

1 Some high-level, practical ways to eliminate (or tame) Single Points of Failure:
2
3 Build N+1 redundancy across failure domains
4 Run at least two instances of every critical component (gateway, app,
DB/broker/registry) across different AZs/hosts; use quorum/replication where supported.
5
6 Keep services stateless; externalize state to HA stores
7 Avoid in-memory sessions; use JWT or Spring Session+Redis (clustered). For data, use
replicated databases/queues with clear RPO/RTO and tested restores.
8
9 Health-checked load balancing and automated failover
10 Put a managed L4/L7 LB (or HA pair) in front, with health checks, connection draining,
and multi-target DNS/anycast so traffic shifts away from unhealthy nodes.
11
12 Resilience and graceful degradation in software
13 Enforce timeouts, retries with jittered backoff, circuit breakers, bulkheads, and
backpressure; design idempotent handlers and safe fallbacks to prevent cascades.
14
15 Operate for failure: observe, drill, and automate
16 Centralized logs/traces/metrics with SLOs, runbooks, and alerts; regular chaos/DR
drills; blue-green/canary deploys; infra-as-code and versioned, rollback-able config.