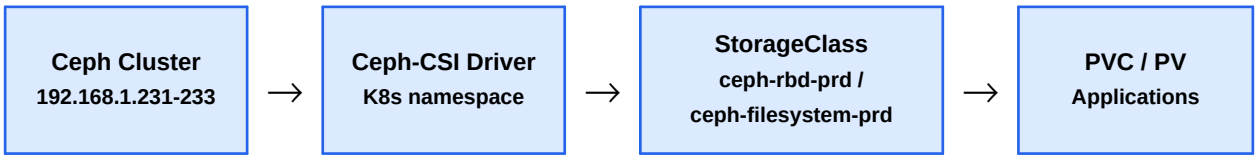
Ceph + Kubernetes

StorageClass Integration Guide

RBD Block Storage (RWO) | CephFS Filesystem (RWO/RWX) | Production Ready

Cluster Information

Cluster ID	`d08e3e42-115e-11f0-a5ad-005056b31542`
MON 1	`192.168.1.231:6789 (INFRA-CEPH-01)`
MON 2	`192.168.1.232:6789 (INFRA-CEPH-02)`
MON 3	`192.168.1.233:6789 (INFRA-CEPH-03)`
Ceph Version	`16.2.15 Pacific (stable)`
OSD	`9 OSDs : 9 up, 9 in — 279 GiB total`
MDS	`myfs (actif) + prd-k8s-fs (actif)`



PARTIE 1 — RBD Block Storage (RWO)

Ceph RBD (RADOS Block Device) fournit un stockage bloc haute performance. Il supporte uniquement le mode **ReadWriteOnce (RWO)** — un seul node à la fois. Idéal pour les bases de données, applications stateful.

STEP 1 — Créer le Pool RBD sur Ceph (INFRA-CEPH-01)

```
# Créer le pool dédié K8s
ceph osd pool create prd-k8S 64 64

# Initialiser pour RBD
rbd pool init prd-k8S

# Vérifier
ceph osd pool ls
```

✓ **Résultat obtenu : pool 'prd-k8S' created**

STEP 2 — Créer l'utilisateur Ceph pour K8s

```
ceph auth get-or-create client.prd-k8S \
mon 'profile rbd' \
osd 'profile rbd pool=prd-k8S' \
mgr 'profile rbd pool=prd-k8S'

# Récupérer la clé
ceph auth get-key client.prd-k8S
# Résultat : AQBc8a5pqH70DhAAysOtN/D1WV0bF48uhl2nNw==

# Clé admin
ceph auth get-key client.admin
# Résultat : AQAJ6e9nOPc0NRAAJ1fw+KheU/jSxY5s58jy4Q==

# Cluster ID
ceph fsid
# Résultat : d08e3e42-115e-11f0-a5ad-005056b31542
```

STEP 3 — Installer Ceph-CSI RBD sur Kubernetes

```
# Ajouter le repo Helm
helm repo add ceph-csi https://ceph.github.io/csi-charts
helm repo update

# Créer le namespace
kubectl create namespace ceph-csi-rbd

# Créer values-ceph-csi.yaml
cat &gt; values-ceph-csi.yaml &lt;&lt;EOF
csiConfig:
- clusterID: "d08e3e42-115e-11f0-a5ad-005056b31542"
monitors:
- "192.168.1.231:6789"
- "192.168.1.232:6789"
- "192.168.1.233:6789"
provisioner:
replicaCount: 2
storageClass:
create: false
secret:
create: false
```

```
EOF

# Installer
helm install ceph-csi-rbd ceph-csi/ceph-csi-rbd \
--namespace ceph-csi-rbd \
--values values-ceph-csi.yaml

# Vérifier tous les pods Running
kubectl get pods -n ceph-csi-rbd -w
```

STEP 4 Créer le Secret Kubernetes pour RBD

```
# ceph-secret.yaml
apiVersion: v1
kind: Secret
metadata:
name: csi-rbd-secret-prd
namespace: ceph-csi-rbd
stringData:
userID: prd-k8S
userKey: AQBc8a5pqH70DhAAysOtN/D1WV0bF48uhl2nNw==

kubectl apply -f ceph-secret.yaml
kubectl get secret -n ceph-csi-rbd
```

STEP 5 Créer la StorageClass RBD

```
# storageclass-ceph-rbd.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
name: ceph-rbd-prd
annotations:
storageclass.kubernetes.io/is-default-class: "false"
provisioner: rbd.csi.ceph.com
parameters:
clusterID: "d08e3e42-115e-11f0-a5ad-005056b31542"
pool: prd-k8S
imageFormat: "2"
imageFeatures: layering
csi.storage.k8s.io/provisioner-secret-name: csi-rbd-secret-prd
csi.storage.k8s.io/provisioner-secret-namespace: ceph-csi-rbd
csi.storage.k8s.io/controller-expand-secret-name: csi-rbd-secret-prd
csi.storage.k8s.io/controller-expand-secret-namespace: ceph-csi-rbd
csi.storage.k8s.io/node-stage-secret-name: csi-rbd-secret-prd
csi.storage.k8s.io/node-stage-secret-namespace: ceph-csi-rbd
reclaimPolicy: Delete
allowVolumeExpansion: true
mountOptions:
- discard

kubectl apply -f storageclass-ceph-rbd.yaml
kubectl get storageclass
```

STEP 6 Tester avec un PVC RWO

```
# test-pvc-prd.yaml
apiVersion: v1
kind: PersistentVolumeClaim
```

```
metadata:
name: test-pvc-ceph-prd
namespace: default
spec:
accessModes:
- ReadWriteOnce
storageClassName: ceph-rbd-prd
resources:
requests:
storage: 1Gi

kubectl apply -f test-pvc-prd.yaml
kubectl get pvc test-pvc-ceph-prd -w
# Résultat : STATUS = Bound ✓
```

✓ PVC Bound confirmé : pvc-dfc90a02-dee5-474a-a33b-f013436327a6 — 1Gi RWO

Troubleshooting RBD — Port Conflit 8081

Si deux drivers CSI sont installés (RBD + CephFS), le port liveness **8081** entre en conflit. Solution : assigner le port **8082** à CephFS.

```
# Vérifier le port occupé
ss -tlnp | grep 8081

# Corriger via helm upgrade
helm upgrade ceph-csi-cephfs ceph-csi/ceph-csi-cephfs \
--namespace ceph-csi-cephfs \
--set nodeplugin.httpMetrics.containerPort=8082
```

PARTIE 2 — CephFS Filesystem (RWO + RWX)

CephFS est un système de fichiers distribué qui supporte à la fois **ReadWriteOnce (RWO)** et **ReadWriteMany (RWX)**. Idéal pour les fichiers partagés, logs, media, configurations.

STEP 1 — Créer les Pools et le Filesystem dédié K8s

```
# Sur INFRA-CEPH-01

# Créer le pool metadata
ceph osd pool create prd-k8s-cephfs-meta 32 32

# Créer le pool data
ceph osd pool create prd-k8s-cephfs-data 64 64

# Créer le nouveau filesystem
ceph fs new prd-k8s-fs prd-k8s-cephfs-meta prd-k8s-cephfs-data

# Vérifier
ceph fs ls
# Résultat :
# name: myfs, metadata pool: cephfs.myfs.meta
# name: prd-k8s-fs, metadata pool: prd-k8s-cephfs-meta
```

STEP 2 — Déployer les Daemons MDS pour prd-k8s-fs

■ **CRITIQUE : Sans MDS actif, le provisioning CephFS échoue avec DeadlineExceeded.**

```
# Vérifier l'état MDS avant déploiement
ceph mds stat
# Résultat initial : prd-k8s-fs:0 ← PROBLÈME, 0 MDS actif

# Déployer les MDS
ceph orch apply mds prd-k8s-fs \
--placement="2 INFRA-CEPH-01 INFRA-CEPH-02 INFRA-CEPH-03"

# Attendre 30 secondes puis vérifier
watch ceph mds stat
# Résultat attendu :
# prd-k8s-fs:1 {prd-k8s-fs:0=prd-k8s-fs.INFRA-CEPH-02.xxx=up:active}

# Vérifier le FS
ceph fs status prd-k8s-fs
# RANK 0 active prd-k8s-fs.INFRA-CEPH-02 ← OK
```

✓ **MDS actif confirmé : prd-k8s-fs:1 up:active + 1 standby**

STEP 3 — Créer le Subvolume Group 'csi'

■ **REQUIS : Sans ce groupe, le provisioner échoue avec "subvolume group 'csi' does not exist".**

```
# Créer le subvolume group pour ceph-csi
ceph fs subvolumegroup create prd-k8s-fs csi

# Vérifier
ceph fs subvolumegroup ls prd-k8s-fs
# Résultat : [{"name": "csi"}]
```

STEP 4 — Créer l'utilisateur CephFS

```
ceph auth get-or-create client.prd-k8s-cephfs \
mon 'allow r' \
mds 'allow rw fsname=prd-k8s-fs' \
osd 'allow rw pool=prd-k8s-cephfs-data, allow rw pool=prd-k8s-cephfs-meta' \
mgr 'allow rw'

# Récupérer la clé
ceph auth get-key client.prd-k8s-cephfs
# Résultat : AQDX9K5pTNIYFhAAFg5FsWot0FEaI+kh0SAapw==
```

STEP 5 Installer Ceph-CSI CephFS sur Kubernetes

```
# Créer le namespace
kubectl create namespace ceph-csi-cephfs

# Installer avec port 8082 (éviter conflit avec RBD sur 8081)
helm install ceph-csi-cephfs ceph-csi/ceph-csi-cephfs \
--namespace ceph-csi-cephfs \
--set csiConfig[0].clusterID="d08e3e42-115e-11f0-a5ad-005056b31542" \
--set csiConfig[0].monitors[0]="192.168.1.231:6789" \
--set csiConfig[0].monitors[1]="192.168.1.232:6789" \
--set csiConfig[0].monitors[2]="192.168.1.233:6789" \
--set nodeplugin.httpMetrics.containerPort=8082

# Vérifier tous les pods 3/3 Running
kubectl get pods -n ceph-csi-cephfs -w
```

✓ Tous les pods 3/3 Running après correction port 8082

STEP 6 Créer le Secret Kubernetes pour CephFS

```
# cephfs-secret-prd.yaml
apiVersion: v1
kind: Secret
metadata:
name: csi-cephfs-secret-prd
namespace: ceph-csi-cephfs
stringData:
userID: prd-k8s-cephfs
userKey: AQDX9K5pTNIYFhAAFg5FsWot0FEaI+kh0SAapw==
adminID: admin
adminKey: AQAJ6e9nOPc0NRAAJ1fw+KheU/jSxY5s58jy4Q==

kubectl apply -f cephfs-secret-prd.yaml
kubectl get secret -n ceph-csi-cephfs
```

STEP 7 Créer la StorageClass CephFS

```
# storageclass-cephfs-prd.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
name: ceph-filesystem-prd
provisioner: cephfs.csi.ceph.com
parameters:
clusterID: "d08e3e42-115e-11f0-a5ad-005056b31542"
fsName: prd-k8s-fs
pool: prd-k8s-cephfs-data
csi.storage.k8s.io/provisioner-secret-name: csi-cephfs-secret-prd
csi.storage.k8s.io/provisioner-secret-namespace: ceph-csi-cephfs
csi.storage.k8s.io/controller-expand-secret-name: csi-cephfs-secret-prd
```

```
csi.storage.k8s.io/controller-expand-secret-namespace: ceph-csi-cephfs
csi.storage.k8s.io/node-stage-secret-name: csi-cephfs-secret-prd
csi.storage.k8s.io/node-stage-secret-namespace: ceph-csi-cephfs
reclaimPolicy: Delete
allowVolumeExpansion: true

kubectl apply -f storageclass-cephfs-prd.yaml
kubectl get storageclass
```

STEP 8 Tester PVC RWO et RWX

```
# Test RWO
cat &lt;&lt;EOF | kubectl apply -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
name: test-cephfs-rwo
spec:
accessModes:
- ReadWriteOnce
storageClassName: ceph-filesystem-prd
resources:
requests:
storage: 2Gi
EOF

# Test RWX
cat &lt;&lt;EOF | kubectl apply -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
name: test-cephfs-rwx
spec:
accessModes:
- ReadWriteMany
storageClassName: ceph-filesystem-prd
resources:
requests:
storage: 2Gi
EOF

# Vérifier les 2
kubectl get pvc | grep cephfs
```

PARTIE 3 — Troubleshooting & Commandes Utiles

Erreurs rencontrées et solutions

Erreur	Cause	Solution
`Port 8081 already in use`	RBD occupe le port 8081	`CephFS → port 8082 via helm upgrade`
`DeadlineExceeded`	MDS non actif pour prd-k8s-fs	`ceph orch apply mds prd-k8s-fs`
`subvolume group 'csi' does not exist`	Groupe manquant dans CephFS	`ceph fs subvolumegroup create prd-k8s-fs csi`
`operation already exists`	PVC bloqué en cache provisioner	`kubectl delete pvc + rollout restart provisioner`

Commandes de diagnostic rapide

```
# ■■ CEPH ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
ceph status # Santé globale
ceph mds stat # État MDS
ceph fs status prd-k8s-fs # État filesystem
ceph fs subvolumegroup ls prd-k8s-fs # Subvolume groups
rbd ls prd-k8S # Images RBD créées
ceph osd pool ls # Liste des pools

# ■■ KUBERNETES ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
kubectl get storageclass # StorageClasses
kubectl get pvc -A # Tous les PVC
kubectl get pv # Tous les PV
kubectl get pods -n ceph-csi-rbd # Pods RBD
kubectl get pods -n ceph-csi-cephfs # Pods CephFS

# ■■ LOGS ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
kubectl logs -n ceph-csi-rbd \
&lt;provisioner-pod&gt; -c csi-rbdplugin --tail=50

kubectl logs -n ceph-csi-cephfs \
&lt;provisioner-pod&gt; -c csi-cephfsplugin --tail=50

# ■■ DEBUG PVC ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
kubectl describe pvc &lt;pvc-name&gt; # Events du PVC
kubectl rollout restart deployment \
ceph-csi-cephfs-provisioner -n ceph-csi-cephfs
```

Récapitulatif Final — StorageClasses en Production

StorageClass	Provisioner	Modes	Usage
ceph-rbd-prd	rbd.csi.ceph.com	RWO	DB, Postgres, Redis, apps stateful
ceph-filesystem-prd	cephfs.csi.ceph.com	RWO + RWX	Fichiers partagés, logs, media

Checklist Connectivité

✓	Ports MON ouverts (6789) des workers K8s vers 192.168.1.231-233
✓	Ports OSD ouverts (6800-7300) des workers K8s vers le cluster Ceph
✓	Module kernel RBD chargé sur tous les workers (modprobe rbd)
✓	Module kernel CephFS chargé si besoin (modprobe ceph)
✓	MDS actif pour prd-k8s-fs (ceph mds stat)
✓	Subvolume group 'csi' créé (ceph fs subvolumegroup ls)