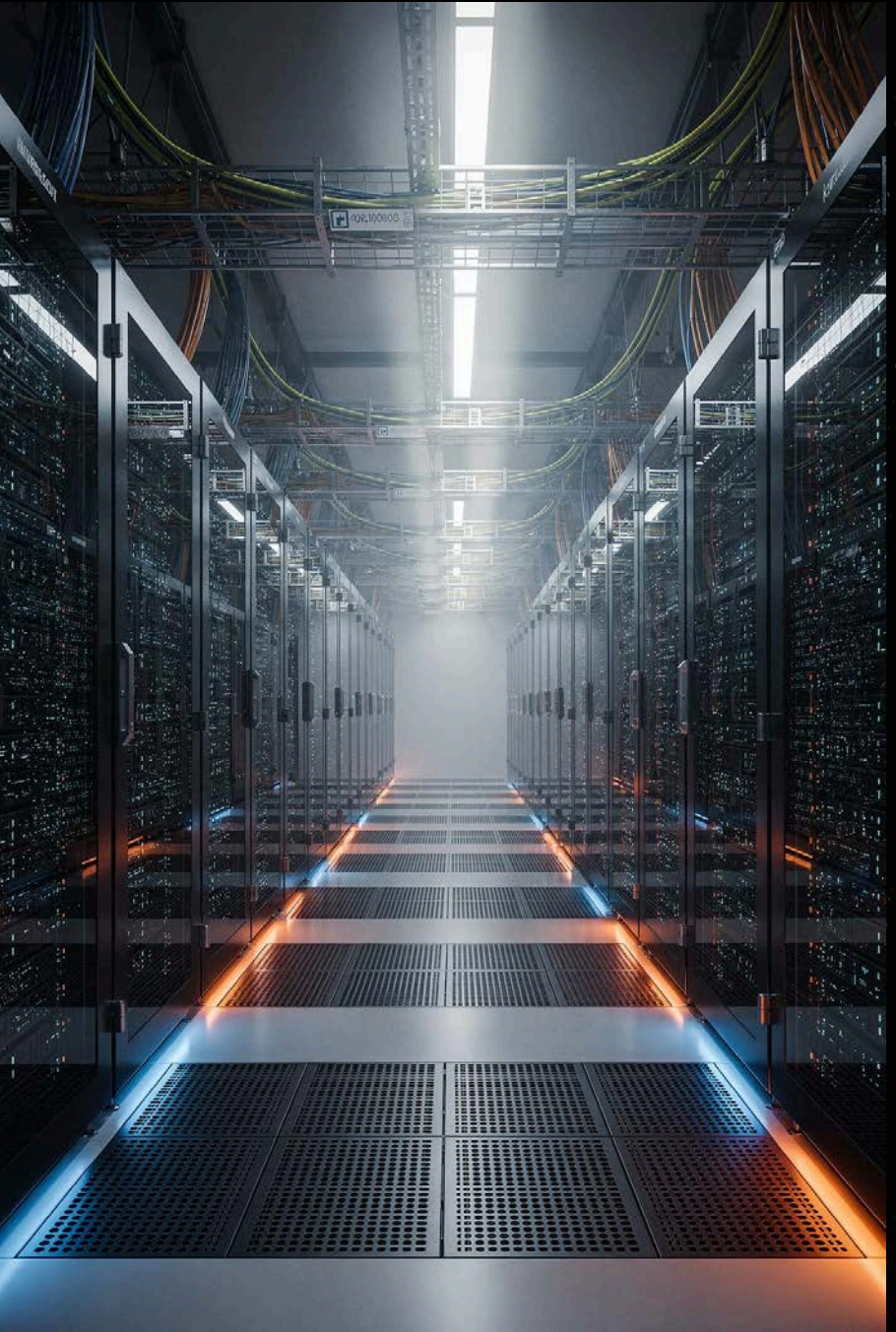


Building Highly Available Databases with Cloud Native Reliability Practices

Conf42 Site Reliability Engineering (SRE) 2026

Meet Your Speaker

Rajesh Kumar Balusu

Database Architect
AGAP Technologies Inc

Database Architecture

Designing resilient, high-performance database systems for production environments

Cloud Native Infrastructure

Kubernetes, containerization, and SRE-aligned operational practices

Reliability Engineering

SLO definition, migration strategies, and disaster recovery planning

Session Agenda

A structured 30-minute deep dive across three core pillars of database reliability

01

Reliability Assessment & Planning

SLO definition, infrastructure review, capacity planning, and traffic controls

02

Cloud Native Platforms for SRE

Containers, Kubernetes, CI/CD pipelines, and operational metrics

03

Structured Migration with the 6 Rs

Dependency analysis, phased execution, HA principles, and post-migration optimization

Why Database Reliability Demands SRE Ownership

As organizations modernize production systems, databases are often the last mile of reliability the layer where architectural gaps become outages.

Uptime Pressure

Business SLAs increasingly demand five nines leaving less than 5 minutes of downtime annually

Hidden SPOFs

Legacy environments hide single points of failure that surface only during incidents

Operational Drift

Aging components and undocumented workloads make reliability unpredictable

Defining Reliability Objectives

CHAPTER 1 — ASSESSMENT & PLANNING

Before any architecture change, SREs must establish measurable reliability targets grounded in business requirements not aspirational guesses.

SLO Service Level Objective

Target uptime, e.g., 99.999% (five nines = <5.26 min/year downtime)

RTO Recovery Time Objective

Maximum acceptable time to restore service after a failure event

RPO Recovery Point Objective

Maximum tolerable data loss window, driving backup and replication frequency

Latency Thresholds

P99 query response targets defining acceptable read/write performance bounds

CHAPTER 1 — ASSESSMENT & PLANNING

Infrastructure Review: Finding What Breaks

A structured review surfaces hidden reliability risks before they become production incidents.

Single Points of Failure

Identify standalone database nodes, single-region deployments, or unmirrored storage volumes

Aging Components

EOL software, unsupported database versions, and hardware nearing end-of-support windows

Workload Saturation

CPU spikes, memory pressure, and IOPS ceilings that erode reliability under peak load

Security Gaps

Unencrypted connections, over-privileged service accounts, and missing audit logging

CHAPTER 1 — ASSESSMENT & PLANNING

Capacity Planning & Traffic Controls

Observable Capacity Metrics

Effective capacity planning is metric-driven, not intuition-driven. Key signals to track:

- CPU & memory utilization trends (p95, p99)
- Peak transaction rates and concurrency patterns
- Storage growth velocity and IOPS headroom
- Connection pool saturation rates

Goal: maintain reliability headroom without over-provisioning cost.

Traffic Distribution & Failover

Foundational reliability controls using battle-tested tools:

NGINX	HAProxy
Layer 7 load balancing, upstream health checks, and read replica routing	TCP-level proxying, active/passive failover, and backend health monitoring

CHAPTER 2 — CLOUD NATIVE PLATFORMS

Cloud Native Practices Mapped to SRE Outcomes

Cloud native platforms are not just deployment tools — they are reliability primitives that directly support SLO achievement when configured intentionally.

CHAPTER 2 — CLOUD NATIVE PLATFORMS

Kubernetes as a Reliability Engine

1

Container-Based Consistency

Immutable images eliminate environment drift between dev, staging, and production

2

Self-Healing Orchestration

Kubernetes detects pod failures via liveness probes and restarts automatically reducing MTTR

3

Workload-Driven Scaling

HPA and KEDA scale replicas based on CPU, memory, or custom queue-depth signals

4

CI/CD with Rollback Criteria

Automated pipelines enforce quality gates; rollback triggers on error rate breach

CHAPTER 2 — CLOUD NATIVE PLATFORMS

Operational Metrics That Signal Reliability Health

Cloud native practices must be evaluated against DORA-aligned metrics connecting engineering activity to reliability outcomes.

Deployment Frequency

How often you ship validated changes higher frequency with stability signals a mature pipeline

Mean Time to Recovery (MTTR)

Time from incident detection to full service restoration the core SRE recovery metric

Change Failure Rate

Percentage of deployments causing degradation or requiring rollback your quality signal

CHAPTER 3 — DATABASE MIGRATION

The 6 Rs Migration Framework

A structured framework for evaluating and executing database migrations — balancing reliability continuity with modernization goals.

Rehost

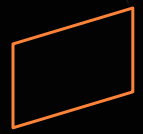
Lift and shift to cloud infrastructure with minimal changes

Replatform

Migrate with minor optimizations, e.g., managed RDS instead of self-hosted

Repurchase

Move to a SaaS database model, replacing self-managed instances

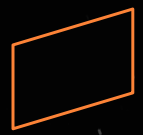


Refactor

Re-architect for cloud native patterns — microservices, event sourcing, CQRS

Retain

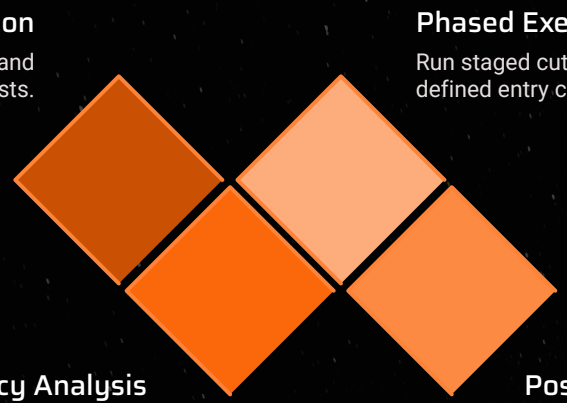
Keep on-premises where latency, compliance, or cost justifies it



Retire

Decommission databases with no active consumers — reduce attack surface

Migration Execution: From Analysis to Optimization



Each phase has defined entry/exit criteria to preserve reliability continuity throughout the migration window.

Phase Highlights

Dependency Analysis

Map all upstream/downstream consumers before touching any schema or connection string

Total Cost Evaluation

Include compute, storage, egress, licensing, and operational overhead — not just instance pricing

Post-Migration Optimization

Validate SLOs, tune query plans, right-size resources, and decommission source environments

CHAPTER 3 — DATABASE MIGRATION

High Availability: Core Reliability Requirements

Redundancy

Multi-AZ and multi-region replicas eliminate single points of failure at the infrastructure layer

Automated Failover

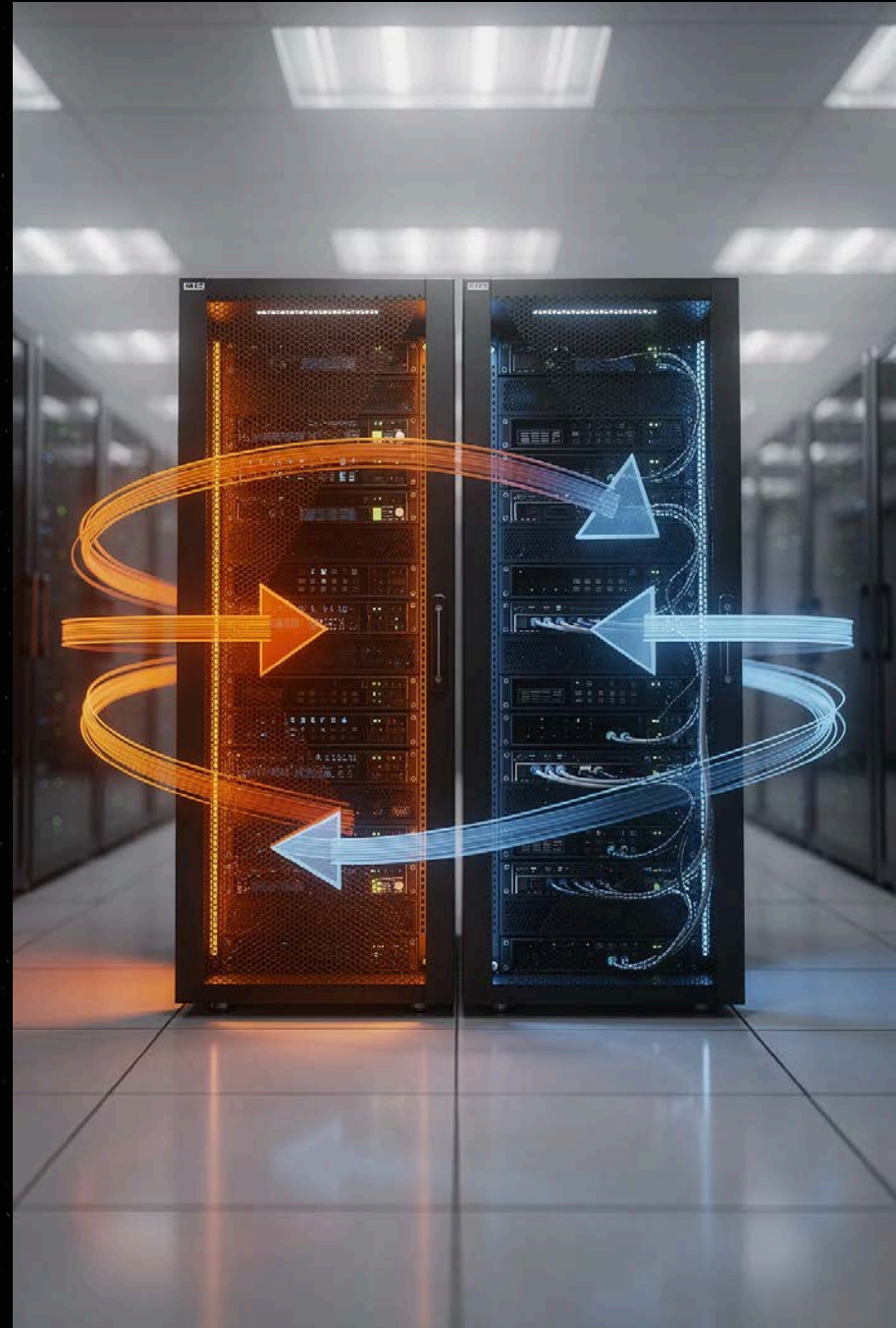
Health-check driven promotion of standby replicas — minimizing human-in-the-loop MTTR

Data Replication

Synchronous replication for zero RPO; asynchronous for geo-distributed read scaling

Resilience Testing

Regular chaos experiments and failover drills validate HA assumptions before incidents do



Key Takeaways

What every SRE and database architect should carry forward from this session:

1

Measure Before You Migrate

Define SLOs, RTO, and RPO first — every architectural decision should trace back to these targets

2

Kubernetes Amplifies Reliability

Self-healing, scaling, and CI/CD rollback criteria are SRE tools — not just developer conveniences

3

The 6 Rs Prevent Costly Mistakes

Not every database needs refactoring — choosing the right migration path protects stability

4

HA Must Be Tested, Not Assumed

Redundancy and failover configurations are only reliable when regularly exercised under realistic conditions

Thank You

Rajesh Kumar Balusu Database Architect, AGAP Technologies Inc

Presented at **Conf42 Site Reliability Engineering (SRE) 2026** — Session: Building Highly Available Databases with Cloud Native Reliability Practices