



Veille Technologique

Technique

8 octobre 2026 – Fréquence : Bi-hebdomadaire (Mardi et Jeudi)

Synthèse & Sujet du Jour

OpenShift Data Foundation (Rook-Ceph) : stockage défini par logiciel, CSI et chiffrement par tenant

Le stockage est la couche la plus sous-estimée du Platform Engineering. ODF industrialise Rook-Ceph sur Kubernetes, expose des StorageClasses CSI (ceph-rbd, cephfs) et autorise un chiffrement par-PV avec HashiCorp Vault comme KMS : une valeur de clé par tenant, une frontière cryptographique par namespace, un socle souverain pour l'IA d'entreprise.

1. Périmètre

Famille de veille : 2. Métiers, Compétences & Technologies (Indicateur Qualiopi 24).

Fréquence d'exécution : Bi-hebdomadaire (Mardi et Jeudi à 08h30).

Diffusion : [Radar Web Aperta \(FR\)](#) — [LinkedIn Aperta Scientia](#) — [X @ApertaScientia](#).

2. Sujet du Jour — Article Publié sur le Radar Web

Sur une plateforme Kubernetes, on dimensionne le CPU, on plafonne la mémoire, on surveille la latence réseau — et on découvre les vrais problèmes le jour où un PersistentVolumeClaim reste en **Pending**. Le stockage est la couche la plus sous-estimée du Platform Engineering : il décide de la résilience, de la portabilité et, de plus en plus, de la conformité. Sur OpenShift, cette couche porte un nom : **OpenShift Data Foundation (ODF)**, l'opérateur qui transforme des disques bruts en un cluster **Ceph** consommable par l'API Kubernetes.

1. ODF : un cluster Ceph gouverné par un operator

ODF n'invente pas un stockage : il industrialise **Rook**, l'operator qui déploie et réconcilie **Ceph** sur Kubernetes (voir la [documentation Rook Ceph](#)). Trois familles de démons composent le cœur :

- **MON (Monitor)** — garde le *cluster map* (topologie, état des OSD, placement groups) ; le quorum des MON est le point de décision unique du cluster.
- **MGR (Manager)** — expose les métriques, le dashboard et les modules d'administration.
- **OSD (Object Storage Daemon)** — un processus par disque : c'est lui qui stocke réellement les données, réplique et répare.

Là où un SAN classique demande un administrateur stockage dédié, ODF offre des **CRDs Kubernetes** : on déclare un **StorageCluster**, Rook fait converger l'état réel. C'est le même contrat déclaratif que le reste de la plateforme — et donc le même pipeline GitOps.

2. Un seul cluster, trois interfaces de consommation



Le stockage Ceph est exposé au cluster via des **drivers CSI** qui provisionnent dynamiquement les volumes :

- **ocs-storagecluster-ceph-rbd** — volumes **bloc RBD** (RADOS Block Device), accès **RWO** (ReadWriteOnce). Le choix par défaut pour les bases de données, les états applicatifs et les *StatefulSets*.
- **ocs-storagecluster-cephfs** — **CephFS**, système de fichiers partagé en accès **RWX** (ReadWriteMany). Indispensable pour les volumes partagés par plusieurs Pods (artefacts de build, modèles, pipelines).
- **S3 / Object Storage** via **NooBaa (Multicloud Object Gateway)** — une API S3 compatible pour les sauvegardes, les *data lakes* et les backends d'entraînement.

Vu de la plateforme, ce sont des **StorageClasses** standard : un PVC + un nom de classe, et CSI provisionne le volume. C'est cette abstraction qui rend le stockage portable — et testable en CI comme n'importe quel manifeste.

3. Chiffrement : deux granularités, une seule clé de décision

C'est ici qu'ODF dépasse le SAN boîte-noire. Le chiffrement Ceph repose sur une **enveloppe à deux clés** : chaque volume possède une **DEK** (Data Encryption Key) qui chiffre les données, elle-même chiffrée par une **KEK** (Key Encryption Key) détenue par un **KMS externe**. ODF s'intègre à **HashiCorp Vault** comme KMS. Deux niveaux coexistent :

- **Chiffrement cluster-wide** — toutes les données au repos sont chiffrées, une seule politique. C'est le socle, décrit pas-à-pas dans l'article [OpenShift Data Foundation and HashiCorp Vault securing data](#) du **Red Hat Developer Blog**.
- **Chiffrement par-PV (StorageClass)** — souvent **insuffisant** en environnement multi-tenant : une conformité peut exiger **une clé distincte par projet, tenant ou niveau de sensibilité**.

4. Le cas multi-tenant : isoler les frontières de chiffrement par namespace

L'article [Enhance data security in OpenShift Data Foundation](#) (Red Hat Developer Blog, 15 juillet 2025) déroule exactement ce scénario. En activant **StorageClass encryption** dans l'assistant d'installation ODF, le cluster obtient une **StorageClass chiffrée supplémentaire** :

```
$ oc get sc
NAME                                     PROVISIONER                                RECLAIMPOLICY
ocs-storagecluster-ceph-rbd             openshift-storage.rbd.csi.ceph.com Delete
ocs-storagecluster-ceph-rbd-encrypted (default) openshift-storage.rbd.csi.ceph.com Delete
ocs-storagecluster-cephfs               openshift-storage.cephfs.csi.ceph.com Delete
```

L'isolation se construit en quatre gestes, chacun mappé sur une primitive Kubernetes :

1. **Un rôle et une ACL Vault par tenant** — chaque namespace obtient son propre chemin (**kv/tenant-a/**), donc sa propre frontière cryptographique.
2. **Un ConfigMap dans le namespace** — surcharge la configuration CSI pour pointer le volume vers le bon chemin Vault.
3. **Un ServiceAccount dédié**, lié au rôle Vault via **Kubernetes Auth** (**bound_service_account_namespaces**).
4. **Des PVC dans le namespace** consommant la StorageClass chiffrée — chaque volume est scellé par la KEK du tenant.

Point d'ingénierie souvent raté : **Kubernetes déconseille les tokens de ServiceAccount longue durée** (depuis



v1.22) ; le bon réflexe est un **TTL court** dans la policy Vault, avec rotation automatique via la **TokenRequest API**. Le secret ne vit plus dans un Secret éternel, il vit dans le temps.

5. Pourquoi c'est structurant pour les plateformes IA (AS300)

Le stockage n'est pas un sujet *DevOps* périphérique : c'est le substrat de l'IA d'entreprise.

- **Entraînement distribué** — les datasets doivent être montés en **RWX** par plusieurs workers ; CephFS devient le point de rendez-vous des jobs.
- **RAG et bases vectorielles** — un backend comme **Milvus** s'appuie sur des volumes bloc **RBD** pour ses index persistants, avec snapshots CSI.
- **Model store & registre** — les artefacts de modèles et les checkpoints sont des volumes *stateful* : **snapshots et clones CSI** permettent de versionner un état sans copie complète.
- **Souveraineté** — chiffrement at-rest à clés maîtrisées (**KMS Vault**) et S3 interne (**NooBaa**) : les données d'entraînement ne quittent pas le périmètre.

3. Impacts Pédagogiques & Cursus Aperta Scientia

- **Cursus AS200 (DevOps Platform Engineer — 399h)** : module stockage cloud-native — architecture Rook-Ceph (MON/MGR/OSD, placement groups, réplication), drivers CSI et provisionnement dynamique, StorageClasses RBD (RWO) vs CephFS (RWX), snapshots/clones CSI, déploiement déclaratif du **StorageCluster** par OpenShift GitOps (ArgoCD), étiquetage des PV et politiques d'accès.
- **Cursus AS300 (AI Platform Engineer — 399h)** : stockage des plateformes IA sur Red Hat OpenShift AI — montage RWX des datasets pour l'entraînement distribué, backend RBD de persistance pour les bases vectorielles (Milvus/Qdrant) des pipelines RAG, model store et checkpoints versionnés par snapshots CSI, chiffrement at-rest par tenant avec KMS Vault pour la conformité et la souveraineté des données.
- **Compétence transverse validée** : raisonner une politique de chiffrement (cluster-wide vs par-PV), définir la frontière cryptographique d'un tenant et l'automatiser par GitOps.

4. Sources & Références Techniques

- **Red Hat Developer Blog** — *Enhance data security in OpenShift Data Foundation* : <https://developers.redhat.com/articles/2025/07/15/enhance-data-security-openshift-data-foundation> (consulté le 2026-10-08)
- **Red Hat Developer Blog** — *OpenShift Data Foundation and HashiCorp Vault securing data* : <https://developers.redhat.com/articles/2025/04/10/openshift-data-foundation-cluster-wide-encryption-hashicorp-vault> (consulté le 2026-10-08)
- **Red Hat** — *OpenShift Data Foundation (page produit)* : <https://www.redhat.com/en/technologies/cloud-computing/openshift-data-foundation>
- **Red Hat Documentation** — *OpenShift Data Foundation 4.22* : https://docs.redhat.com/en/documentation/red_hat_openshift_data_foundation
- **Rook** — *Rook Ceph Documentation* : <https://rook.io/docs/rook/latest-release/Getting-Started/intro/>



- Ceph — *Ceph Documentation* : <https://docs.ceph.com/en/latest/>
- Kubernetes CSI — *CSI Developer Documentation* : <https://kubernetes-csi.github.io/docs/>



Contacts & Communauté :

- Direction des Formations : learning@apertascientia.io
- Référent Handicap & Bien-être (Care) : care@apertascientia.io
- X (Twitter) : [@ApertaScientia](https://twitter.com/ApertaScientia)
- Communauté Discord : discord.gg/eDJzR5Vk85