OpenShift Kubernetes Cluster

High Availability & Backup
& Disaster Recovery

Technical Documentation & Best Practices

On-Premise Environment
Version 1.0

February 4, 2026

Table of Contents

Executive Summary

This document provides comprehensive guidance on implementing high availability, backup, and disaster recovery strategies for on-premise OpenShift Kubernetes deployments. These capabilities are essential for ensuring business continuity, minimizing downtime, and protecting against data loss.

The documentation addresses three critical areas:

- **HA Strategy:** Multi-master etcd clustering and application-level high availability
- **Data Protection:** Automated backups with defined RPO and RTO objectives
- **Disaster Recovery:** Failover procedures, geographic redundancy, and DR testing

Implementation of these practices ensures your OpenShift platform can withstand failures while maintaining service continuity and rapid recovery from catastrophic events.

1. HA Strategy

High availability in OpenShift is achieved through redundancy at multiple infrastructure layers. A comprehensive HA strategy eliminates single points of failure and ensures continuous operation during component failures and maintenance activities.

1.1 Multi-Master: etcd Cluster Across Physical Servers

Overview:

The etcd cluster is the critical data store for OpenShift, containing all cluster state and configuration. Implementing a highly available etcd cluster across multiple physical servers ensures control plane resilience. OpenShift requires an odd number of etcd members (typically 3 or 5) to maintain quorum.

Architecture Design Principles:

- Deploy etcd members on dedicated control plane nodes with co-located API servers
- Distribute control plane nodes across physical servers, racks, and availability zones
- Ensure low-latency network connectivity between etcd members (typically less than 10ms RTT)
- Provision dedicated, high-performance SSD storage for etcd with low-latency I/O
- Implement anti-affinity rules to prevent multiple control plane nodes on same infrastructure

etcd Cluster Sizing:

Cluster Size	Failure Tolerance	Use Case
3 members	1 member failure	Standard production deployment for most enterprises
5 members	2 member failures	Enhanced resilience for mission-critical workloads
7+ members	3+ failures	Not recommended due to write latency overhead

etcd Performance Optimization:

- Use SSD storage with sustained IOPS capability of at least 3000 IOPS
- Configure disk quotas (default 8GB) with monitoring to prevent database bloat
- Enable automatic defragmentation to maintain optimal performance
- Monitor etcd metrics including leader elections and disk fsync duration

etcd Health Monitoring Commands:

```
# Check etcd cluster health
oc get etcd -o=jsonpath='{range
.items[0].status.conditions[?(@.type=="EtcdMembersAvailable")]}{.message}{
"\n"}{end}'

# View etcd member list
oc rsh -n openshift-etcd $(oc get pods -n openshift-etcd -l app=etcd -o
name | head -1) \
  etcdctl member list -w table

# Check etcd performance
oc rsh -n openshift-etcd $(oc get pods -n openshift-etcd -l app=etcd -o
name | head -1) \
  etcdctl endpoint status --cluster -w table
```

1.2 Application HA: Pod Anti-Affinity + Multi-Replica Deployments

Overview:

Application-level high availability ensures workloads remain accessible during node failures and maintenance. This is achieved through strategic pod distribution using anti-affinity rules combined with appropriate replica counts.

Multi-Replica Deployment Best Practices:

- Configure minimum 3 replicas for production applications
- Implement pod disruption budgets (PDB) to maintain minimum available replicas
- Use topology spread constraints to distribute pods across failure domains
- Configure liveness and readiness probes for automatic health monitoring

Pod Anti-Affinity Configuration Example:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-application
  namespace: production
spec:
  replicas: 3
  selector:
    matchLabels:
      app: web
  template:
    metadata:
      labels:
        app: web
    spec:
      affinity:
        podAntiAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
          - labelSelector:
              matchExpressions:
              - key: app
                operator: In
                values:
                - web
            topologyKey: kubernetes.io/hostname
      containers:
      - name: web-server
        image: nginx:latest
        resources:
          requests:
            memory: "256Mi"
            cpu: "250m"
          limits:
            memory: "512Mi"
            cpu: "500m"
        livenessProbe:
          httpGet:
            path: /health
            port: 8080
          initialDelaySeconds: 30
          periodSeconds: 10
        readinessProbe:
          httpGet:
            path: /ready
            port: 8080
          initialDelaySeconds: 10
          periodSeconds: 5
```

Pod Disruption Budget Configuration:

```
apiVersion: policy/v1
kind: PodDisruptionBudget
metadata:
  name: web-application-pdb
  namespace: production
spec:
  minAvailable: 2
  selector:
    matchLabels:
      app: web
```

2. Data Protection

Data protection encompasses comprehensive backup strategies and recovery procedures that ensure critical data can be restored following failures. Effective data protection requires automated processes, regular testing, and clear metrics.

2.1 Automated Backup Procedures

Overview:

Automated backup procedures eliminate manual intervention and provide reliable protection for cluster state and application data. OpenShift backup strategies must address both cluster configuration (etcd) and application persistent data.

Backup Components:

- **etcd Database:** Complete cluster state including all Kubernetes objects
- **Persistent Volumes:** Application data using volume snapshots or backup agents
- **Container Images:** Registry contents for deployment and rollback

etcd Backup Automation:

Using CronJob for automated etcd backups:

```
apiVersion: batch/v1
kind: CronJob
metadata:
  name: etcd-backup
  namespace: openshift-etcd
spec:
  schedule: "0 */6 * * *"  # Every 6 hours
  concurrencyPolicy: Forbid
  jobTemplate:
    spec:
      template:
        spec:
          serviceAccountName: etcd-backup
          containers:
          - name: etcd-backup
            image: quay.io/openshift-release-dev/ocp-v4.0-art-dev
            command:
            - /bin/sh
            - -c
            - |
              /usr/local/bin/cluster-backup.sh /mnt/backup
              find /mnt/backup -name 'etcd-backup-*.tar.gz' -mtime +30 -
delete
            volumeMounts:
            - name: backup-storage
              mountPath: /mnt/backup
          restartPolicy: OnFailure
          volumes:
```

```
          - name: backup-storage
            persistentVolumeClaim:
              claimName: etcd-backup-pvc
```

Velero Backup Solution:

Velero is the industry-standard tool for Kubernetes backup and restore, supporting both cluster resources and persistent volumes.

- Install Velero operator from OperatorHub or deploy via Helm
- Configure object storage backend (S3, MinIO, Azure Blob, or GCS)
- Create automated backup schedules for namespaces or entire clusters

Velero Backup Commands:

```
# Create daily backup schedule for production namespace
velero schedule create production-daily \
  --schedule="0 2 * * *" \
  --include-namespaces production \
  --ttl 720h0m0s \
  --snapshot-volumes=true

# Create weekly full cluster backup
velero schedule create cluster-weekly \
  --schedule="0 3 * * 0" \
  --ttl 2160h0m0s \
  --snapshot-volumes=true

# Verify backups
velero backup get
velero backup describe <backup-name>
```

2.2 Recovery Point Objectives (RPO)

Overview:

Recovery Point Objective (RPO) defines the maximum acceptable amount of data loss measured in time. It determines backup frequency and directly impacts business continuity.

RPO Classification:

Tier	RPO Target	Backup Frequency	Use Cases
Mission Critical	0-5 minutes	Continuous/Real-time	Financial transactions, payment processing
Business Critical	15-60 minutes	Every 15-30 minutes	E-commerce, customer portals, order processing
Important	4-24 hours	Daily/Multiple daily	Reporting, analytics, development environments

Achieving RPO Targets:

- **Near-Zero RPO:** Synchronous replication across availability zones
- **Low RPO:** Automated backups every 15-30 minutes with incremental backups
- **Medium RPO:** Daily full backups with appropriate retention policies

2.3 Recovery Time Objectives (RTO)

Overview:

Recovery Time Objective (RTO) defines the maximum acceptable downtime after a disruption. RTO drives infrastructure design decisions including automation level and redundancy requirements.

RTO Classification:

Tier	RTO Target	Recovery Strategy
Mission Critical	< 5 minutes	Active-active deployment, automatic failover
Business Critical	15-60 minutes	Warm standby, automated promotion procedures
Important	4-24 hours	Manual procedures with documented runbooks

Achieving RTO Targets:

- Immediate Recovery: Active-active configurations with global load balancing
- Fast Recovery: Automated restore procedures using Velero
- Standard Recovery: Infrastructure-as-code for rapid cluster reconstruction

3. Disaster Recovery

Disaster recovery encompasses comprehensive planning and procedures required to restore operations following catastrophic events. Effective DR strategies address complete facility loss, regional outages, and cascading failures.

3.1 Failover Procedures

Overview:

Failover procedures define systematic steps to transition operations from a failed primary site to a secondary disaster recovery site. Well-documented and tested procedures are critical for minimizing downtime.

Failover Architecture Options:

- **Active-Passive:** Primary handles all traffic, DR site remains idle until needed
- **Active-Active:** Both sites actively serve traffic with global load balancing

Failover Procedure Steps:

Step	Action	Responsible Team
1. Detection	Monitor alerts trigger failover initiation	Operations/NOC
2. Assessment	Verify primary site failure and scope	Infrastructure Team
3. Decision	Authorize failover activation	Incident Commander
4. Execution	Activate DR site and redirect traffic	Operations Team
5. Verification	Validate application functionality	Application Team
6. Communication	Notify stakeholders of status	Communications Lead

Automated Failover Implementation:

- Configure health checks on primary cluster with automatic DNS updates
- Use global load balancers (F5, AWS Route 53) with failover policies
- Set aggressive health check intervals (30-60 seconds) for rapid detection

3.2 Geographic Redundancy Planning

Overview:

Geographic redundancy protects against regional disasters including natural catastrophes, power outages, and network failures. Effective geographic redundancy requires careful site selection and data replication.

Site Selection Criteria:

- Separate geographic regions with distinct disaster risk profiles
- Minimum 50-100 kilometers separation from primary site
- Low-latency network connectivity (typically < 20ms RTT)
- Independent power grids and network providers

Data Replication Strategies:

- **Synchronous Replication:** Zero data loss, requires low latency
- **Asynchronous Replication:** Better performance, accepts small data loss window

3.3 Disaster Recovery Testing

Overview:

Regular disaster recovery testing validates procedures, identifies gaps, and ensures teams are prepared. Testing frequency should align with business criticality, typically quarterly or semi-annually.

DR Testing Types:

Test Type	Frequency	Description
Tabletop Exercise	Quarterly	Walk through scenarios discussing response actions
Simulation Testing	Semi-annually	Execute procedures in non-production environment
Full DR Test	Annually	Complete failover with actual production traffic

Test Scenarios:

- Complete primary site failure simulating datacenter outage
- Storage system failure requiring restore from backups
- Network connectivity loss between geographic sites
- Ransomware attack scenario with system-wide restoration

Implementation Roadmap

A phased implementation approach manages complexity and ensures operational readiness.

Phase	Timeline	Key Activities
Phase 1: Foundation	Weeks 1-4	etcd HA setup, basic backup automation, monitoring deployment
Phase 2: Application HA	Weeks 5-8	Pod anti-affinity, PDBs, multi-replica deployments
Phase 3: Data Protection	Weeks 9-12	Velero deployment, automated backups, RPO/RTO validation
Phase 4: DR Capabilities	Weeks 13-16	DR site setup, replication, failover procedures
Phase 5: Testing & Ops	Ongoing	Regular DR tests, procedure refinement, training

Tools and Technologies

Category	Tool	Purpose
Backup	Velero	Kubernetes backup and restore
Backup	OADP	OpenShift API for Data Protection
Storage	Ceph/Rook	Distributed storage with replication
Monitoring	Prometheus	Metrics collection and alerting
DR Orchestration	Ansible	Automated failover procedures
Load Balancing	F5/HAProxy	Traffic distribution and failover

Operational Checklist

Daily Operations:

- Monitor backup job completion status
- Review cluster health dashboards
- Verify replication lag within acceptable thresholds

Weekly Operations:

- Perform backup restoration test
- Review and update DR documentation

Monthly Operations:

- Review DR contact lists and escalation procedures
- Test failover automation in non-production

Conclusion

High availability and disaster recovery are foundational pillars of enterprise OpenShift deployments. The strategies outlined in this document provide a comprehensive framework for building resilient infrastructure capable of withstanding failures at multiple levels.

Key Success Factors:

- Redundancy at all infrastructure layers
- Automated backup and recovery procedures
- Clear RPO and RTO objectives aligned with business needs
- Geographic redundancy for regional disaster protection
- Regular testing validating procedures and team readiness

Implementation requires significant investment but delivers reduced downtime, faster recovery, and confidence in platform reliability for mission-critical workloads.

References and Resources

Official Documentation:

- Red Hat OpenShift Backup and Restore: https://docs.openshift.com/container-platform/latest/backup_and_restore/
- Kubernetes High Availability: https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/high-availability/

Tools:

- Velero Documentation: https://velero.io/docs/
- OADP Documentation: https://docs.openshift.com/container-platform/latest/backup_and_restore/application_backup_and_restore/

Standards:

- ISO 22301 Business Continuity Management
- NIST SP 800-34 Contingency Planning Guide

Document Control

Attribute	Value
Version	1.0
Last Updated	February 4, 2026
Document Owner	Infrastructure and Operations Team
Review Cycle	Quarterly or as needed for significant updates
Classification	Internal Use