EN

BCP + Kubernetes SME Guide

Cloud Resilience for Canadian SMEs

RTO/RPO, Velero, ArgoCD, Cloudflare Failover — BOTUM case

Velero + **ArgoCD** + **Cloudflare**

Mars 2026 — Série Cloud Journey B18 (Final)

Introduction: the question nobody asks

Here's the question nobody asks when migrating to the cloud: If our cloud infrastructure goes completely down tonight, how many hours before our customers can resume work? This final guide in the Cloud Journey series answers that question with concrete, production-tested strategies.

RTO and RPO: the two fundamental metrics

RTO (Recovery Time Objective): maximum acceptable time between outage and full recovery. RPO (Recovery Point Objective): maximum tolerable data loss, expressed as time. These two metrics define your DR architecture and its cost.

Application type	Target RTO	Target RPO	Recommended strategy
E-commerce / B2C SaaS	15-30 min	5 min	Active-Active
B2B SaaS / Business apps	1-4 hours	15-30 min	Warm Standby
Internal tools / Back-office	4-24h	1-4h	Pilot Light
Finance / Healthcare	Regulated	Regulated	OSFI E-21 / PHIPA

Cloud DR Strategies: cost vs. RTO matrix

Strategy	RTO	RPO	Monthly Cost	For whom
Cold Backup / Snapshots	4-24 hours	1-24h	5-10% infra	Internal tools, archives
Pilot Light	1-4 hours	15-60min	5-15% infra	Startups, B2B SaaS
Warm Standby	15-60 min	5-15min	20-40% infra	SMEs 50-500 emp., SLAs
Active-Active	< 5 min	< 1 min	80-100% infra	FinTech, eCommerce

Kubernetes for resilience

Auto-healing

If a Pod crashes, K8s restarts it. If a Node goes down, K8s reschedules all its Pods. 'Server crashes' are handled without human intervention, typically in < 2 min.

Rolling deployments

Updates deploy progressively with zero downtime. If a new Pod fails its health check, the rollout pauses automatically.

Health checks

livenessProbe: detects if the app is alive (otherwise → restart). readinessProbe: determines if the Pod is ready (otherwise → removed from LB).

Resource limits

K8s guarantees each Pod a minimum of CPU/RAM (requests) and prevents it from exceeding a maximum (limits). Isolation via cgroups.

High availability architecture

Multi-AZ

Deploy K8s workers across at least 2 AZs. Configuration: NodeAffinity + PodAntiAffinity. Additional cost: near zero.

Load balancing

AWS: ALB + AWS Load Balancer Controller. Azure: Application Gateway. GCP: Cloud Load Balancing. Internal ingress: nginx-ingress or Traefik.

Circuit breaker

Protects against cascading failures. If a dependent service slows down, the circuit breaker 'opens' the connection. Implementations: Istio, Envoy, Resilience4j.

Resilience testing: chaos engineering and fire drills

An untested BCP is fiction. The real question: have we proven this plan works?

- LitmusChaos: open-source tool for K8s — simulates Pod/Node failures, network latency
- Quarterly fire drills: simulate a complete disaster, time the actual recovery
- Runbooks: copyable command lists, stored in git, tested at every fire drill
- Netflix Simian Army (reference): Chaos Monkey, Latency Monkey, Conformity Monkey

Canadian compliance: OSFI E-21, PHIPA, PIPEDA

OSFI E-21

Explicit requirements: RTO/RPO definition, regular recovery testing, critical dependency documentation. Applies to federally regulated financial institutions and their fintech partners.

PHIPA (Ontario)

Health information custodians must maintain continuity plans to ensure health data availability. Data loss not covered by a BCP may constitute a breach.

PIPEDA / Law 25

Notification obligations for security breaches posing real harm risk. A documented and tested BCP demonstrates required due diligence.

Recommended SME stack

Velero

K8s backup: resources + persistent volumes (PVCs). Typical config: hourly backup, 7-day retention, daily snapshot, 30-day retention. Destinations: S3, Azure Blob, GCS.

ArgoCD / Flux (GitOps DR)

DR cluster synced from git. In case of disaster: 1) Switch DNS. 2) ArgoCD has already deployed all apps. 3) Velero restores data.

Cloudflare Load Balancing

Health checks → automatic DNS failover to DR cluster. TTL 30s. Business plan ($200/mo) sufficient for most SMEs. Alternative: Route 53 Health Checks.

Prometheus + Alertmanager

Define SLOs. Alert at 99.9% before the outage hits 99.5%. PagerDuty or OpsGenie for on-call.

Real BOTUM case: RTO achieved in 23 minutes

November 2024: etcd corruption following a poorly orchestrated node update. 2:37 PM: first 503s. 2:41 PM: Alertmanager alert. Here's the complete timeline:

Time	Event	Action
T+0 min	Automatic detection (etcd unhealthy)	Alertmanager → PagerDuty
T+4 min	etcd corruption confirmed via kubectl	On-call engineer
T+8 min	Decision to failover to DR cluster	War room (Slack)
T+10 min	Cloudflare DNS failover triggered (TTL 30s)	Automatic
T+12 min	ArgoCD DR confirms apps deployed + healthy	Automatic (GitOps)
T+18 min	Velero restore of latest data	Runbook script
T+23 min	**100% traffic on DR cluster — services OK**	**Manual validation**

What saved us: DR cluster already synced via ArgoCD (same git repo), Velero snapshot 8 minutes before the incident, runbook tested during September fire drill. Without the BCP: estimated 4-6 hours of complete downtime.

Conclusion: series finale, beginning of resilience

18 posts, a complete cloud journey. Resilience is not a state you achieve — it's a practice you maintain. Start where you are: install Velero this week, write a runbook, schedule a fire drill. Step by step.

Build your cloud BCP with BOTUM

Business continuity planning, resilient architecture, managed Kubernetes — BOTUM teams support you from design to implementation.

→ www.botum.ca/contact