DISASTER RECOVERY & FAILOVER

conduktor

Kafka Disaster Recovery Readiness Checklist

Replication gets your data to the secondary cluster. These six areas determine whether you can actually use it.

Most organizations have invested heavily in Kafka data replication but left critical gaps in the surrounding operational, security, and governance infrastructur

The teams that recover in minutes instead of hours have addressed six technical areas beyond replication, organized around three phases of readiness. This checklist is designed for your next design review, disaster recovery audit, or game day planning session.

Three Phases of Readiness

PHASE 01

Prepare

The work you do before anything breaks determines whether recovery is even possible.



- Define RTO/RPO per domain and workload
- Set up cross-cluster data replication (MirrorMaker, Cluster Linking, etc.)
- Identify Wave 1 applications for priority recovery
- Choose per service: standby cluster (active/passive) or both clusters serving traffic (active/active)
- Map dependencies: apps, topics, schemas, ACLs, certs

PHASE 02

Validate

A disaster recovery plan that hasn't been tested isn't a plan, it's a hypothesis.

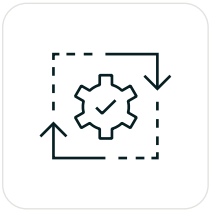


- Run chaos tests in staging, then production game days
- Measure actual failover time during drills
- Test monitoring and decision chain, not just infra
- Verify the runbook is current and executable by on-call, not just the author
- Pre-stage configs in both regions and check for drift

PHASE 03

Execute

Execution should be a calm, sequenced procedure, not improvisation under pressure.



- Switch clients through a stable endpoint, not per-app
- Communicate to stakeholders and compliance teams
- Enforce security and governance during cutover
- Plan failback once stable; reversing the switch needs its own runbook
- Validate Wave 1 apps first, then the long tail

The Readiness Checklist

Six technical areas to verify in your disaster recovery design review.

Security & Identity Parity

- ☐ **TLS and certificate strategy**
  Covers both clusters, including custom CAs and cert renewal processes.
- ☐ **Authentication parity across regions**
  mTLS, SASL, or OAuth configured identically. Cluster-specific credentials pre-provisioned.
- ☐ **ACL/RBAC provisioned on secondary**
  Least-privilege maintained. No broader access on failover than on primary.
- ☐ **Audit logs on secondary**
  Available, retained, and accessible during and after failover for compliance evidence.

Topic & Schema Configuration Parity

- ☐ **Topic configs match across clusters**
  Retention policies, compaction strategies, and quota limits verified in sync.
- ☐ **Schema Registry replicated**
  Schemas and compatibility rules match. Consumers won't fail on schema mismatches after failover.
- ☐ **Configuration drift detection**
  Automated process in place to catch and alert on divergence between primary and secondary.
- ☐ **Consumer offset sync configured**
  Offset translation strategy defined. Reprocessing window understood and acceptable.

Data Protection & Compliance Continuity

- ☐ **Encryption parity verified**
  Field-level and payload-level encryption applied consistently across both clusters.
- ☐ **Data masking rules travel with the data**
  Policies enforced regardless of which cluster serves traffic, not tied to co-located infrastructure.
- ☐ **PII handling consistent across regions**
  Regulated data treated identically. No exposure risk during the failover window.
- ☐ **Region-specific compliance requirements met**
  Secondary site satisfies data residency, sovereignty, and regulatory obligations for all workloads.

Observability & Operational Readiness

- ☐ **Monitoring independent of failed infrastructure**
  Dashboards, alerts, and on-call routing don't depend on the region or cluster that went down.
- ☐ **Replication lag tracked continuously**
  Cross-cluster lag is your effective RPO metric. Alerting thresholds set and tested.
- ☐ **Secondary cluster sized for catch-up load**
  At least 1.5x steady-state capacity to absorb the reconnection burst after failover.
- ☐ **Producer delivery timeouts reviewed**
  delivery.timeout.ms exceeds your expected detection + decision window. Default 2 min is often too short.

Client Switching & Traffic Routing

- ☐ **Client broker discovery method documented**
  How clients find brokers today, and how that changes during failover, is clearly understood.
- ☐ **Failover from a single control point**
  One action switches all clients. No per-app config changes, restarts, or cross-team coordination.
- ☐ **Failover authority defined**
  Clear ownership of who can trigger a switch. Approval chain short enough for a 3 AM incident.
- ☐ **Kafka client versions audited**
  Clients on 3.8+ support rebootstrap (KIP-899). Older clients get stuck and require restarts.

Testing & Continuous Validation

- ☐ **Chaos tests run quarterly at minimum**
  Simulate broker loss, region loss, and auth failures for mission-critical workloads.
- ☐ **Game days include full regional loss**
  Partial failure tests aren't enough. Validate the worst-case scenario your plan is designed for.
- ☐ **Named runbook owner with review cadence**
  Runbooks are versioned, reviewed on a schedule, and updated after every incident or drill.
- ☐ **On-call can execute without the author**
  The person at 3 AM should be able to follow the runbook without the person who wrote it.

Scoring guide: If you can check 20+ items on this list, you're well ahead of most organizations. Fewer than 12 means significant gaps remain. Focus first on the areas where a failure would be unrecoverable: client switching, security parity, and testing

Failover Runbook: A Simple Sequence

No heroics. Simple enough that the person on call at 3 AM can execute it without the person who designed it.

STEP 1	STEP 2	STEP 3	STEP 4	STEP 5
Detect	**Decide**	**Switch**	**Validate**	**Stabilize**
Confirm incident scope. Broker failure, partial, or full regional outage?	Choose recovery mode and approve cutover. Keep the approval chain short.	Route clients via stable endpoint. One action, not dozens.	Wave 1 apps first, then the long tail. Verify end-to-end data flow.	Monitoring, backpressure, comms. Then plan your failback.

How to Know You're Ready

Use these indicators to assess where your disaster recovery program stands today, and where the common blind spots are.

Signs You're on Track

Failover time is measured, not estimated
You've run a drill and have a real number to compare against your RTO target.

Game days run quarterly or more often
Chaos testing is routine, not an annual event. Results are reviewed and runbooks updated afterward.

Failover is a single operational action
Switching traffic doesn't require per-app coordination. One decision, one action, all clients move.

Security posture survives failover
Encryption, RBAC, masking, and audit logging are enforced identically on both clusters.

The on-call engineer can execute the runbook
Anyone on rotation can follow the procedure at 3 AM without calling the person who wrote it.

Failback has its own tested runbook
Reversing the switch is often harder than the initial cutover. It's been planned and validated separately.

Common Pitfalls to Avoid

Treating replication as the entire DR strategy
Data arrives at the secondary but clients can't connect, authenticate, or find the right schemas.

Monitoring that depends on what failed
If dashboards and alerts are in the same region as the primary, you're flying blind during failover.

Failover that requires per-app coordination
If switching 50 services means 50 config changes across 10 teams, your RTO scales with your org chart.

Producer timeouts shorter than your decision window
The default delivery.timeout.ms of 2 minutes means producers silently drop data before anyone decides to fail over.

Configuration drift between clusters
Topic settings, schemas, quotas, and ACLs diverge silently over time. By failover, the secondary behaves differently.

Untested or stale runbooks
If the last drill was a year ago or the author has left, the runbook is a liability, not an asset.

Go deeper:

This checklist is the companion to our full guide on Kafka disaster recovery strategy. Read the full guide

Ready to close the gaps?

Book a 45-minute disaster recovery workshop to review your Kafka estate and build a first-pass wave plan.

Book a Workshop