several9s

PostgreSQL Bi-Directional Replication

Housekeeping items

- Standard presentation, demo and Q&A format
- 2 Polls

AGENDA

- What it is and isn’t and what it solves for
- Its pros and cons and when to use it
- Implementation reqs and process
- What is in store for it in PG 18
- ClusterControl’s role in ops

Your expert presenter

Who: Sebastian Insausti
What: Senior Support Engineer
Where: Severalnines
Why: 18 years DBA, DevOps, SRE, Sysadmin, Network Administrator experience

```
<img src="image.jpg" alt="Image"
width="300" height="300">
```

But First!

Let's Take Your Pulse

What PostgreSQL Bi-Directional Replication is and isn’t

Core concepts — Logical + Streaming Replication

Logical Replication

Replicating individual tables and their data changes between databases.

Based on a publish/subscribe model.

Allows replicating specific tables, even across different PostgreSQL versions.

Streaming Replication

Streaming low-level changes from a primary to one or more standby servers.

Sends binary WAL data.

Creates exact copies of the primary.

Standbys are read-only.

What it is

- 2 or more nodes (data sources and receivers)
- Writes on multiple nodes
- Keeping nodes in sync
- Active-active setups
- Geo-distributed applications
- High Availability envs
- Native support

What it isn’t

- Not native synchronous multi-master replication
- Doesn’t handle conflicts automatically
- Doesn’t replicate DDL
- Not eventually consistent by default (asynchronous replication - lag)

What the feature solves for

What can it do for you?

- Regional Write Availability
- Active-Active Architecture
- Failover with Write Continuity
- Rolling Upgrades and Migrations

What are its pros and cons

PG Bi-Directional Replication — Pros and Cons

Pros

- Multi-Region Write Access
- High Availability
- Scalable Writes
- Zero-Downtime Upgrades
- Table-Level Replication

Cons

- No Built-In Conflict Resolution
- No DDL Replication
- Replication Lag
- Operational Risks (conflicts)
- Increased Complexity

When to and not to use it

When to use it

- Global applications needing local write access in each region.
- Multi-write environments with concurrent writes across zones.
- Zero-downtime upgrades.
- Regional autonomy for disaster recovery.

When not to use it

- Frequent write conflicts or no way to isolate writes.
- Hard-to-manage conflict resolution.
- Frequent DDL changes.
- Strict consistency required.
- Simple HA needs met by primary + standby setup.

Implementation basics

Implementation — requirements

Environment

- PostgreSQL Version: 10 or higher
- Operating System: Ubuntu 20.04+, RHEL 8+, Debian 11+

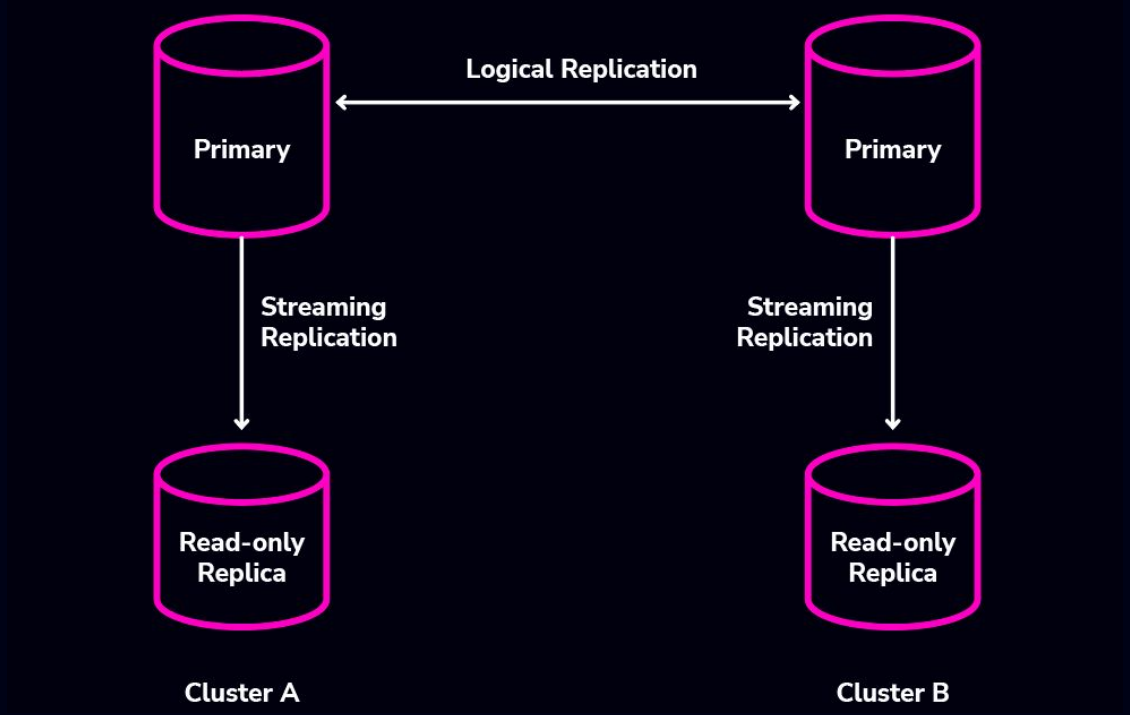
Hardware

- CPU: Minimum 2 CPU
- RAM: 4 GB per node
- Storage: SSDs (or similar)

Networking

- Low-latency inter-node communication.
- Properly configured firewalls
- Authentication for replication

Implementation — basic topology



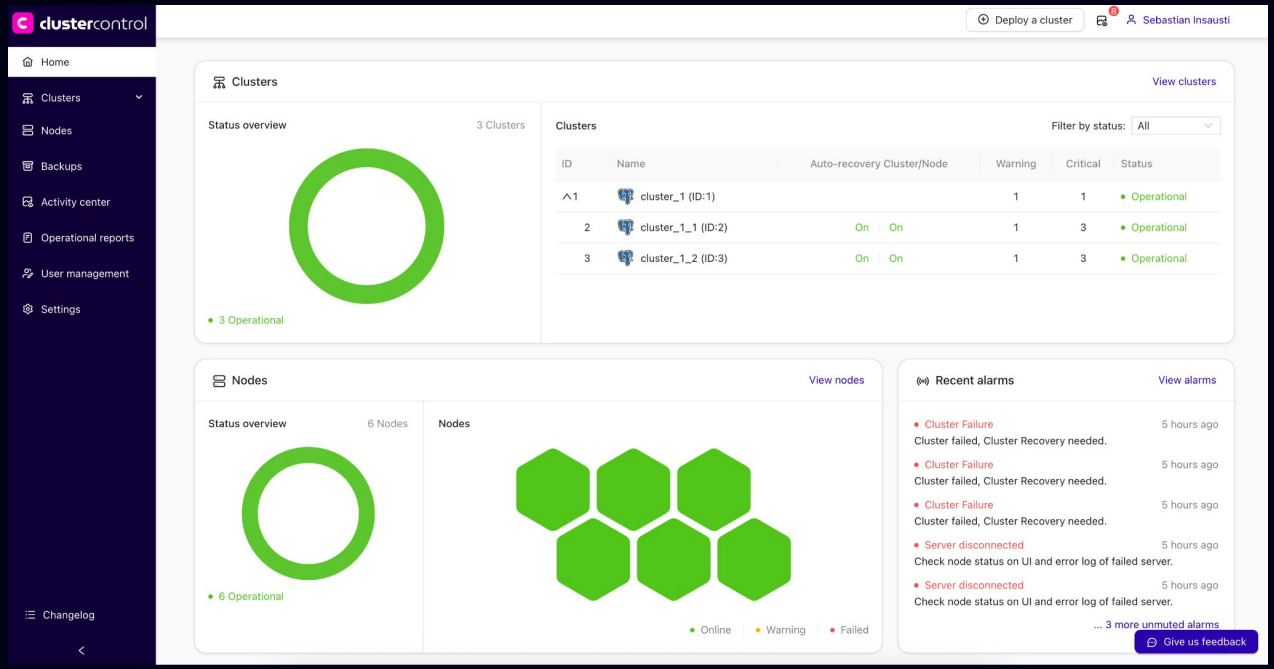
Implementation — manual process

- **Ensure Logical Replication is Enabled:** Set *wal_level = logical*, and configure *max_replication_slots, max_wal_senders, and wal_keep_size.*
- **Create a Replication User:** A user with appropriate privileges on tables to be replicated. Also, add it to *pg_hba.conf*
- **Set Up Publications on Each Primary:** *CREATE PUBLICATION pub_cluster_a FOR TABLE my_table;*
- **Set Up Subscriptions on the Peer Cluster:** *CREATE SUBSCRIPTION sub_from_a CONNECTION 'host=cluster_a user=replicator dbname=mydb' PUBLICATION pub_cluster_a;*
- **Repeat in Reverse:** Cluster B → Cluster A

Implementation — considerations

- Asynchronous Only: Changes are not immediately visible on remote clusters.
- Conflict Risks: No built-in conflict resolution. Design your app/schema carefully.
- Manual DDL Management: Logical replication does not replicate schema changes.
- Replication Lag: Ensure replication lag is monitored and resolved if needed.
- WAL Management: Maintain healthy WAL file retention to prevent disk usage or replication issues.

Implementation — ClusterControl



What its future holds

PostgreSQL 18 (September 2025)

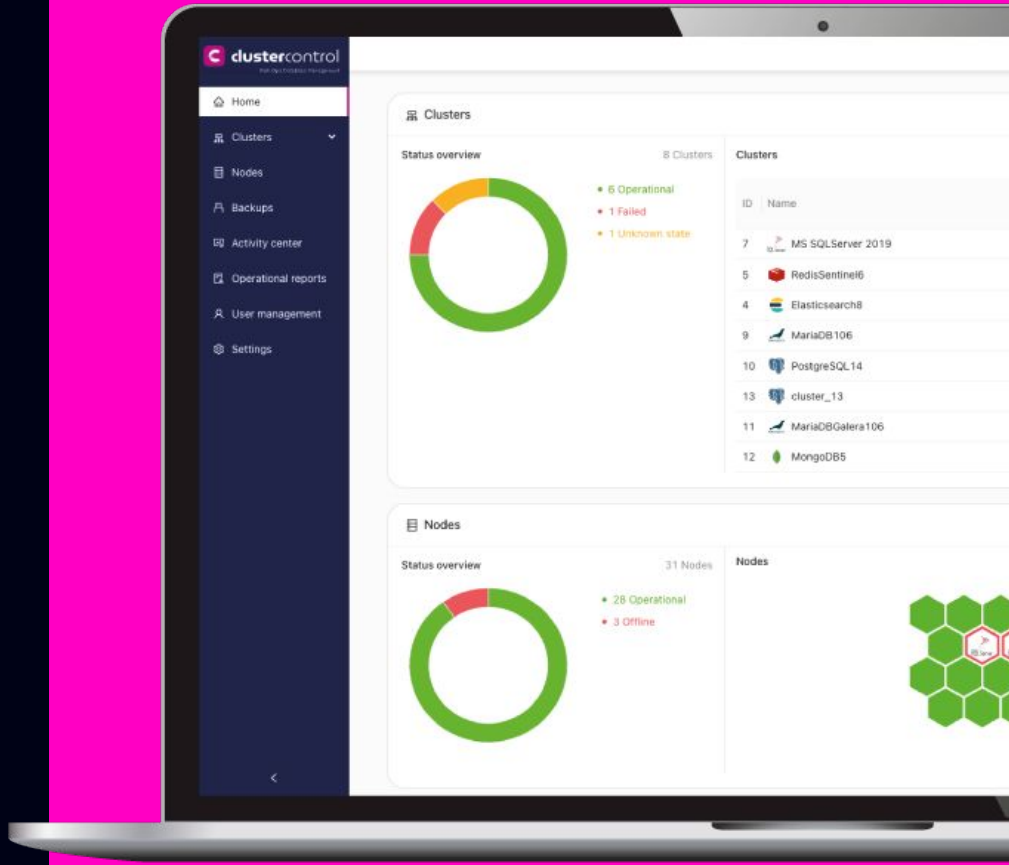
- Replication of Generated Columns
- Improved Conflict Detection
- Replication Slot Timeout
- DDL Replication (still under discussion)
- Built-In Conflict Resolution (still under discussion)

How CC can help ops

Introducing ClusterControl

Database ops orchestration platform to deploy, monitor, manage, and scale database ops in any environment:

- Self-hosted in on-prem and hybrid environments
- Centralized monitoring and management from a single pane of glass
- Supports open-source and source-available databases
- Integrates with popular tooling - Terraform, Ansible, Puppet, etc



ClusterControl - System requirements

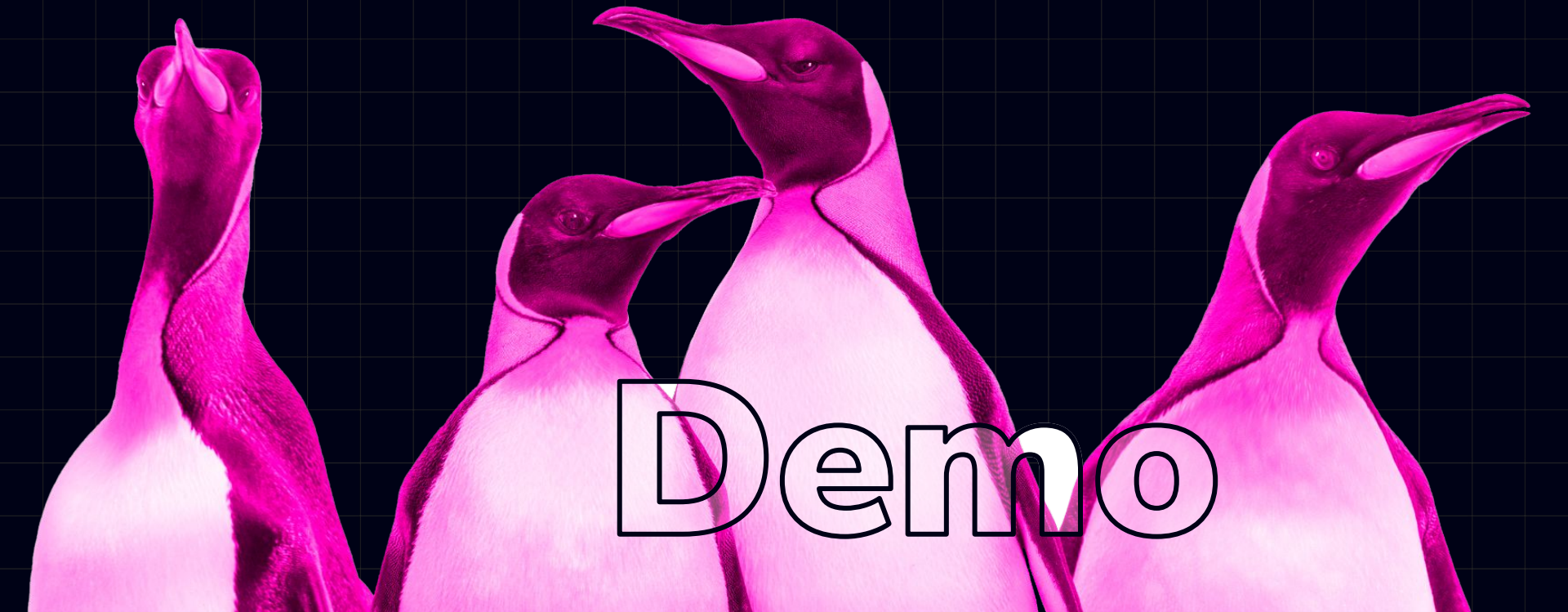
Hardware

- Arch: x86_64 only
- RAM: >2 GB
- CPU: >2 cores
- Disk space: >40 GB
- Cloud platforms: AWS, Google Cloud, Microsoft Azure Oc

Operating Systems

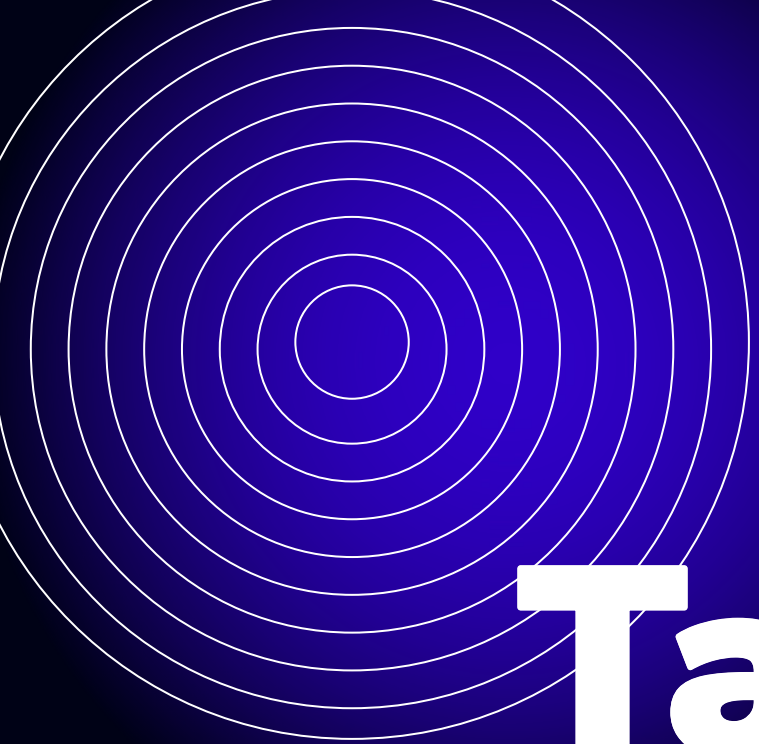
- RHEL 8.x/9.x
- Rocky Linux 8.x/9.x
- AlmaLinux 8.x/9.x
- Ubuntu 18.04/20.04/22.04/24.04 LTS
- Debian 10.x/11.x/12.x
- SUSE Linux Enterprise Server 15 SP3/15 SP4

Short



Demo

Q&A



And Lastly!

Let's Take Your Final Pulse

Thank you!