STORMATICS

Designing Layered High-Availability Architectures for PostgreSQL on a Budget

SREcon Americas

26-Mar-2026

STORMATICS

The Problem

Most PostgreSQL HA deployments are either over-engineered for the actual risk, or under-built for what's actually at stake

Both cost you money - in different ways

Over-Engineered

- Multi-region sync replication for a non-prod database
- Complex Patroni clusters nobody can troubleshoot
- Paying 4x for infrastructure you don't need

Under-Built

- Production databases with no standbys and no backups
- RPO and RTO are undefined and untested
- "We'll add HA later" - and then outage hits

STORMATICS

A Different Approach

What if you could build HA incrementally?

Could you add protection as your risk profile and budget actually demand it?

Additive Layers

Each layer builds on the last.
Nothing gets ripped out.

Budget-Aware

Pay only for the resilience
you actually need right now.

Operationally Honest

Don't build what your
team can't maintain.

STORMATICS

Start with Outcomes

Before choosing a topology, align on three things.

Failure Scope

What are you protecting against?
A host fails. A zone goes away.
A full region outage. Human error.

RPO

How much data loss is tolerable?
15 minutes of transactions?
Or close to zero?

RTO

How fast must service return?
Back in 30 minutes?
Or under 2 minutes?

My stance: you get strong availability outcomes by layering in the right order. And it saves money.

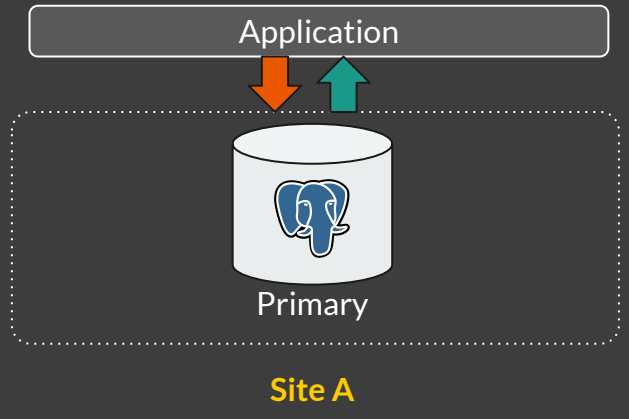
STORMATICS

The Layered HA Model

Layer	What It Adds	RPO	RTO
0	Single Node	Undefined	Hours+
1	+ Offsite Backups	Hours	Hours
2	+ WAL Archiving (PITR)	Minutes	Hours
3	+ Hot Standby (Async)	Seconds	**Seconds**
4	+ Hot Standby (Sync)	**Zero**	**Seconds**
5	+ Cross-Region Warm Standby	Minutes	Minutes

STORMATICS

Layer 0 - The Baseline



Write
Read

RPO

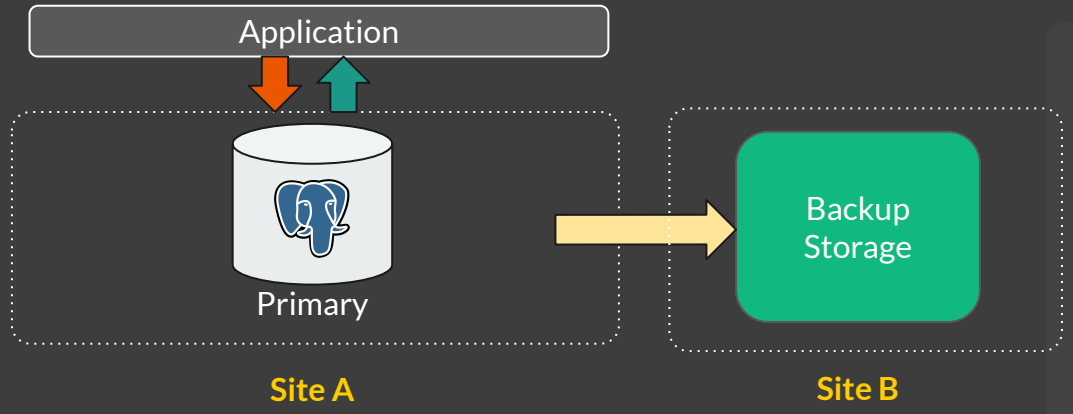
Undefined

RTO

Hours to days

STORMATICS

Layer 1 - The First Safety Net



RPO

Hours

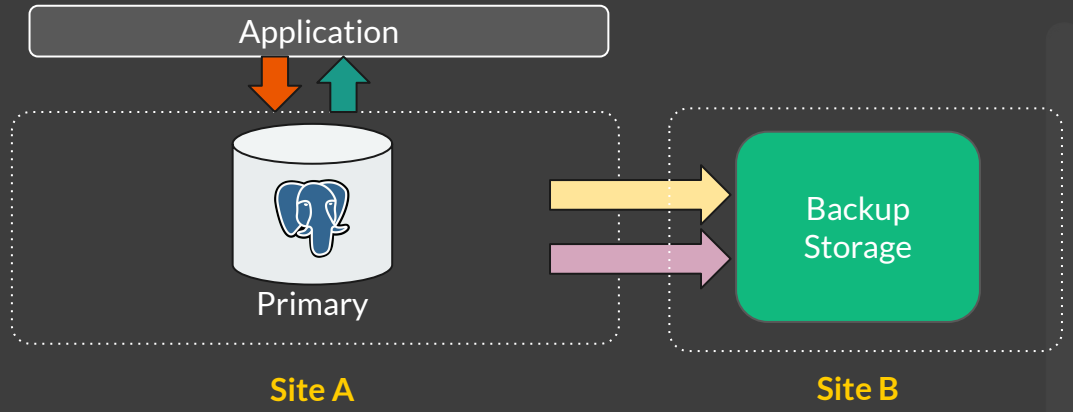
RTO

Hours

- Write
- Read
- Backup

STORMATICS

Layer 2 - Point in Time Recovery



RPO

Minutes

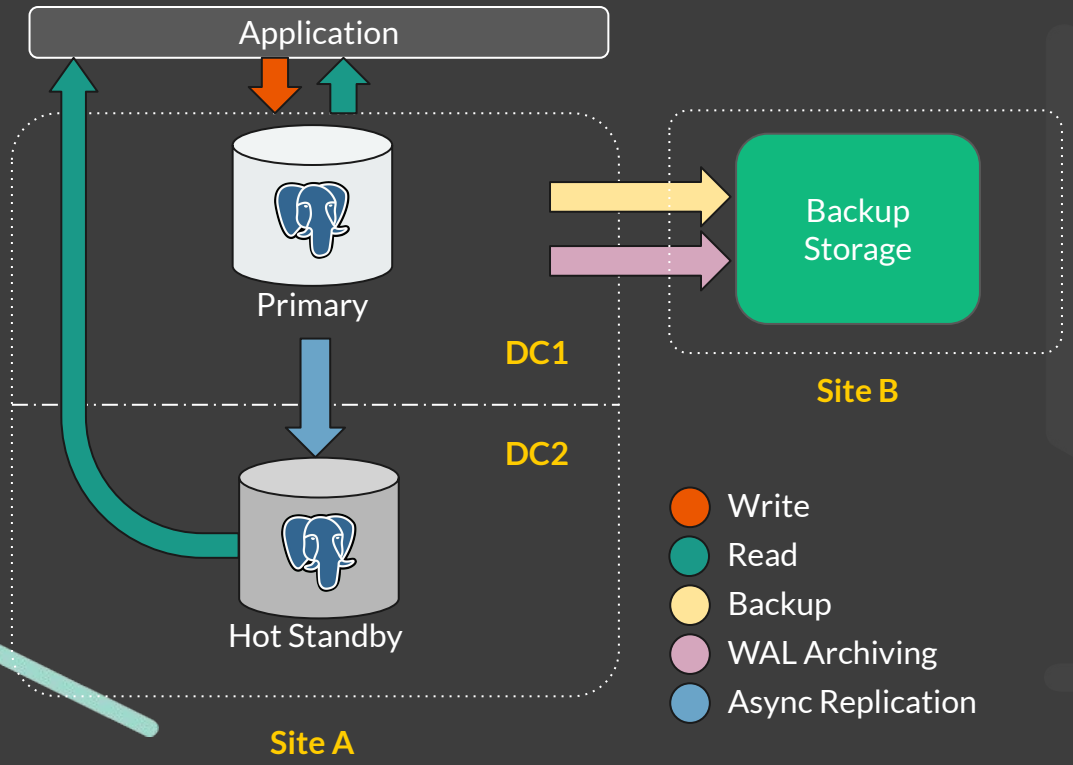
RTO

Hours

- Write
- Read
- Backup
- WAL Archiving

STORMATICS

Layer 3 - The First True HA Layer



RPO

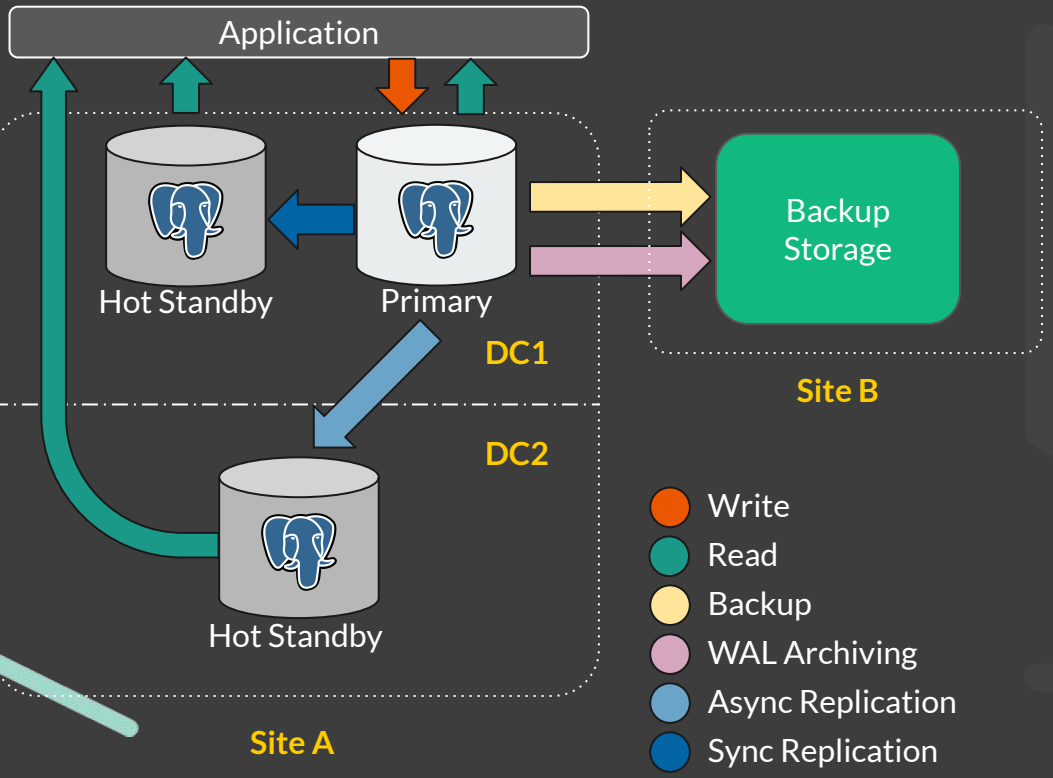
Seconds

RTO

Seconds

STORMATICS

Layer 4 - Zero Data Loss



RPO

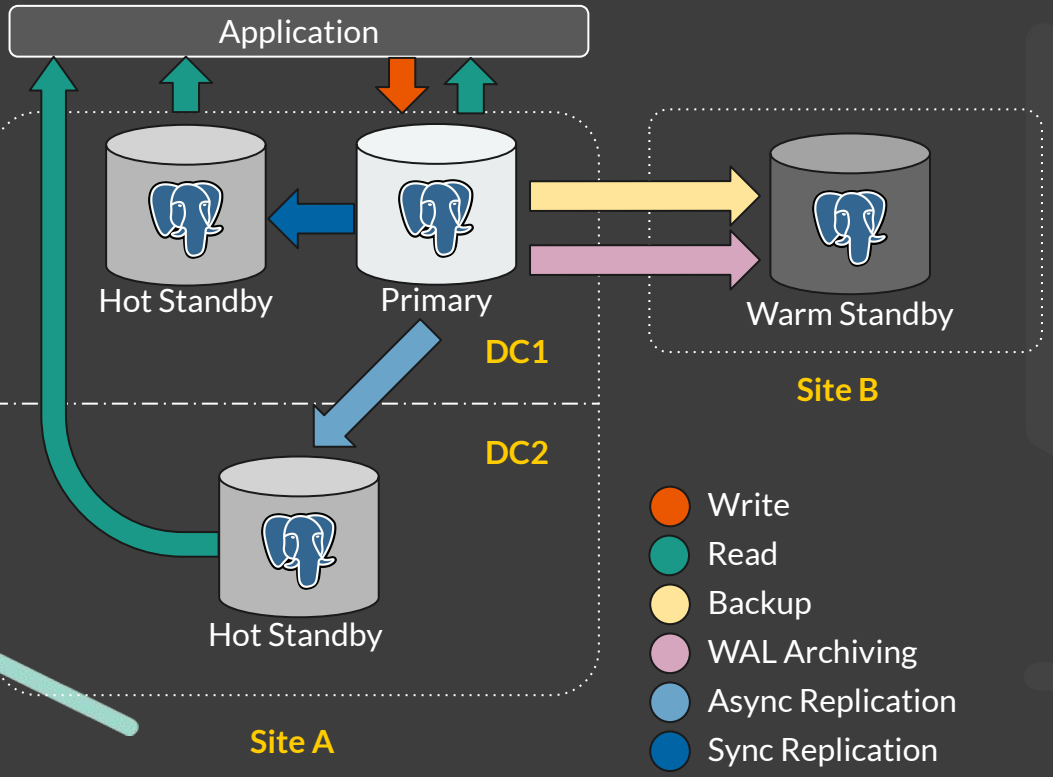
Zero

RTO

Seconds

STORMATICS

Layer 5 - Survive a Regional Outage



RPO	Cross-Region RPO
Zero	Minutes
RTO	**Cross-Region RTO**
Seconds	Minutes

STORMATICS

Which layer do you need?

	Acceptable Downtime	Acceptable Data Loss	Budget	Team Expertise
Layer 0	Any	Any	Minimal	None
Layer 1	Hours	Hours	Low	Basic
Layer 2	Hours	Minutes	Low	Basic
Layer 3	Seconds	Seconds	Moderate	Intermediate
Layer 4	Seconds	Zero	Higher	Advanced
Layer 5	Minutes*	Minutes*	Highest	Advanced

** Layer 5 RPO/RTO is for cross-region failover only. Within-region failover is still seconds (Layer 4).*

STORMATICS

The Split-Brain Problem

Split-brain: two nodes both believe they are the primary and accept writes.
This is the worst failure mode in any HA system. Data diverges and manual intervention is required to reconcile.

HOW PATRONI PREVENTS IT

Fencing the Old Primary

Patroni changes pg_hba.conf and restarts the old primary to reject connections. No writes can reach it.

DCS-Based Consensus

Only the node holding the DCS leader key can be primary. No key, no writes. The DCS is the single source of truth.

Watchdog Integration

If Patroni can't demote the old primary gracefully, the watchdog reboots the entire node. Brutal, but effective.

STORMATICS

Patroni: The Failover Engine

Patroni uses PostgreSQL's built-in replication to build an automated HA cluster.

1

Leader Election

Uses etcd/ZooKeeper/Consul for distributed consensus. One primary, automatic promotion on failure.

2

Health Checks

Continuous monitoring of PostgreSQL processes, replication lag, and disk health. Configurable thresholds.

3

Automatic Failover

Promotes the most up-to-date standby. Fences the old primary. Updates the Distributed Configuration Store (DCS).

STORMATICS

Sync Replication: The Hidden Costs

Synchronous replication gives you RPO=0. But it comes with trade-offs that can hurt more than the problem you're solving.

Write Latency Increase

Every COMMIT waits for at least one standby to confirm WAL receipt. Same-AZ: <1ms penalty. Cross-AZ: 2-10ms. Cross-region: 50-200ms.

Medium

Sync Standby Failure Stalls Writes

If the sync standby goes down, the primary blocks all writes until reconfigured. Patroni's synchronous_mode_strict controls this behaviour.

High

Operational Complexity

Your team needs to understand synchronous_standby_names, synchronous_commit settings, and how Patroni manages sync membership.

Medium

STORMATICS

Cost Saving Techniques

Right-Size Your Standbys

Standbys don't need the same instance type as the primary. A smaller instance handles replication and read queries fine. Scale up only if promoted.

Compress and Deduplicate Backups

pgBackRest supports compression and deduplication out of the box. A 500GB database often compresses to under 100GB in backup storage.

Stagger Your Layers

Don't deploy all 5 layers at once. Start at Layer 2 or 3. Add layers as risk grows. Every layer you defer is infrastructure cost saved.

STORMATICS

Lessons from Production

1. Test your failover regularly. A failover you've never tested is a hypothesis, not a plan.
2. Backups exist but restore is untested? That's Schrödinger's backup; you only know when there's an outage.
3. WAL archiving configured but not monitored? Make sure the consumer is consuming, or files pile up on the producer.
4. Replication slots retain WAL longer than expected. Monitor them, and ask 'why' when they grow.
5. Sync replication without a clear failure policy. Write the rule down, test it, make it visible to on-call.
6. Read traffic routed to standbys without thinking about staleness. Choose the right queries for replicas.
7. Don't skip the watchdog. It's the last line of split-brain defence and it's free.

STORMATICS

Key Takeaways

HA is a spectrum, not a switch. Build incrementally.

Start at Layer 2 or 3. Earn your way to Layer 5.

Don't build what your team can't operate.

Topology matters more than replication mode.

Test everything. Untested HA is theatre.

Questions?

pg_umair