# CloudForge Kubernetes Management Guide

**Document ID:** PRD-CF-030 **Last Updated:** 2024-03-01 **Owner:** CloudForge Product Team **Classification:** Public

---

## Overview

CloudForge provides fully managed Kubernetes clusters with automated operations, security hardening, and seamless integration with other NovaTech products. This guide covers cluster creation, configuration, and best practices.

---

## Cluster Creation

### Quick Start

```
# Create a basic cluster
cloudforge k8s create \
  --name production-cluster \
  --region us-west-2 \
  --version 1.29 \
  --node-count 3 \
  --node-type m6i.xlarge
```

### Full Configuration

```
# cluster.yaml
apiVersion: cloudforge.novatech.com/v1
kind: KubernetesCluster
metadata:
  name: production-cluster
  environment: production
spec:
  version: "1.29"
  region: us-west-2

  networking:
    vpc:
      cidr: 10.0.0.0/16
```

```
  pod_cidr: 10.244.0.0/16
  service_cidr: 10.96.0.0/12
  network_policy: calico

control_plane:
  high_availability: true
  logging:
    enabled: true
    retention_days: 30

node_pools:
  - name: general
    instance_type: m6i.xlarge
    min_size: 3
    max_size: 10
    disk_size: 100
    labels:
      workload-type: general
    taints: []

  - name: compute
    instance_type: c6i.2xlarge
    min_size: 0
    max_size: 20
    disk_size: 50
    labels:
      workload-type: compute
    taints:
      - key: workload
        value: compute
        effect: NoSchedule

addons:
  - name: aws-load-balancer-controller
  - name: cluster-autoscaler
  - name: metrics-server
  - name: aws-ebs-csi-driver
```

---

## Cluster Versions

### Supported Versions

| Version | Status | Support Until |
|---|---|---|
| 1.29 | Current | March 2025 |
| 1.28 | Supported | December 2024 |
| 1.27 | Supported | September 2024 |
| 1.26 | Deprecated | June 2024 |

### Upgrade Process

```
# Check available upgrades
cloudforge k8s upgrade check --cluster production-cluster

# Plan upgrade
cloudforge k8s upgrade plan \
  --cluster production-cluster \
  --version 1.29

# Execute upgrade
cloudforge k8s upgrade apply \
  --cluster production-cluster \
  --version 1.29 \
  --strategy rolling
```

### Upgrade Strategies

| Strategy | Description | Downtime |
|---|---|---|
| Rolling | Nodes upgraded one at a time | None |
| Blue-Green | New nodes created, traffic shifted | None |
| In-Place | Direct upgrade (dev only) | Brief |

---

## Node Pools

### Adding Node Pools

```
# Add GPU node pool
node_pools:
  - name: gpu
    instance_type: p4d.24xlarge
    min_size: 0
    max_size: 4
```

```
    disk_size: 200
    gpu:
      enabled: true
      driver_version: "535"
    labels:
      nvidia.com/gpu: "true"
    taints:
      - key: nvidia.com/gpu
        value: "true"
        effect: NoSchedule
```

### Spot Instances

```
node_pools:
  - name: spot-workers
    instance_type: m6i.xlarge
    min_size: 0
    max_size: 50
    spot:
      enabled: true
      max_price: "0.10"  # Optional price cap
      interruption_behavior: terminate
    labels:
      lifecycle: spot
    taints:
      - key: lifecycle
        value: spot
        effect: NoSchedule
```

### Auto-Scaling

```
autoscaling:
  enabled: true
  cluster_autoscaler:
    scale_down_delay: 10m
    scale_down_utilization_threshold: 0.5
    skip_nodes_with_local_storage: false
    skip_nodes_with_system_pods: true
    balance_similar_node_groups: true
```

---

## Networking

### Load Balancers

```yaml
# Application Load Balancer
apiVersion: v1
kind: Service
metadata:
  name: web-service
  annotations:
    service.beta.kubernetes.io/aws-load-balancer-type: "external"
    service.beta.kubernetes.io/aws-load-balancer-nlb-target-type: "ip"
    service.beta.kubernetes.io/aws-load-balancer-scheme: "internet-facing"
spec:
  type: LoadBalancer
  ports:
    - port: 443
      targetPort: 8080
  selector:
    app: web
```

### Ingress

```yaml
# Ingress with ALB
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: app-ingress
  annotations:
    kubernetes.io/ingress.class: alb
    alb.ingress.kubernetes.io/scheme: internet-facing
    alb.ingress.kubernetes.io/certificate-arn: arn:aws:acm:...
spec:
  rules:
    - host: app.example.com
      http:
        paths:
          - path: /api
            pathType: Prefix
            backend:
              service:
                name: api-service
                port:
                  number: 80
          - path: /
```

```
            pathType: Prefix
            backend:
              service:
                name: web-service
                port:
                  number: 80
```

### Network Policies

```
# Restrict pod communication
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: api-network-policy
  namespace: production
spec:
  podSelector:
    matchLabels:
      app: api
  policyTypes:
    - Ingress
    - Egress
  ingress:
    - from:
        - podSelector:
            matchLabels:
              app: web
      ports:
        - protocol: TCP
          port: 8080
  egress:
    - to:
        - podSelector:
            matchLabels:
              app: database
      ports:
        - protocol: TCP
          port: 5432
```

---

## Security

### RBAC Configuration

```yaml
# Developer role
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: developer
  namespace: development
rules:
  - apiGroups: [""]
    resources: ["pods", "services", "configmaps"]
    verbs: ["get", "list", "watch", "create", "update", "delete"]
  - apiGroups: ["apps"]
    resources: ["deployments", "replicasets"]
    verbs: ["get", "list", "watch", "create", "update", "delete"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: developer-binding
  namespace: development
subjects:
  - kind: Group
    name: developers
    apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: developer
  apiGroup: rbac.authorization.k8s.io
```

### Pod Security Standards

```yaml
# Namespace with restricted security
apiVersion: v1
kind: Namespace
metadata:
  name: production
  labels:
    pod-security.kubernetes.io/enforce: restricted
    pod-security.kubernetes.io/audit: restricted
    pod-security.kubernetes.io/warn: restricted
```

### SecureVault Integration

```yaml
# Inject secrets from SecureVault
apiVersion: apps/v1
kind: Deployment
metadata:
  name: app
spec:
  template:
    metadata:
      annotations:
        vault.novatech.com/agent-inject: "true"
        vault.novatech.com/role: "app"
        vault.novatech.com/agent-inject-secret-db: "secret/data/db"
    spec:
      containers:
        - name: app
          env:
            - name: DB_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: db-credentials
                  key: password
```

---

## Storage

### Storage Classes

```yaml
# SSD storage class
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: fast-ssd
provisioner: ebs.csi.aws.com
parameters:
  type: gp3
  iops: "10000"
  throughput: "500"
volumeBindingMode: WaitForFirstConsumer
allowVolumeExpansion: true
```

### Persistent Volume Claims

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: data-volume
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: fast-ssd
  resources:
    requests:
      storage: 100Gi
```

### EFS for Shared Storage

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: efs-shared
provisioner: efs.csi.aws.com
parameters:
  provisioningMode: efs-ap
  fileSystemId: fs-xxxxxxxx
  directoryPerms: "700"
```

---

## Observability

### Metrics with DataLens

```yaml
# Enable DataLens integration
observability:
  datalens:
    enabled: true
    metrics:
      scrape_interval: 30s
      retention: 15d
    logs:
      enabled: true
      retention: 7d
    traces:
      enabled: true
      sampling_rate: 0.1
```

### Custom Metrics

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: prometheus-config
data:
  prometheus.yml: |
    scrape_configs:
      - job_name: 'app-metrics'
        kubernetes_sd_configs:
          - role: pod
        relabel_configs:
          - source_labels: [__meta_kubernetes_pod_annotation_prometheus_io_scrape]
            action: keep
            regex: true
```

### Logging

```yaml
# Fluent Bit configuration
logging:
  provider: fluent-bit
  config:
    outputs:
      - name: datalens
        match: "*"
        region: us-west-2
      - name: s3
        match: "*"
        bucket: logs-archive
        region: us-west-2
```

---

## CI/CD Integration

### DevPipeline Integration

```yaml
# .devpipeline/deploy.yaml
stages:
  - build
  - deploy

jobs:
```

```
  build:
    stage: build
    script:
      - docker build -t $IMAGE_NAME:$CI_COMMIT_SHA .
      - docker push $IMAGE_NAME:$CI_COMMIT_SHA

  deploy:
    stage: deploy
    script:
      - cloudforge k8s kubeconfig --cluster production-cluster
      - kubectl set image deployment/app app=$IMAGE_NAME:$CI_COMMIT_SHA
      - kubectl rollout status deployment/app
```

### GitOps with Flux

```
# Enable Flux
gitops:
  flux:
    enabled: true
    source:
      type: git
      url: https://github.com/novatech/k8s-config
      branch: main
    sync:
      interval: 1m
      path: ./clusters/production
```

---

## Cost Management

### Resource Quotas

```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: team-quota
  namespace: team-alpha
spec:
  hard:
    requests.cpu: "100"
    requests.memory: 200Gi
    limits.cpu: "200"
    limits.memory: 400Gi
```

```
    persistentvolumeclaims: "50"
    services.loadbalancers: "5"
```

### Cost Allocation

```
# Tag resources for cost allocation
metadata:
  labels:
    cost-center: engineering
    team: platform
    environment: production
```

---

## Maintenance

### Maintenance Windows

```
maintenance:
  windows:
    - name: weekly
      schedule: "0 4 * * SUN"  # 4 AM Sunday
      duration: 2h
      type: node-updates
    - name: monthly
      schedule: "0 3 1 * *"    # 3 AM 1st of month
      duration: 4h
      type: cluster-updates
```

### Backup and Restore

```
# Backup cluster resources
cloudforge k8s backup create \
  --cluster production-cluster \
  --include-namespaces production,staging \
  --storage s3://backups/k8s/

# Restore from backup
cloudforge k8s backup restore \
  --cluster production-cluster \
  --backup-id backup-20240720 \
  --namespace production
```

---

## Troubleshooting

### Common Issues

| Issue | Solution |
|---|---|
| Nodes not joining | Check security groups, IAM roles |
| Pods pending | Check resource quotas, node capacity |
| DNS not resolving | Verify CoreDNS pods running |
| Load balancer not created | Check AWS LB controller logs |

### Diagnostic Commands

```
# Cluster health
cloudforge k8s health --cluster production-cluster

# Node diagnostics
cloudforge k8s diagnose node \
  --cluster production-cluster \
  --node ip-10-0-1-123

# Network diagnostics
cloudforge k8s diagnose network \
  --cluster production-cluster \
  --namespace production
```

---

*Related Documents: Getting Started (PRD-CF-001), Networking Guide (PRD-CF-040), SecureVault K8s Integration (PRD-SV-030)*