



CASE STUDY

Keeping Payments Moving — CoralPay's 99.99% Uptime Story with ClusterControl

CoralPay

Industry: Financial services

Technologies: Galera, MariaDB

Products: ClusterControl

Environment: 2 data centers

Use case

Operating high-availability MySQL Galera and MariaDB Replication environments with full on-premises control for national-scale payment-processing workloads.

Why Severalnines

ClusterControl provides the automation and management capabilities needed to improve system uptime, strengthen database visibility, and significantly reduce the operational burden of managing complex payment environments.

Summary

If you own both uptime and budget, you need a database platform that proves out on paper and in prod. CoralPay adopted Severalnines ClusterControl to standardize their MySQL Galera and MariaDB Replication operations — moving reporting workloads to replicas, automating failover and recovery, and giving leaders a single source of truth for cluster health. The result is tangible: 99.99% measured availability, fewer after-hours incidents, and a leaner operational model that scales without adding headcount.

Background

CoralPay is a Central Bank of Nigeria licensed, Tier-1 payments switch serving banks and merchants nationwide. Reliability isn't a nice-to-have, it is the contract. The team runs on-prem MySQL Galera and MariaDB Replication with some MSSQL Server and wanted to improve the following operations across two production clusters supporting an average of 15 TPS with peaks of up to 45 TPS, and sizeable schemas of approximately 386.2 GiB and 320.3 GiB:

- near real-time replication,
- clean read offload, and
- consolidated visibility, so a small team could manage a complex estate.

CoralPay's challenge

Before ClusterControl, database operations were siloed and unpredictable. Reporting jobs sometimes competed with payment traffic; recoveries demanded hands-on work; and visibility was fragmented. With a junior DBA in the chair, leadership needed a UI-first control plane that would make the riskier ops, e.g. replication, failover, backups, etc., consistent and auditable.

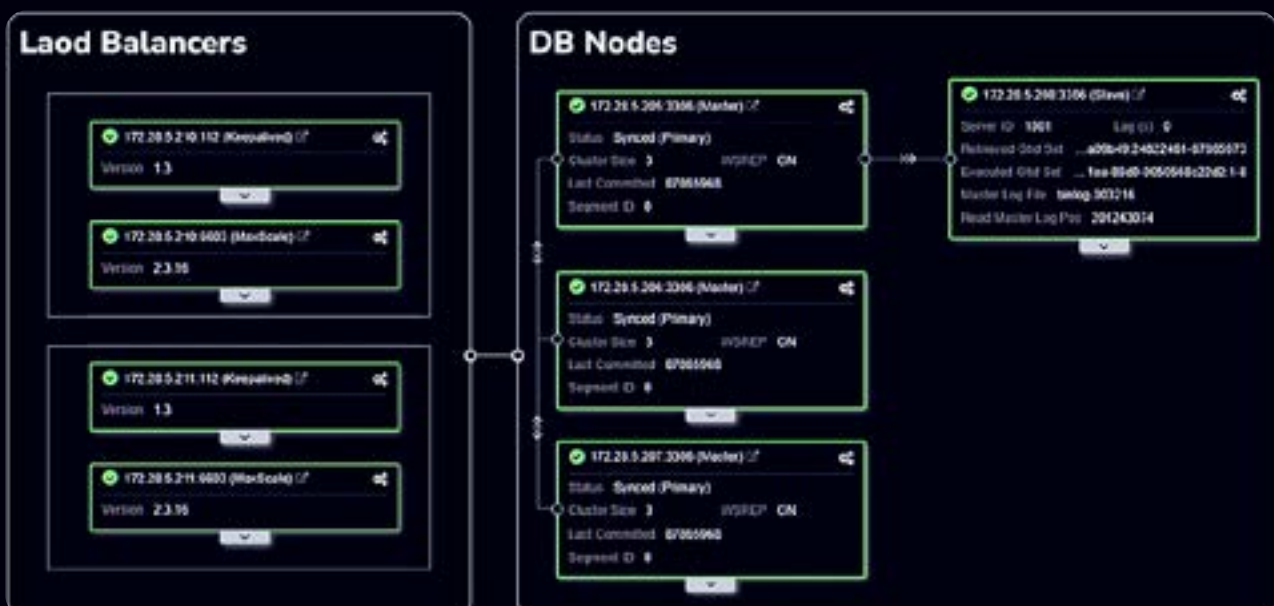
From a business lens, the cost of uncertainty was real: higher on-call load, slower recovery, and avoidable SLA exposure. The selection criteria reflected both sides of the house:

- automation with auto-healing,
- end-to-end observability and alerting, and
- an interface that reduces command-line usage while preserving fine-grained control.

Selecting ClusterControl

CoralPay standardized on **Severalnines ClusterControl** as the control plane for clusters, replicas, backups, and the load-balancing tier — all from one console.

Architectural snapshot (at time of writing): two data centers, multi-node MySQL Galera, and MariaDB Replication clusters, read replicas for reporting, and a MaxScale load-balancing layer — all orchestrated and monitored through ClusterControl — current replica lag show 0s:



Rollout, briefly: with Severalnines on standby, the first production cluster went live in ~2 days; subsequent clusters were deployed by CoralPay alone. Automated failover and guided recovery replaced ad-hoc runbooks. Unified observability surfaced slow or unindexed queries before they harmed transactions. Backups and restores became policy-driven rather than ticket-driven.

What this unlocks technically: predictable failover behavior; push-button replica management for read offload; consistent backups and restore drills; and query-level visibility to remove duplicate or unindexed hotspots.

Experience highlight

Confidence changed the operating rhythm. Reporting traffic moved cleanly to replicas, primaries stayed reserved for payments, and incidents no longer required round-the-clock deep dive CLI sessions. Vendor support was responsive during rollout and rarely needed afterward — a sign the platform became routine.

Measurable results and benefits

CoralPay reports a steadier, quieter database layer — and measurably so. Availability is tracking at 99.99%, while stakeholders also point to improved uptime and response time as the business KPIs that matter most. The platform currently supports an average of 15 TPS, with peaks of approximately 30 to 45 TPS. Replication monitoring shows 0s lag in normal conditions, helping the team keep reporting and analytics activity away from production payment flows.



“We hardly need to worry about production DB monitoring. Alerts are frequent and ClusterControl’s auto-recovery keeps everything in sync. Life is easier for the team.”

Chukwuka Okoh, Executive Director IT & Operations

Auto-healing and guided recovery shorten the path from incident to resolution. Proactive alerts cut down firefighting. Read replicas absorb reporting spikes, so customer-facing throughput stays crisp during peaks. A small team now runs a sophisticated estate without calling in senior DBAs for day-to-day tasks.

Wrapping up

By adopting ClusterControl, CoralPay replaced manual heroics with dependable automation, clear visibility, and a deliberate split between transactional and reporting workloads. For technical leaders, the through line is simple — less risk, better performance where it matters, and an operating model that scales without a linear headcount curve.



Ready to automate your databases anywhere?

Download now and you'll be running your database in just minutes.

Get started