year or .014 minutes a day.
2. Big budget on hand, right?
3. Management is behind you, watching your back.
4. You have a lot of luck!!

PERCONA



Areas of vulnerability

Ironically the database part is the easiest.

You usually can control the proxy and coordination segments together.

Minimize non-essential network traffic, segment as much as possible.

PERCONA

Wrap up!

Some reading

https://www.postgresql.org/docs/15/high-availability.html

https://docs.percona.com/postgresql/12/solutions/high-availability.html

Percona Live -
https://www.percona.com/live/conferences

May 22–24 at the Denver Marriott Tech Center!

Thank You!

David.Stokes@Percona.com
@Stoker
Speakerdeck.com/Stoker