Red Hat Enterprise Linux

THALES - MDS

Workshop - Storage OpenShift

Luigi Colagiorgio
Architect

Red Hat

Document Information

Originator

Red Hat Consulting

Owner

Red Hat Consulting – Confidential; Restricted Distribution

Distribution

Do not forward or copy without written permission.

Confidentiality

This is a confidential document between Red Hat, Inc. and Customer (“Client”).

All information supplied to the Client for the purpose of this project is to be considered Red Hat confidential.

Disclaimer

This document is not a quote, and does not represent an official Statement of Work. If acceptable, a formal quote can be issued, which will include a contract with the scope of work, cost, and any Client requirements if necessary.

Contacts Red Hat

Name	Role	Mail	Phone
Luigi Colagiorgio	Architect OpenShift	lcolagio@redhat.com	07 70 21 98 73

Red Hat

Contacts THALES MDS

Contacts présents au workshop

Name	Role	Mail	Mobile
Sylvie GALLOUX	Scrum Master	sylvie.galloux@thalesgroup.com	
Boris HOUEE	Product Owner	boris.houee@thalesgroup.com	
Pierre NOVAT	Architecte MDS	pierre.novat@thalesgroup.com	
Stephane SOLIER	Architecte Sécurité	stephane.solier@thalesgroup.com	
Yann LOPEZ	Chef de Projet	yann.lopez@thalesgroup.com	
Jean-Loic MAUDUY	Architecte Logiciel	jean-loic.mauduy@thalesgroup.com	

Rappel des Demandes / Questions

- Configuration du stockage local avec l'opérateur Local Storage?
- Quid du framework CSI ?
- Quid du stockage avec Vmware ?
- Instroduction à OCS4 ...

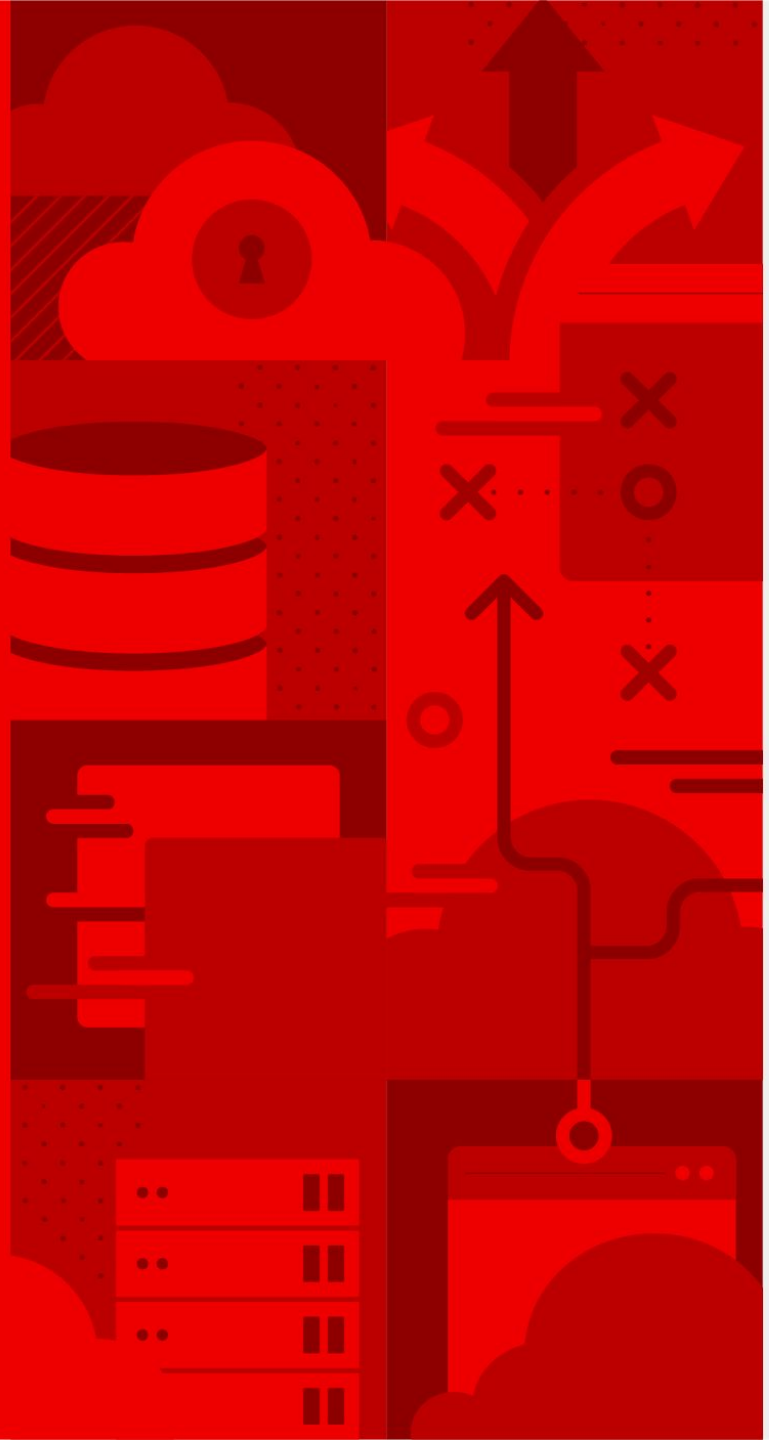
Goals and Agenda

Goals

- Comprendre / passer en revue les fonctionnalités du stockage Openshift:
  CSI, Local Storage, Vmware ...

Agenda

- Stockage Persistent
- Option de Stockage Persistent
- OpenShift CSI
- CSI driver Vmware
- Local Storage
  - Local Storage Principes
  - Local Storage Installation
  - Local Storage Use Case
  - Bug Fixes



Stockage Persistent

Red Hat

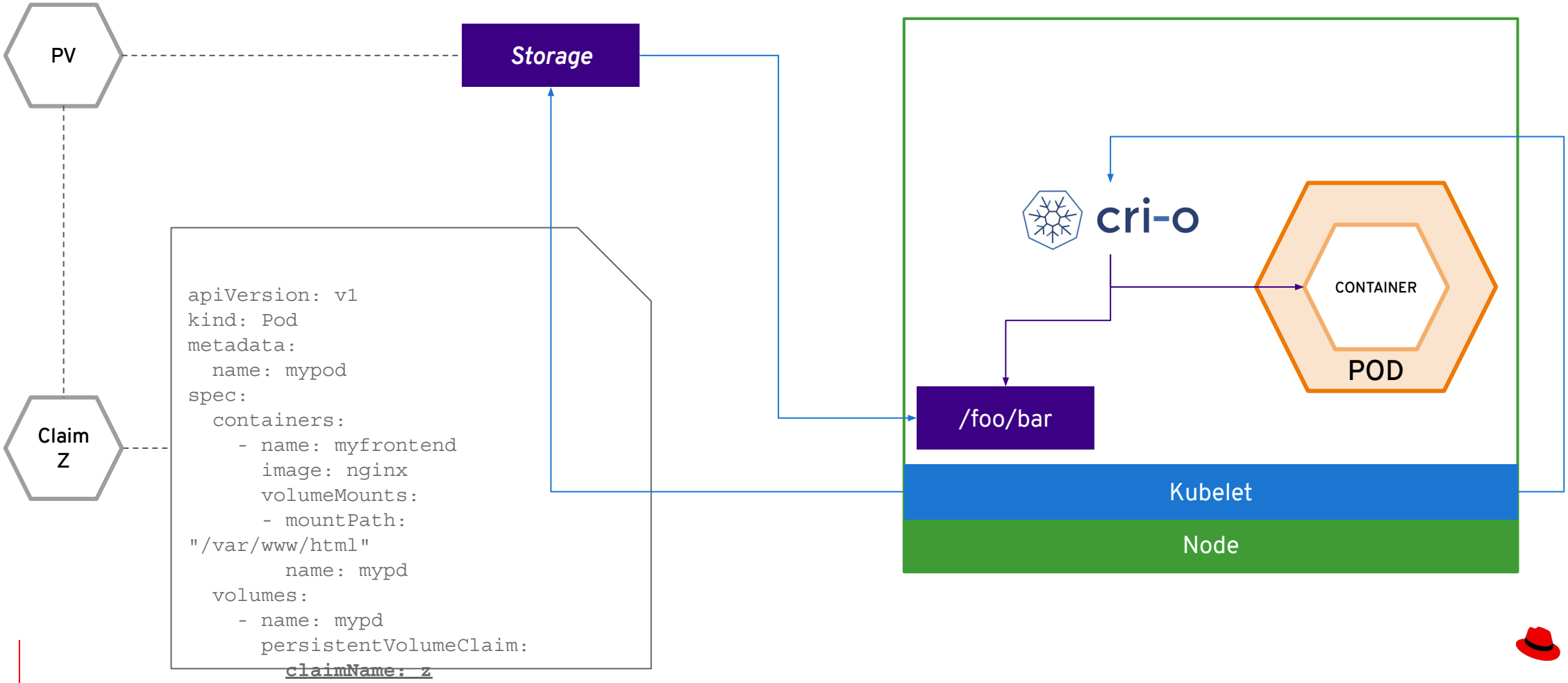
Stockage Persistent

A broad spectrum of
static and dynamic storage endpoints

NFS	OpenStack Cinder	iSCSI	Azure Disk	AWS EBS	FlexVolume
GlusterFS	Ceph RBD	Fiber Channel	Azure File	GCE Persistent Disk	VMWare vSphere VMDK
		NetApp Trident*	Container Storage Interface (CSI)**		

Stockage Persistent

Consommation d'un PV



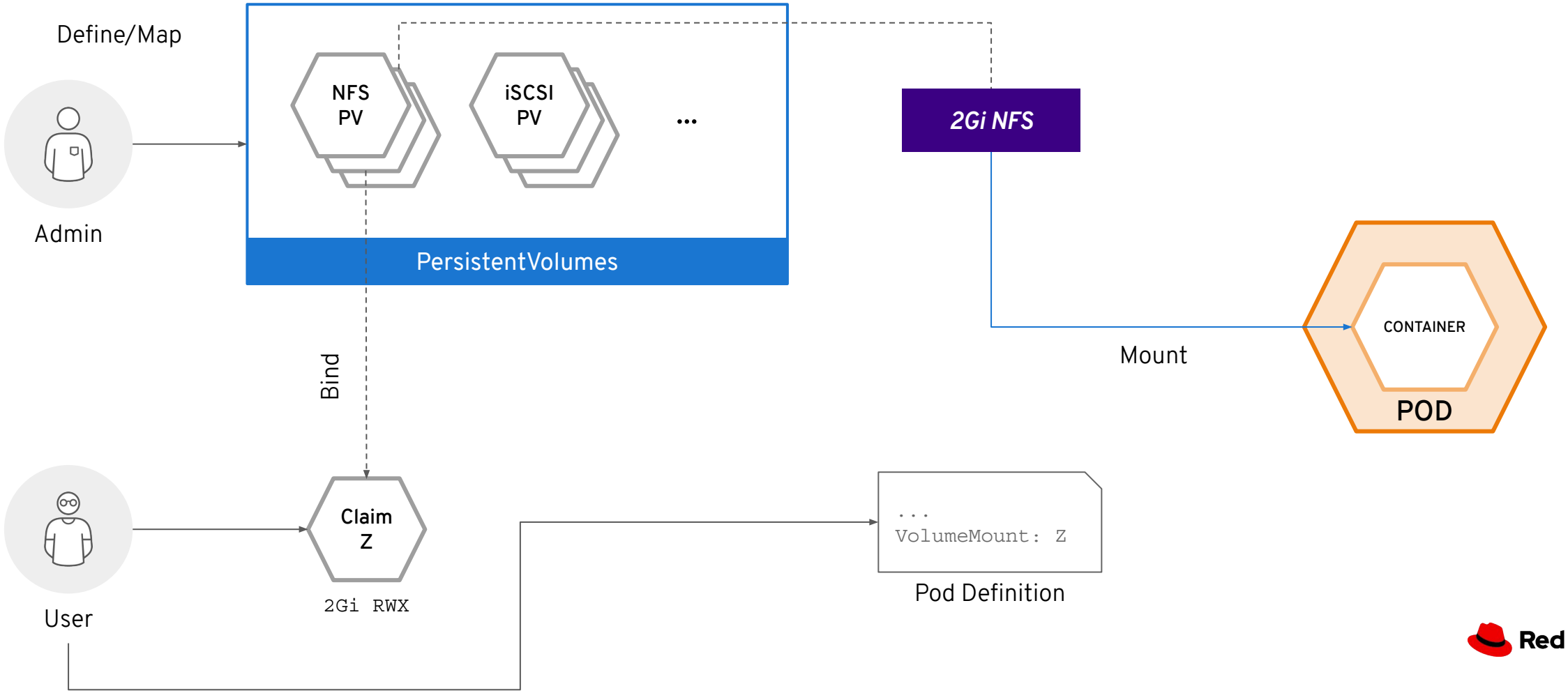
Stockage Persistent

Consommation d'un PV

- Un **PersistentVolumeClaim (PVC)** est liée à un **volume persistant (PV),** et un volume persistant correspond à un stockage réel
- Un utilisateur définit un workload qui demande à OpenShift d'attacher le stockage PVC à un point de montage à l'intérieur du conteneur
- Lorsque le master dit au kubelet d'exécuter le pods, il dit d'abord à l'hôte de monter le stockage au niveau du système d'exploitation en toute sécurité pour le conteneur
- Ensuite, le Kubelet lance les conteneurs et demande au CRI-O de fixer le point de montage de l'OS comme un volume à l'intérieur du conteneur

Stockage Persistent

Provisionning Statique du Storage



Red Hat

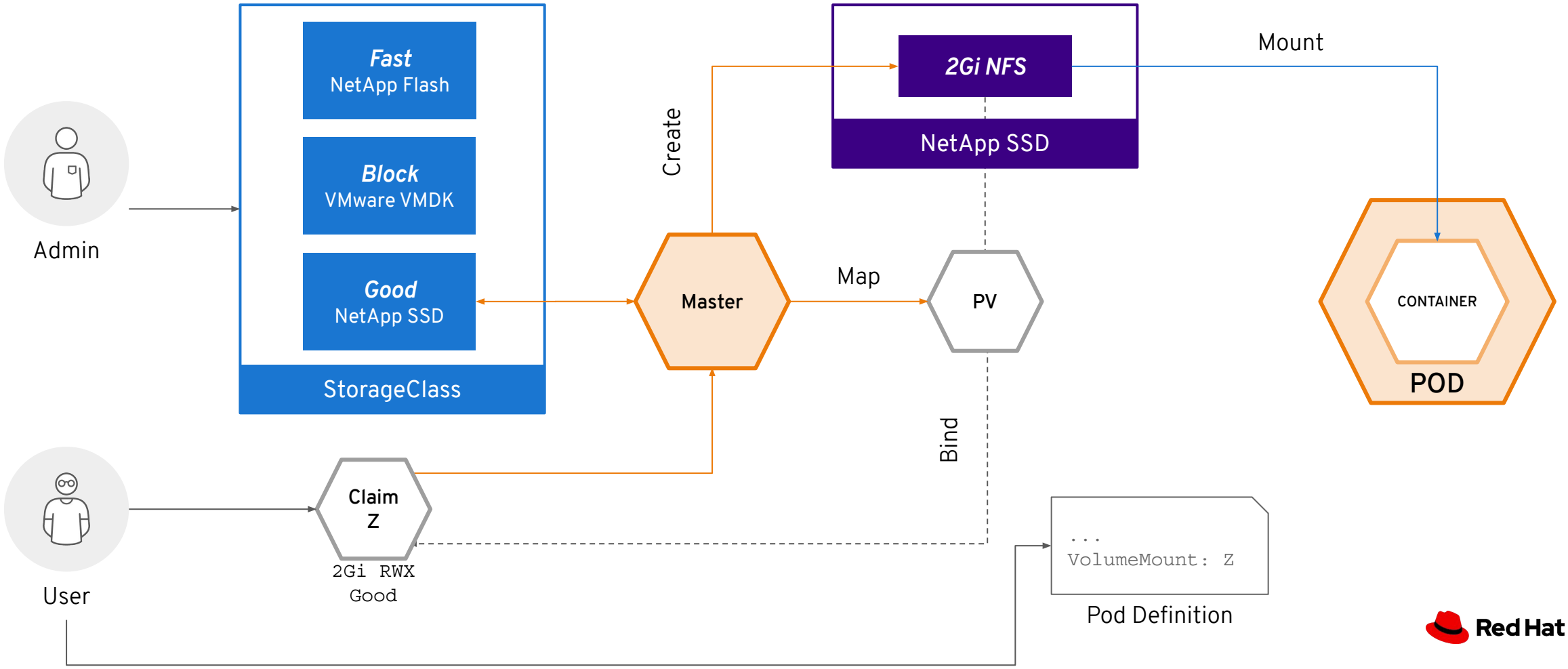
Stockage Persistent

Provisionning Statique du Storage

- Les **administrateurs** définissent **un pool de volumes persistants (PV)** qui sont gérés par des solutions de stockage en réseau comme NFS, iSCSO, AWS EBS, etc. et les rendent disponibles globalement dans le cluster OpenShift.
- Les **utilisateurs** peuvent, dans le contexte de leurs projets / namespace, créer une demande de volume persistant (PVC) afin de demander qu'un volume persistant soit disponible dans leurs pods. Dans la définition du pod, un développeur peut se référer à la PVC et monter le volume persistant demandé à l'intérieur du pod selon un chemin arbitraire.
- Si un pod est **redémarré**, OpenShift monte à nouveau le même volume persistant dans le pod afin que les données du pod soient disponibles. Les PVC survivent aux conteneurs qui les utilisaient.

Stockage Persistent

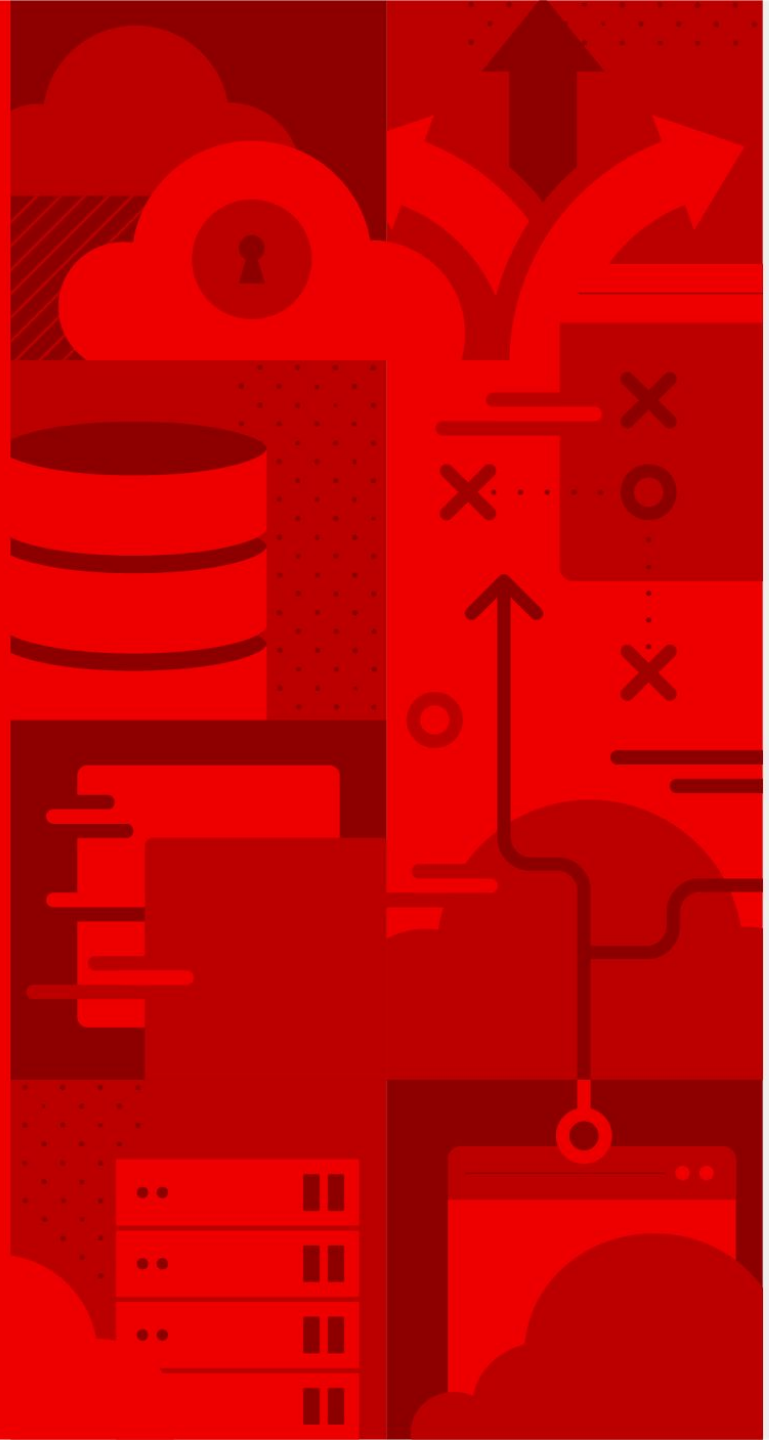
Provisioning Dynamic du Storage



Stockage Persistent

Provisioning Dynamic du Storage

- Le **Dynamic provisioning** permet de provisionner des volumes persistants à la demande lorsque les **utilisateurs le demandent** plutôt que de les **prédéfinir** à l'avance par les administrateurs
- **StorageClass** est un modèle de fourniture de volumes persistants sur un réseau de stockage. OpenShift fournit un ensemble de **provisionneurs** qui déterminent quels **plugins** de volume doivent être utilisés pour provisionner les volumes. OpenShift prend également en charge des plugins tiers qui ne font pas partie de Kubernetes, tels que NetApp Trident
- L'administrateur crée un catalogue des **StorageClasses** disponibles dans le cluster OpenShift. Les noms des StorageClass sont des noms arbitraires pour communiquer leurs caractéristiques
- Les **utilisateurs** peuvent créer une demande de volume persistant et spécifier le nom d'une **StorageClass** pour indiquer à OpenShift le type de volume persistant qui doit être provisionné pour eux



Option de Stockage Persistent

Presenter's Name
Title

Red Hat

Option de Stockage Persistent

- Usage suivant le type de storage

Table 1. Available storage options

Storage type	Description	Examples
Block	• Presented to the operating system (OS) as a block device • Suitable for applications that need full control of storage and operate at a low level on files bypassing the file system • Also referred to as a Storage Area Network (SAN) • Non-shareable, which means that only one client at a time can mount an endpoint of this type	AWS EBS and VMware vSphere support dynamic persistent volume (PV) provisioning natively in OpenShift Container Platform.
File	• Presented to the OS as a file system export to be mounted • Also referred to as Network Attached Storage (NAS) • Concurrency, latency, file locking mechanisms, and other capabilities vary widely between protocols, implementations, vendors, and scales.	RHEL NFS, NetApp NFS [1], and Vendor NFS
Object	• Accessible through a REST API endpoint • Configurable for use in the OpenShift Container Platform Registry • Applications must build their drivers into the application and/or container.	AWS S3

https://docs.openshift.com/container-platform/4.4/scalability_and_performance/optimizing-storage.html#available-persistent-storage-options_persistent-storage

Option de Stockage Persistent

- Storage type technologie recommandé suivant le type d'accès

Table 2. Recommended and configurable storage technology

Storage type	ROX [2]	RWX [3]	Registry	Scaled registry	Metrics [4]	Logging	Apps
Block	Yes [5]	No	Configurable	Not configurable	Recommended	Recommended	Recommended
File	Yes [5]	Yes	Configurable	Configurable	Configurable [6]	Configurable [7]	Recommended
Object	Yes	Yes	Recommended	Recommended	Not configurable	Not configurable	Not configurable [8]

https://docs.openshift.com/container-platform/4.4/scalability_and_performance/optimizing-storage.html#recommended-configurable-storage-technology_persistent-storage

Option de Stockage Persistent

- Type d'accès supporté suivant les plug-ins

Table 2. Supported access modes for PVs

Volume Plug-in	ReadWriteOnce	ReadOnlyMany	ReadWriteMany
AWS EBS	✅	-	-
Azure File	✅	✅	✅
Azure Disk	✅	-	-
Cinder	✅	-	-
Fibre Channel	✅	✅	-
GCE Persistent Disk	✅	-	-
HostPath	✅	-	-
iSCSI	✅	✅	-
Local volume	✅	-	-
NFS	✅	✅	✅
Red Hat OpenShift Container Storage See Available dynamic provisioning plug-ins for more information.	`ceph-rbd`	-	`ceph-fs`
VMware vSphere	✅	-	-

https://docs.openshift.com/container-platform/4.2/storage/understanding-persistent-storage.html#pv-access-modes_understanding-persistent-storage

Option de Stockage Persistent

- iSCSI et Fibre Channel n'ont pas de mécanisme de fencing

Table 1. Access modes

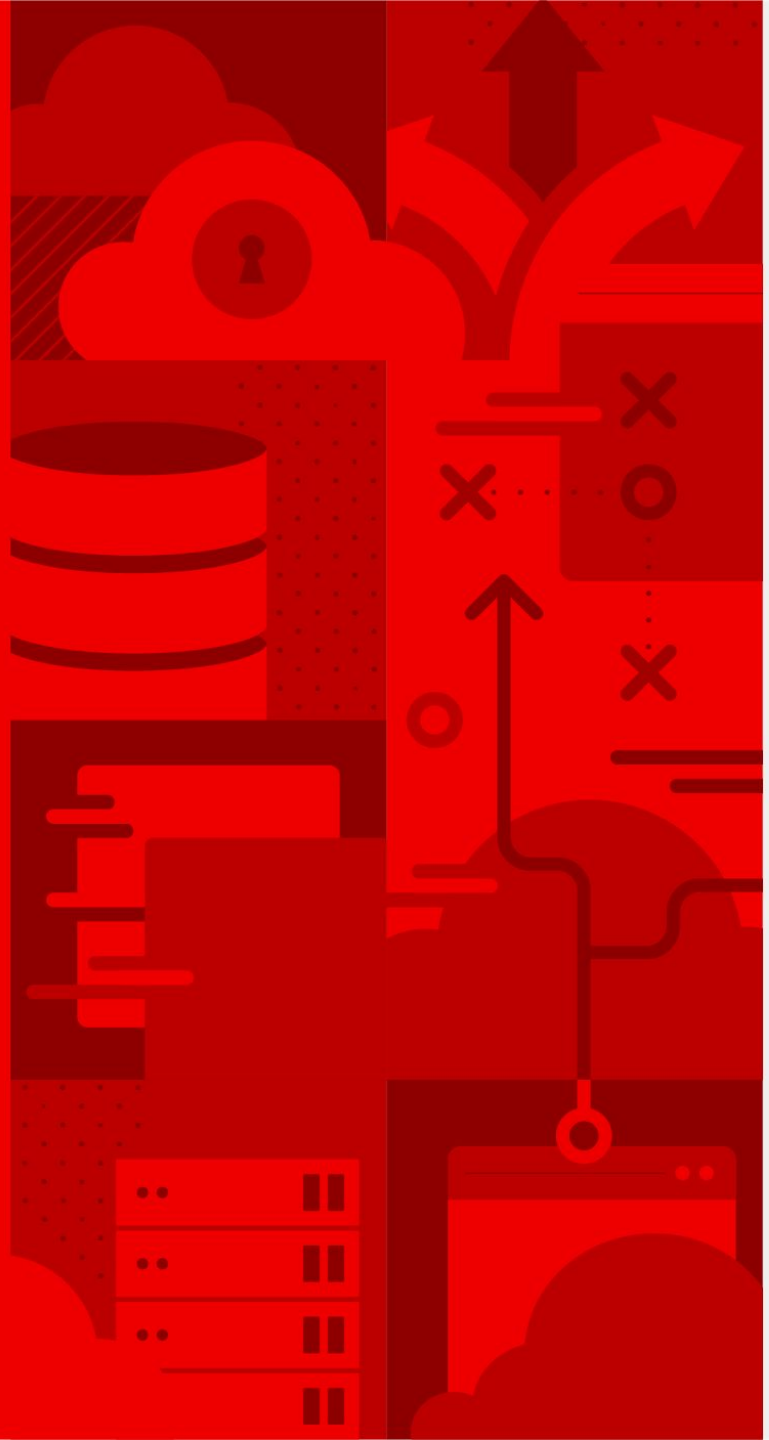
Access Mode	CLI abbreviation	Description
ReadWriteOnce	`RWO`	The volume can be mounted as read-write by a single node.
ReadOnlyMany	`ROX`	The volume can be mounted as read-only by many nodes.
ReadWriteMany	`RWX`	The volume can be mounted as read-write by many nodes.

A volume's `AccessModes` are descriptors of the volume's capabilities. They are not enforced constraints. The storage provider is responsible for runtime errors resulting from invalid use of the resource.

For example, NFS offers **ReadWriteOnce** access mode. You must mark the claims as `read-only` if you want to use the volume's ROX capability. Errors in the provider show up at runtime as mount errors.

iSCSI and Fibre Channel volumes do not currently have any fencing mechanisms. You must ensure the volumes are only used by one node at a time. In certain situations, such as draining a node, the volumes can be used simultaneously by two nodes. Before draining the node, first ensure the Pods that use these volumes are deleted.

https://docs.openshift.com/container-platform/4.4/storage/understanding-persistent-storage.html



OPENSHIFT CSI

OpenShift Container Storage Interface

Presenter's Name
Title

Red Hat

OPENSHIFT CSI

Plugins CSI (Container Storage Interface)

- **Le plugin CSI** permet à la plateforme OpenShift Container Platform de consommer le stockage **des back ends de stockage** qui implémentent **le plugin CSI en tant que stockage persistan**t.
- OpenShift Container Platform **n'est pas livré avec CSI drivers.**
- Il est recommandé d'utiliser les **drivers CSI fournis par la communauté ou les fournisseurs de stockage.**
- Les instructions d'installation diffèrent selon le driver, et se trouvent dans la documentation de chaque driver.
- Les **drivers CSI** sont livrés sous forme d'**images de conteneurs** et peuvent être **packagés dans un opérateur**

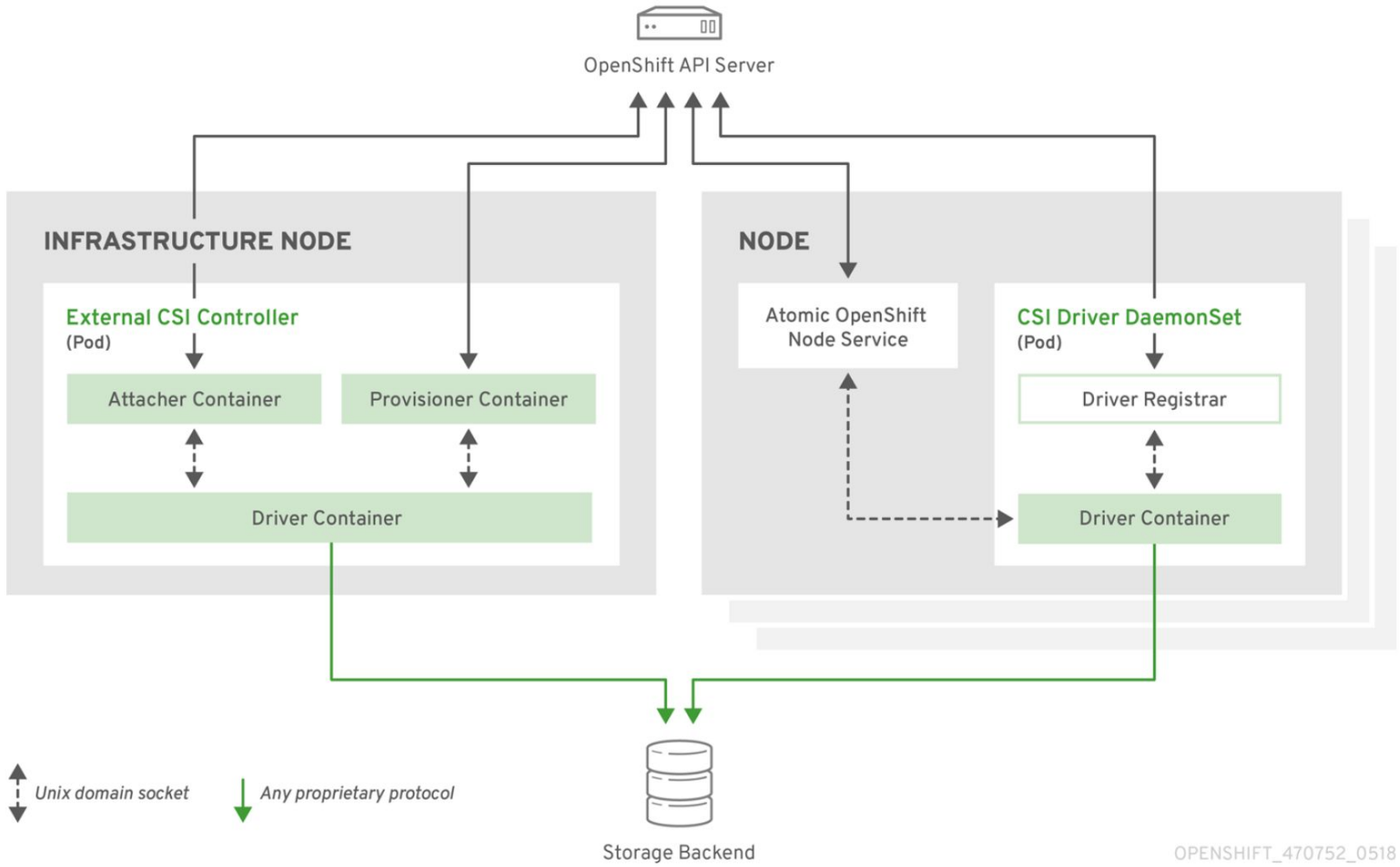
OPENSHIFT CSI

Architecture CSI

- Pour utiliser un backend de stockage compatible CSI dans OpenShift Container Platform, l'administrateur du cluster doit déployer **plusieurs composants** qui servent de bridge entre OpenShift Container Platform et le drivers de stockage.

- **Aperçu des composants** s'exécutant dans des **pods** dans le cluster OpenShift Container Platform.

- Il est possible d'exécuter **plusieurs drivers CSI pour différents backends de stockage**. Chaque driver a besoin de son propre déploiement de contrôleurs externes et de DaemonSet avec le driver et le registraire CSI.



OPENSHIFT CSI

External CSI controllers

Le contrôleur externe CSI est un déploiement d'1 ou plusieurs pods avec 3 conteneurs :

- **1 conteneur CSI Attacher externe**
  qui traduit les appels d'attachement et de détachement de la plate-forme de conteneurs OpenShift en appels respectifs de ControllerPublish et ControllerUnpublish au driver CSI.

- **1 conteneur CSI provisioner externe**
  qui traduit les appels de mise à disposition et de suppression de la plate-forme de conteneurs OpenShift en appels CreateVolume et DeleteVolume respectifs au driver CSI.

- **1 conteneur de driver CSI**
  Les conteneurs CSI Attacher et CSI Provisioner communiquent avec le conteneur CSI Driver en utilisant les UNIX Domain Sockets, ce qui garantit qu'aucune communication CSI ne quitte le pod. Le driver CSI n'est pas accessible de l'extérieur du pods.

OPENSHIFT CSI

CSI Driver DaemonSet

Le **driver CSI DaemonSet** exécute 1 pod sur chaque nœud qui permet à OpenShift Container Platform de monter le stockage fourni par le driver CSI sur le nœud et de l'utiliser dans les workloads utilisateurs (pods) comme volumes persistants (PV).
Le pod avec le driver CSI installé contient les conteneurs suivants :

- **1 CSI driver registrar,**
  qui enregistre le driver CSI dans le service du noeud openshift. Le process du noeud openshift fonctionnant sur le nœud se connecte alors directement au driver CSI en utilisant le Domain Socket UNIX disponible sur le nœud.

- **1 CSI driver**
  Le CSI driverl déployé sur le nœud doit avoir le moins d'informations d'identification sur le backend de stockage possible. OpenShift Container Platform n'utilisera le jeu de plug-ins du nœud que pour les appels CSI tels que NodePublish/NodeUnpublish et NodeStage/NodeUnstage, si ces appels sont mis en œuvre.

OPENSHIFT CSI

OpenShift et CSI

- **Container Storage Interface (CSI)** est une fonctionnalitée qui apparaît avec **OpenShift 4.2** qui permet à la solution Openshift Container Storage (**OCS**) et aux **partenaires Stockage** d'utiliser leurs plugins CSI.

Plugins In-tree

- **Kubernetes** propose déjà un système de plugins de volume qui facilite la consommation de différents types de stockage: blocs, fichiers. Mais **l'ajout de nouveaux plugins de volume a été difficile** parce que les **plugins de volume sont actuellement "in-tree"**:
  - Ils font partie du **code de base de Kubernetes et sont livrés avec les binaires de base de Kubernetes**
  - Les Partenaire de stockage souhaitant ajouter le support de leur système de stockage à Kubernetes (ou même corriger un bug dans un plugin de volume existant) **doivent se conformer au processus de publication Kubernetes.**

OPENSHIFT CSI

Plugins CSI

- Avec **l'adoption** de la Container Storage Interface (**CSI**), la couche de volume de **Kubernetes** devient vraiment **extensible**.
  - Les développeurs de stockage tiers peuvent désormais écrire et déployer des plugins de volume exposant de nouveaux systèmes de stockage dans Kubernetes sans jamais avoir à toucher au code Kubernetes de base.
  - Il va en résuler encore plus d'options pour le stockage (ex: snapshot) afin de supporter des workload conteneurisées de type stateful.

Trajectoire des plug-ins de stockage

- L'API du framework CSI a été mis à disposition à partir de la version OCP 4.2
- Des drivers CSI contenus dans les opérateurs sont apparus les mois suivants, telque OCS
- Les plugins in-tree et externes vont cohabiter jusqu'à ce que les versions CSI soient disponibles.

Le partenaires doivent rejoindre le portail partenaire TSAnet, remplir leur formulaire de soumission et indiquer à quel point ils vont s'engager avec l'équipe Storage OpenShift pour la validation de l'architecture et les exigences.

OPENSHIFT CSI

Q2 2020
OpenShift 4.5

DEV
- Customize navigation in Dev Console
- Knative event sources in Dev Console
- Form-based chart values during install
- Helm Chart upgrade, rollback and uninstall

APP
- Build Operator catalogs in container images
- Helm workflows in Console
- Monitor application workloads (TP)

PLATFORM
- OpenShift virtualization GA
- VMware vSphere (IPI) support
- Node Terminal Access in the Console
- Compact 3-node clusters for bare metal
- Compliance Operator
- HTTP/2 and gRPC Support for Router
- Graceful shutdown and recovery procedure
- Log forwarding GA
- Logging update to Elasticsearch v6
- AWS Spot instance support & IAM Identity
- Metering proxy support
- **CSI clone GA**
- Air-gapped cluster update procedure

HOSTED
- OSD on Google Cloud Platform
- ARO 20+ new deployment regions
- OCM multi-cluster dashboard

H2 2020
OpenShift 4.6

DEV
- Improved getting started experience for devs
- OpenShift Serverless Eventing GA
- OpenShift Pipelines (Tekton) GA
- Jenkins Operator TP

APP
- Monitor application workloads (GA)
- OPM tool for curating Operator catalogs
- Operator dependency tools v2
- OpenShift Builds (v2) TP

PLATFORM
- OVN GA, OVN Egress Firewall/Router/IP
- IPv6 (single/dual stack on control plane)
- Bare metal (IPI) GA
- Remote worker nodes for Edge
- Realtime kernel (TP, RAN use-cases only)
- AWS support for GovCloud, C2S, and China
- Microsoft Azure Government (MAG) support
- VMware vSphere 7.0 support
- Improved cloud credential handling
- Disconnected OpenShift Update Service
- GCP & Azure spot instances
- **CSI resize/snapshot GA**
- Windows containers GA
- Etcd improvements
- OAuth secure storage & inactivity timeout
- Enhanced RHCOS static networking UX
- OSD ISO 27001 certification
- ARO government region support

2021
OpenShift 4.7+

DEV
- OpenShift Builds (v2) GA
- Jenkins Operator GA
- Schema based forms for Event Sources

APP
- Improvements to GitOps experience
- Improvements to Operator management
- Hybrid Operators with Operator-SDK
- Simplify Operator Lifecycle interactions

PLATFORM
- Single node cluster
- Enable user namespaces
- Utilize cgroups v2
- Microsoft Hyper-V (UPI) support
- Azure Stack Hub and HCI
- Alibaba Cloud support
- Network Enhancements derived from OVN
- **Local storage support in OCS AWS**
- CVO↔OLM cluster upgrade dependencies

...more to come

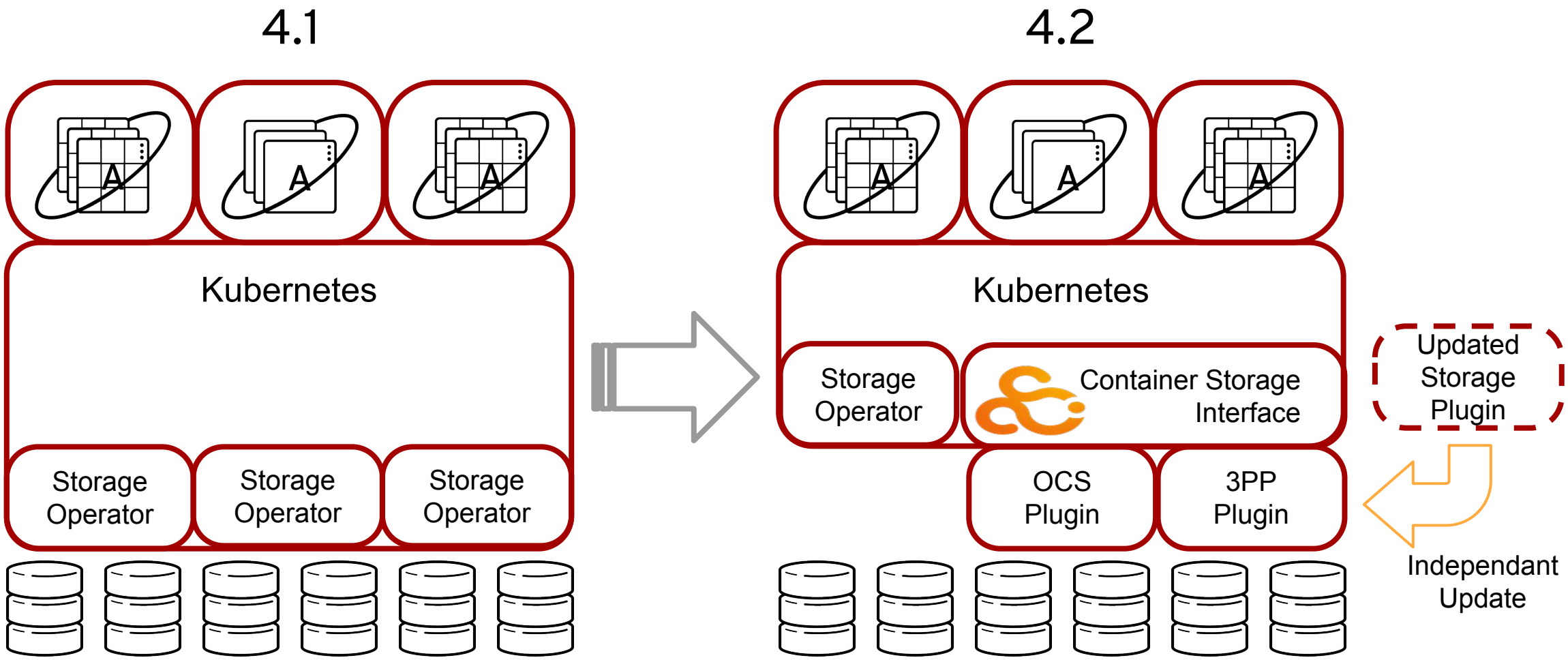
HOSTED
- Red Hat OpenShift on AWS by Amazon
- OpenShift Online Pro on OCP 4.X

OPENSHIFT CSI

- Aucun changement concernant le support support des plugins / drivers in-tree
- CSI Operators
  - **CSI Manilla GA**
    CSI Manilla fournit du Shared Filesystems (RWX) en tant que service dans les environnements Openstack
  - **CSI AWS EBS (Tech Preview)**
- CSI Capabilities
  - **Inline/Ephemeral Volumes (Tech Preview)**
    Au lieu d'utiliser d'utiliser les objets API de Kubernetes (Secrets et ConfigMaps), vous pouvez utiliser le protocole CSI pour obtenir un secret de via un coffre fort tel que Hashicorp Vault, Azure key store ou autre.
  - **Cloning GA**
  - **CSI Driver (Upstream)**

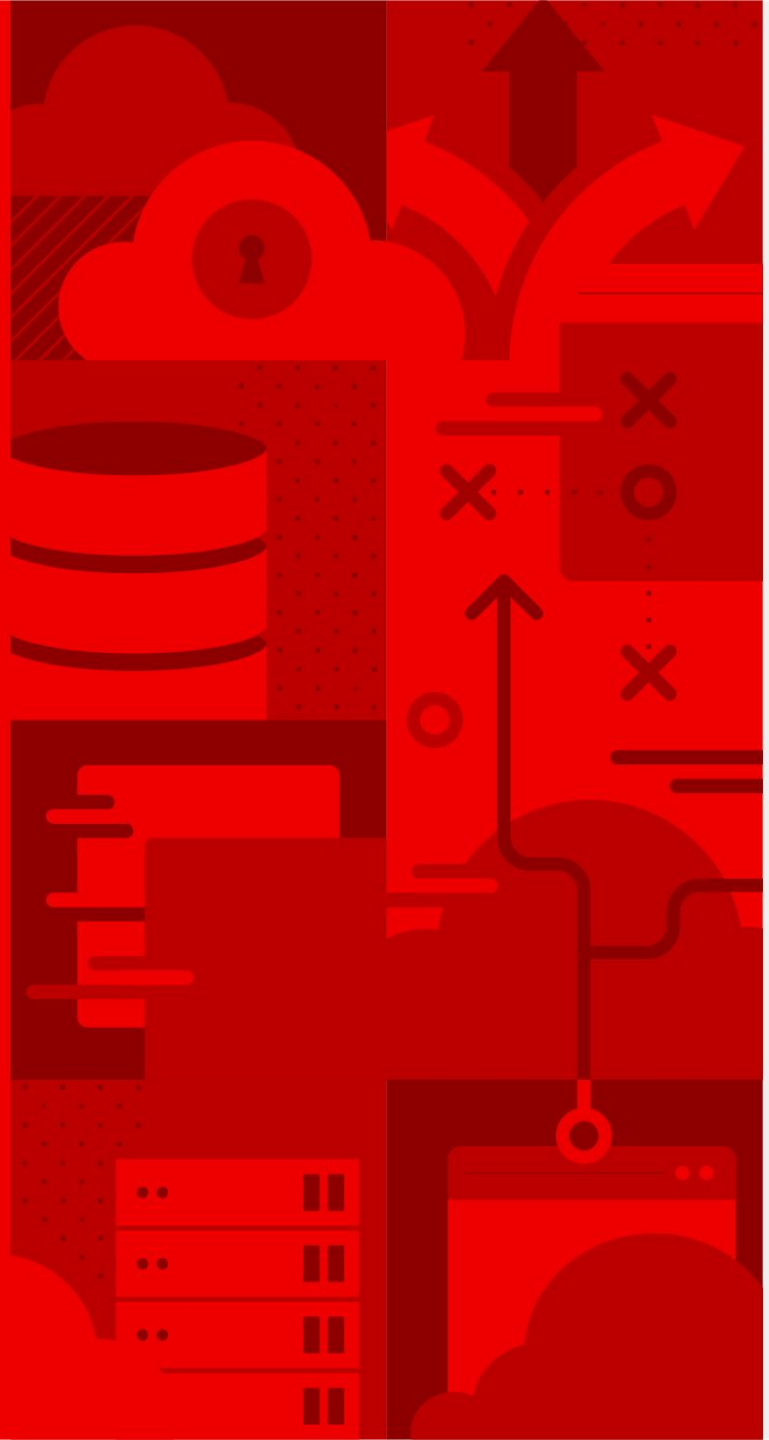
OCP Supported	
AWS EBS	Fibre Channel
Azure File & Disk	HostPath
GCE PD	Local Volume
VMware vSphere Disk	Raw Block
NFS	iSCSI
Supported via OCS	
File , Block, Raw Block, Object	
Supported via OSP	
Cinder	

OPENSHIFT CSI



CSI: ENABLING OCS AND PARTNERS

Red Hat



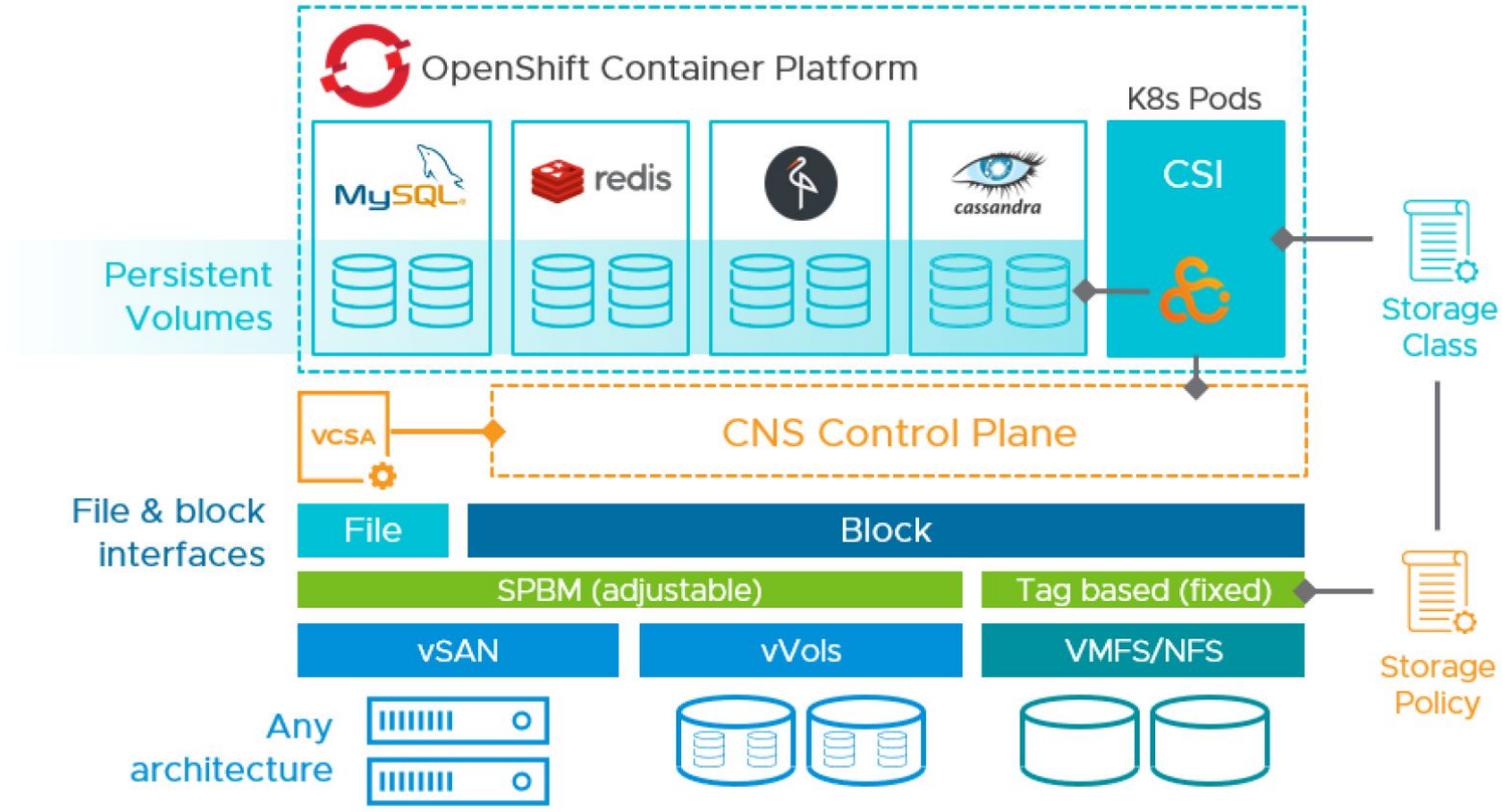
CSI driver Vmware

Red Hat

CSI driver Vmware

Architecture CSI / Vmware

- Openshift avec vSphere peut exploiter l'interface CNS (Cloud native Storage) de Vmware via le driver CSI sur vSAN.



Red Hat

CSI driver Vmware

- **Version plugin du plugin CSI et vSphere**
  La version CSI 2.0 a été annoncée en même temps que vSphere 7. Elle intégrera par exemple la fonctionnalités de **Persistent Volume Resize**

- *vSphere CSI driver currently supports various flavors*

- ***Vanilla Kubernetes***
  - *vSphere CSI driver for block volumes*
  - *vSphere CSI driver for file volumes*

- ***vSphere 7 with Kubernetes***
  - *The vSphere CSI driver on Project Pacific is called CNS CSI driver*
  - *Currently it has supports block volumes*

CNS Feature	vSphere 6.7U3 CNS 1.0	vSphere 7.0 CNS 2.0	
	Block	Block	File
Create volume (dynamic & static)	✓	✓	✓
Delete volume	✓	✓	✓
Authorization	✓	✓	✓
Cache DB	✓	✓	✓
Update API	✓	✓	✓
Full Sync	✓	✓	✓
Query APIs	✓	✓	✓
Resize volume		✓	
Cluster Flavor		✓	✓
WCP Block Guest Cluster		✓	

CSI driver Vmware

- Installation et configuration du driver **vSphere CSI Driver** sur **OpenShift 4.4**
  - Le concept de fournisseurs de cloud basés sur des plateformes à Kubernetes est en train de migrer vers une approche plus agnostique.
  - Une partie importante de tout fournisseur est le stockage Persistent du conteneur que le cloud fournit.
  - Le **Kubernetes container storage interface** ou **CSI** fournit une interface permettant d'exposer des systèmes de stockage de blocs et de fichiers arbitraires à des workload de container dans Kubernetes.
    - Installing and configuring vSphere CSI Driver on OpenShift 4.3
    - Red Hat OpenShift 4 on Vmware Vsan

CSI driver Vmware

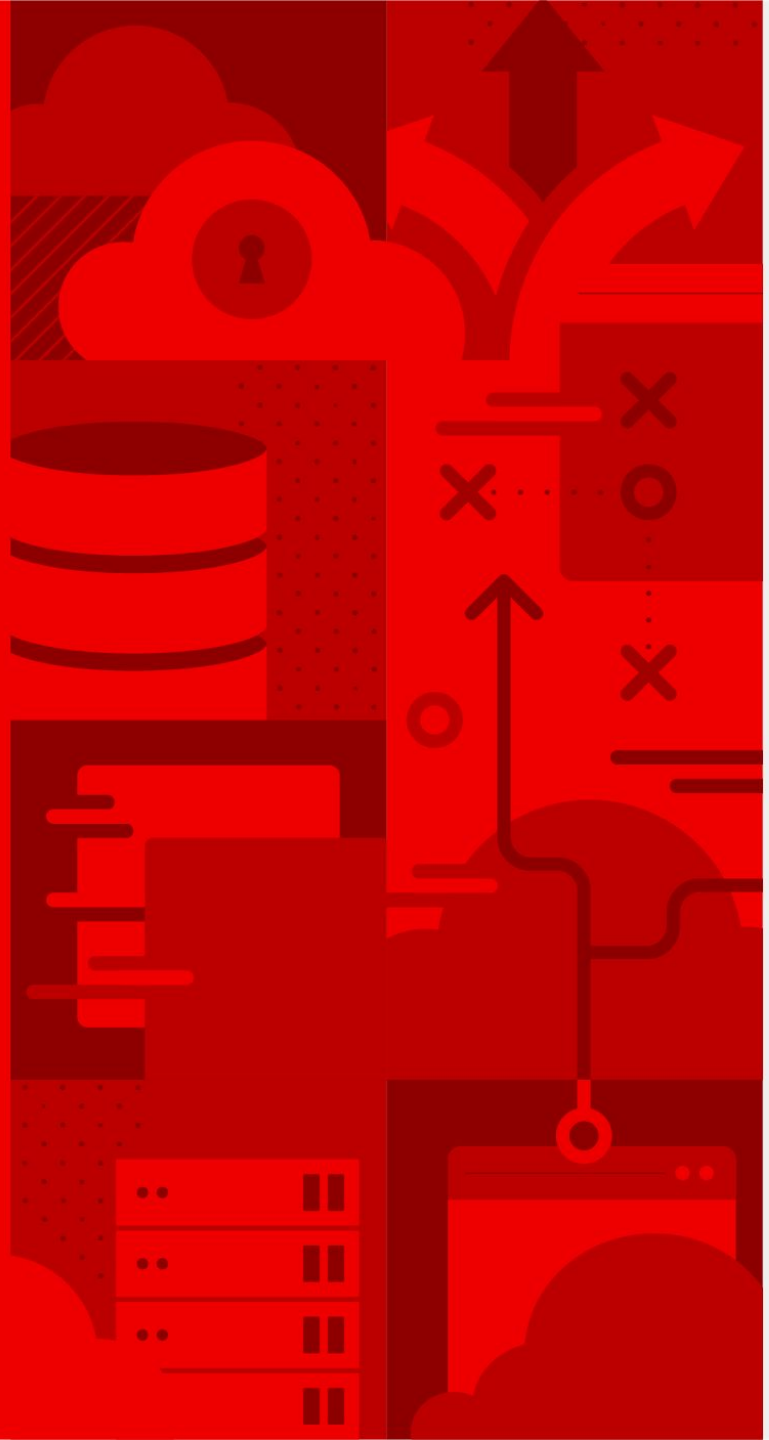
- **Passer en revue le blog**
  - Installing and configuring vSphere CSI Driver on OpenShift 4.3
    - Platform Requirements
    - Installing the vSphere Cloud Provider Interface
    - Storage Tags for Storage Policy Names
    - Install the vSphere CSI Driver
    - Storage Class creation and PVC deployment

CSI driver Vmware

Repository plugin vsphere CSI

- kubernetes-sigs/vsphere-csi-driver
  - Ce repository fournit des outils et des scripts pour construire et tester le provider vSphere CSI dans un environnement Kubernetes.
  - Ce driver est dans un état GA stable et convient à une utilisation en production. Il a actuellement besoin de vSphere 6.7 U3 ou plus pour fonctionner.
  - Le driver CSI, lorsqu'il est utilisé sur Kubernetes, nécessite également l'utilisation de l'interface vSphere vSphere Cloud Provider Interface CPI.
  - Le driver a été testé avec, et est supporté sur, K8s 1.14 et plus.

OPTIONAL SECTION MARKER OR TITLE



Local Storage

Red Hat

Principe Operateur Local Storage

- OpenShift Container Platform peut être dotée d'un stockage permanent en utilisant des volumes locaux.
- Les volumes Persistents locaux vous permettent d'accéder à des périphériques de stockage locaux, tels qu'un disque ou une partition, en utilisant l'interface PVC standard.
- Les volumes locaux peuvent être utilisés sans scheduller manuellement les Pods sur le nœuds, car le système est conscient des contraintes entre le nœud et le volume.
- Toutefois, les volumes locaux restent soumis à la disponibilité du nœud sous-jacent et ne conviennent pas à toutes les applications.

https://docs.openshift.com/container-platform/4.4/storage/persistent_storage/persistent-storage-local.html

Principe Operateur Local Storage

- Permettre la création de stockage local sur des nœuds Master et d'Infra

2. Optional: Allow local storage creation on master and infrastructure nodes.

You might want to use the Local Storage Operator to create volumes on master and infrastructure nodes, and not just worker nodes, to support components such as logging and monitoring.

To allow local storage creation on master and infrastructure nodes, add a toleration to the DaemonSet by entering the following commands:

```
$ oc patch ds local-storage-local-diskmaker -n local-storage -p '{"spec": {"template": {"spec": {"tolerations":[{"operator": "Exists"}]}}}}'
$ oc patch ds local-storage-local-provisioner -n local-storage -p '{"spec": {"template": {"spec": {"tolerations":[{"operator": "Exists"}]}}}}'
```

```
---
spec:
  template:
        spec:
        tolerations:
        - operator: Exists
```


https://docs.openshift.com/container-platform/4.4/storage/persistent_storage/persistent-storage-local.html

Principe Operateur Local Storage

- Pour créer la ressource de local volume, celle-ci doit définir les nœuds et les chemins d'accès aux volumes locaux.

```
apiVersion: "local.storage.openshift.io/v1"
kind: "LocalVolume"
metadata:
  name: "local-disks"
  namespace: "local-storage" 1
spec:
  nodeSelector: 2
    nodeSelectorTerms:
    - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
          values:
          - ip-10-0-140-183
          - ip-10-0-158-139
          - ip-10-0-164-33
  storageClassDevices:
    - storageClassName: "local-sc"
      volumeMode: Filesystem 3
      fsType: xfs 4
      devicePaths: 5
        - /path/to/device 6
```

1. The namespace where the Local Storage Operator is installed.
2. Optional: A node selector containing a list of nodes where the local storage volumes are attached. This example uses the node host names, obtained from `oc get node`. If a value is not defined, then the Local Storage Operator will attempt to find matching disks on all available nodes.
3. The volume mode, either `Filesystem` or `Block`, defining the type of the local volumes.
4. The file system that is created when the local volume is mounted for the first time.
5. The path containing a list of local storage devices to choose from.
6. Replace this value with your actual local disks filepath to the LocalVolume resource, such as `/dev/xvdg`. PVs are created for these local disks when the provisioner is deployed successfully.

https://docs.openshift.com/container-platform/4.4/storage/persistent_storage/persistent-storage-local.html

Principe Operateur Local Storage

- L'opérateur créé sur chaque node ou est attaché un device local un pod **provisioner** et **diskmaker** sous forme de DaemonSet.

```
$ oc get all -n local-storage

NAME                                          READY   STATUS    RESTARTS   AGE
pod/local-disks-local-provisioner-h97hj       1/1     Running   0          46m
pod/local-disks-local-provisioner-j4mnn       1/1     Running   0          46m
pod/local-disks-local-provisioner-kbdnx       1/1     Running   0          46m
pod/local-disks-local-diskmaker-ldldw         1/1     Running   0          46m
pod/local-disks-local-diskmaker-lvrv4         1/1     Running   0          46m
pod/local-disks-local-diskmaker-phxdq         1/1     Running   0          46m
pod/local-storage-operator-54564d9988-vxvhx   1/1     Running   0          47m

NAME                               TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)     AGE
service/local-storage-operator     ClusterIP   172.30.49.90     <none>        60000/TCP   47m

NAME                                           DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE
daemonset.apps/local-disks-local-provisioner   3         3         3       3            3
daemonset.apps/local-disks-local-diskmaker     3         3         3       3            3

NAME                                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/local-storage-operator   1/1     1            1           47m

NAME                                                DESIRED   CURRENT   READY   AGE
replicaset.apps/local-storage-operator-54564d9988   1         1         1       47m
```

https://docs.openshift.com/container-platform/4.4/storage/persistent_storage/persistent-storage-local.html

Principe Operateur Local Storage

- Les PersistentVolumes sont créés à l'aide du local volume provisioner

```
$ oc get pv

NAME                CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM   STORAGECLASS
local-pv-1cec77cf   100Gi      RWO            Delete           Available           local-sc
local-pv-2ef7cd2a   100Gi      RWO            Delete           Available           local-sc
local-pv-3fa1c73    100Gi      RWO            Delete           Available           local-sc
```


https://docs.openshift.com/container-platform/4.4/storage/persistent_storage/persistent-storage-local.html

Principe Operateur Local Storage

- Le PVC est créé en utilisant la StorageClass définit dans le local storage.

```yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: local-pvc-name (1)
spec:
  accessModes:
  - ReadWriteOnce
  volumeMode: Filesystem (2)
  resources:
    requests:
      storage: 100Gi (3)
  storageClassName: local-sc (4)
```

1. Name of the PVC.
2. The type of the PVC. Defaults to `Filesystem`.
3. The amount of storage available to the PVC.
4. Name of the StorageClass required by the claim.

https://docs.openshift.com/container-platform/4.4/storage/persistent_storage/persistent-storage-local.html

Principe Operateur Local Storage

- On peut ensuite déclarer le PVC à l'intérieur du PodOn peut ensuite déclarer le PVC à l'intérieur du Pod.

```
apiVersion: v1
kind: Pod
spec:
  ...
  containers:
    volumeMounts:
    - name: localpvc          (1)
      mountPath: "/data"      (2)
  volumes:
  - name: localpvc
    persistentVolumeClaim:
      claimName: localpvc     (3)
```

1. Name of the volume to mount.
2. Path inside the Pod where the volume is mounted.
3. Name of the existing PVC to use.

https://docs.openshift.com/container-platform/4.4/storage/persistent_storage/persistent-storage-local.html

Principe Operateur Local Storage

- Des Taints peuvent être appliqués aux nœuds pour les empêcher d'exécuter un workload en général. Pour permettre au Local Storage Operator d'utiliser des nœuds avec une tainte , on peut ajouter des Taint à la définition du Pod ou du DaemonSet. Cela permet aux ressources créées de s'exécuter sur ces nœuds Taintés.

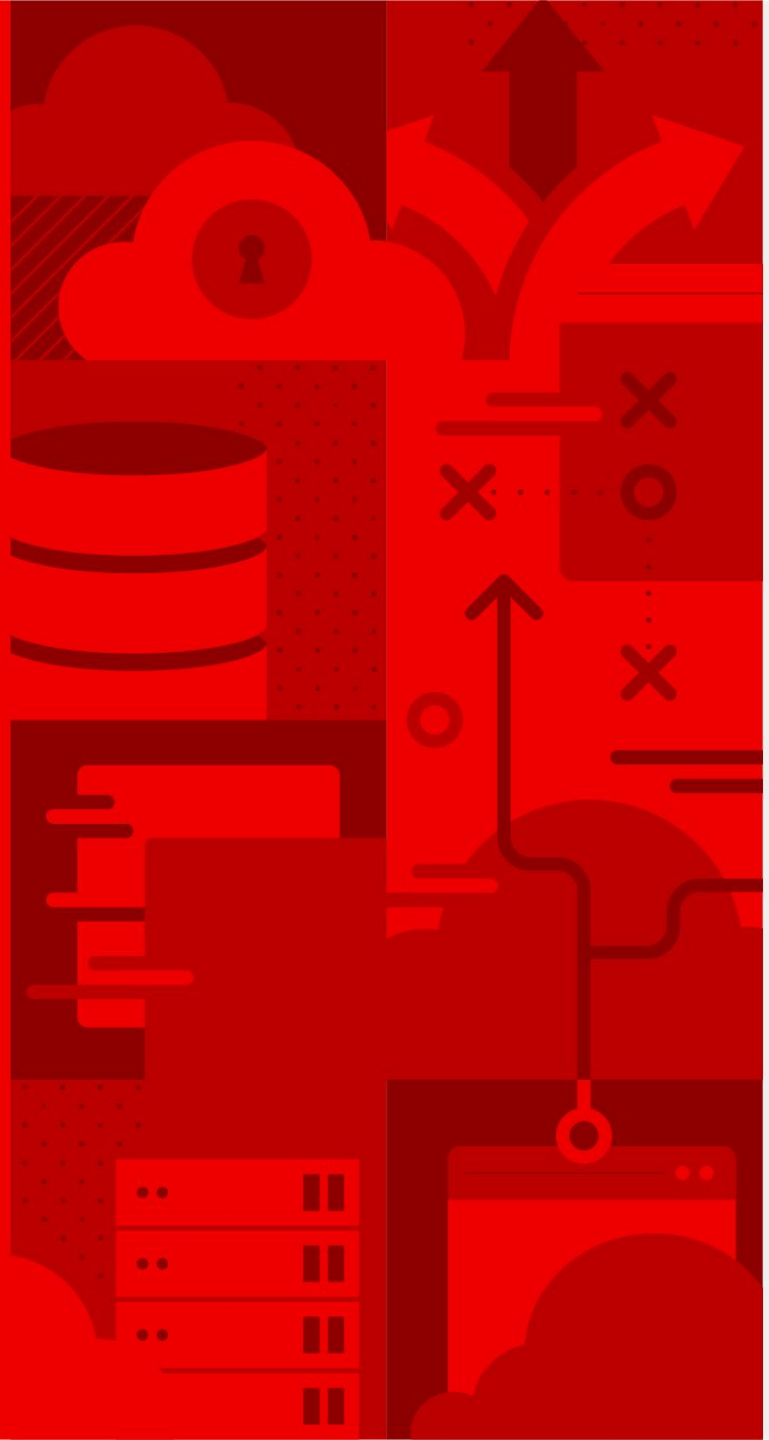
```
apiVersion: "local.storage.openshift.io/v1"
kind: "LocalVolume"
metadata:
  name: "local-disks"
  namespace: "local-storage"
spec:
  tolerations:
    - key: localstorage          (1)
      operator: Equal            (2)
      value: "localstorage"      (3)
  storageClassDevices:
    - storageClassName: "localblock-sc"
      volumeMode: Block          (4)
      devicePaths:               (5)
        - /dev/xvdg
```

1. Specify the key that you added to the node.
2. Specify the `Equal` operator to require the `key`/`value` parameters to match. If operator is 'Exists`, the system checks that the key exists and ignores the value. If operator is `Equal`, then the key and value must match.
3. Specify the value `local` of the tainted node.
4. The volume mode, either `Filesystem` or `Block`, defining the type of the local volumes.
5. The path containing a list of local storage devices to choose from.

https://docs.openshift.com/container-platform/4.4/storage/persistent_storage/persistent-storage-local.html

Principe Operateur Local Storage

- Suppression complète des "local storage" créés et de l'opérateur local storage.

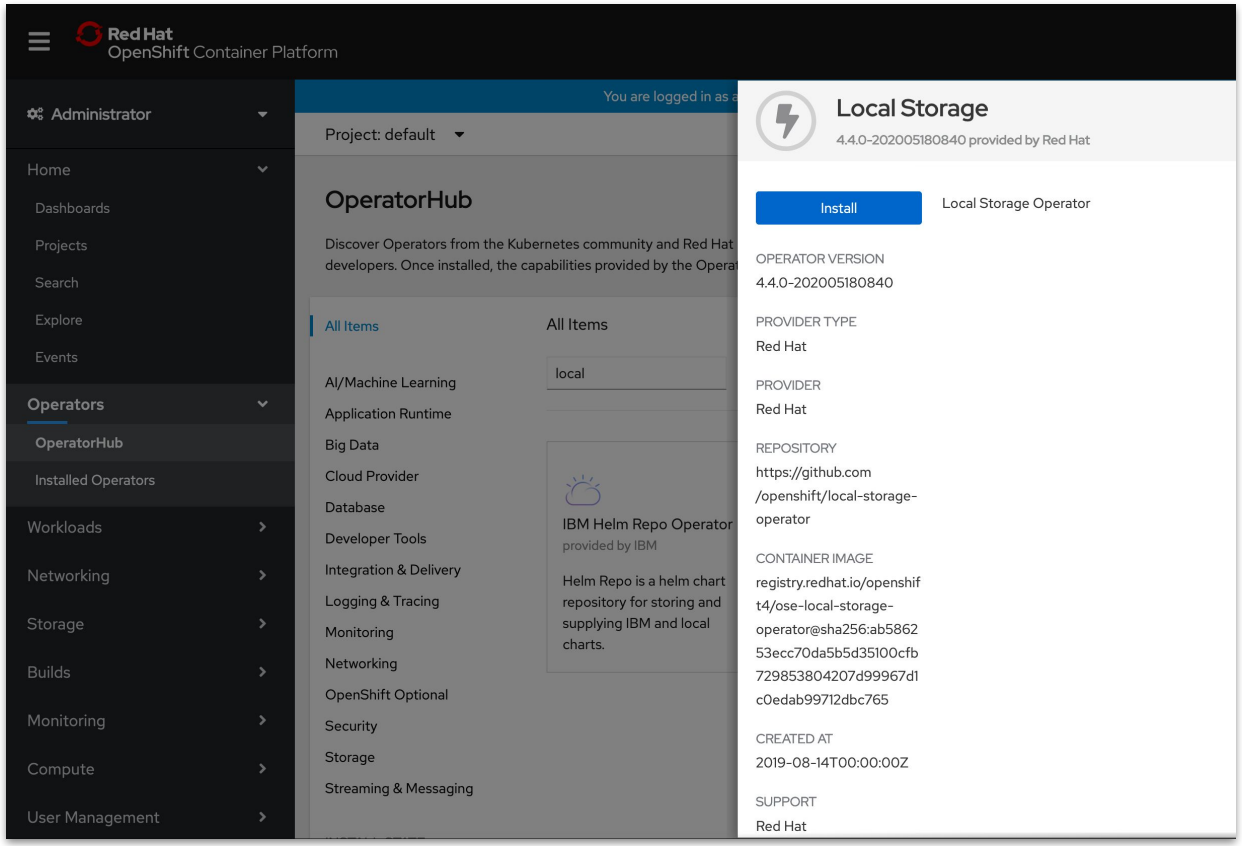


Installation

Red Hat

Installation Operateur Local Storage

- Creer un project Local-Storage: "**oc new-project local-storage**"
- Installer via l'opérateur hub l'opérateur **local-storage**"

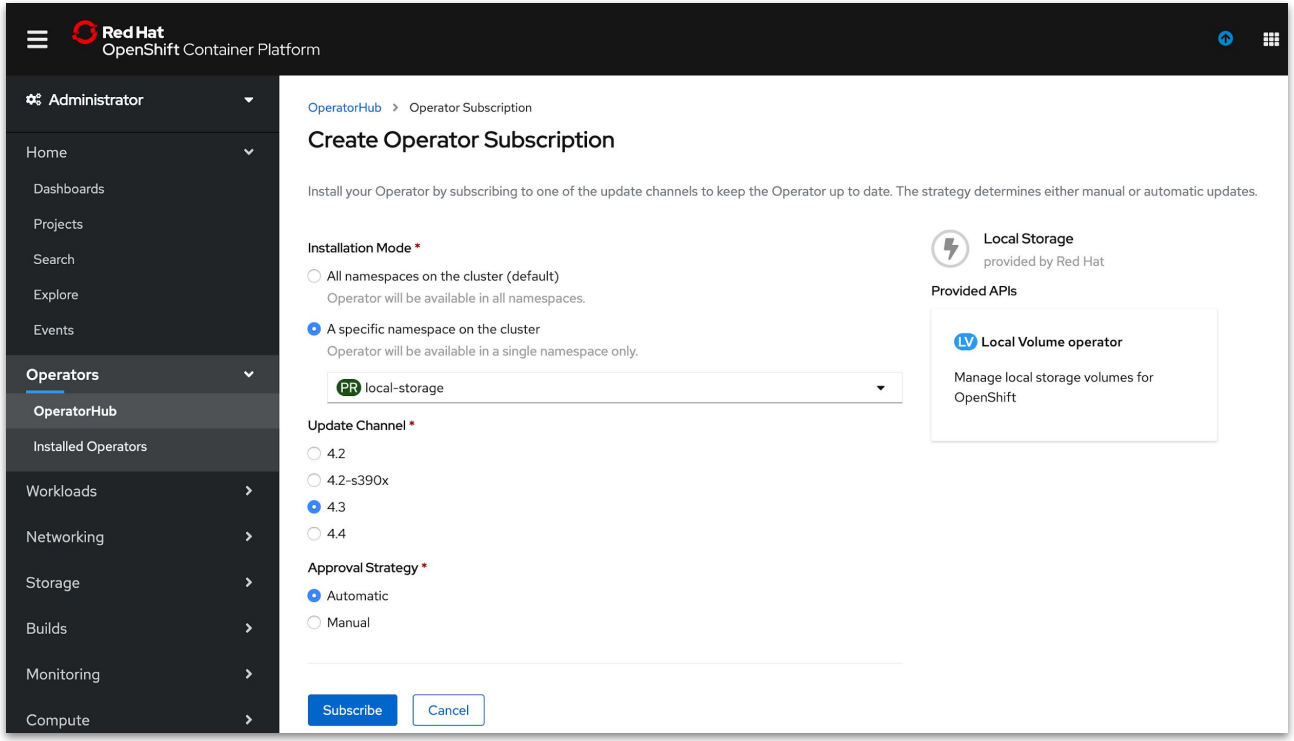



```
oc new-project local-storage

oc create -f - <<EOF
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
 name: local-storage-xxxx
 namespace: local-storage
spec:
 targetNamespaces:
   - local-storage
EOF
```

Installation Operateur Local Storage

- Créer une souscription dans le namespace **local-storage**"

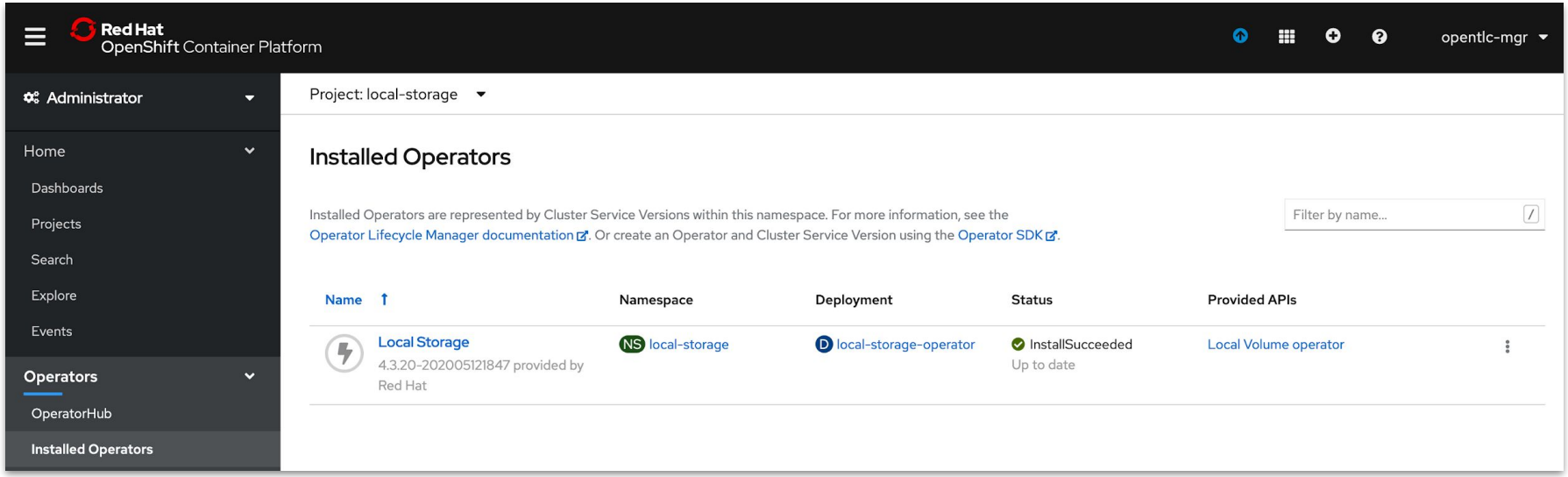



```yaml
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
 name: local-storage-operator
 namespace: local-storage
spec:
 channel: '4.3'
 installPlanApproval: Automatic
 name: local-storage-operator
 source: redhat-operators
 sourceNamespace: openshift-marketplace
```


Red Hat

Installation Operateur Local Storage

- Vérifier l'installation de l'opérateur

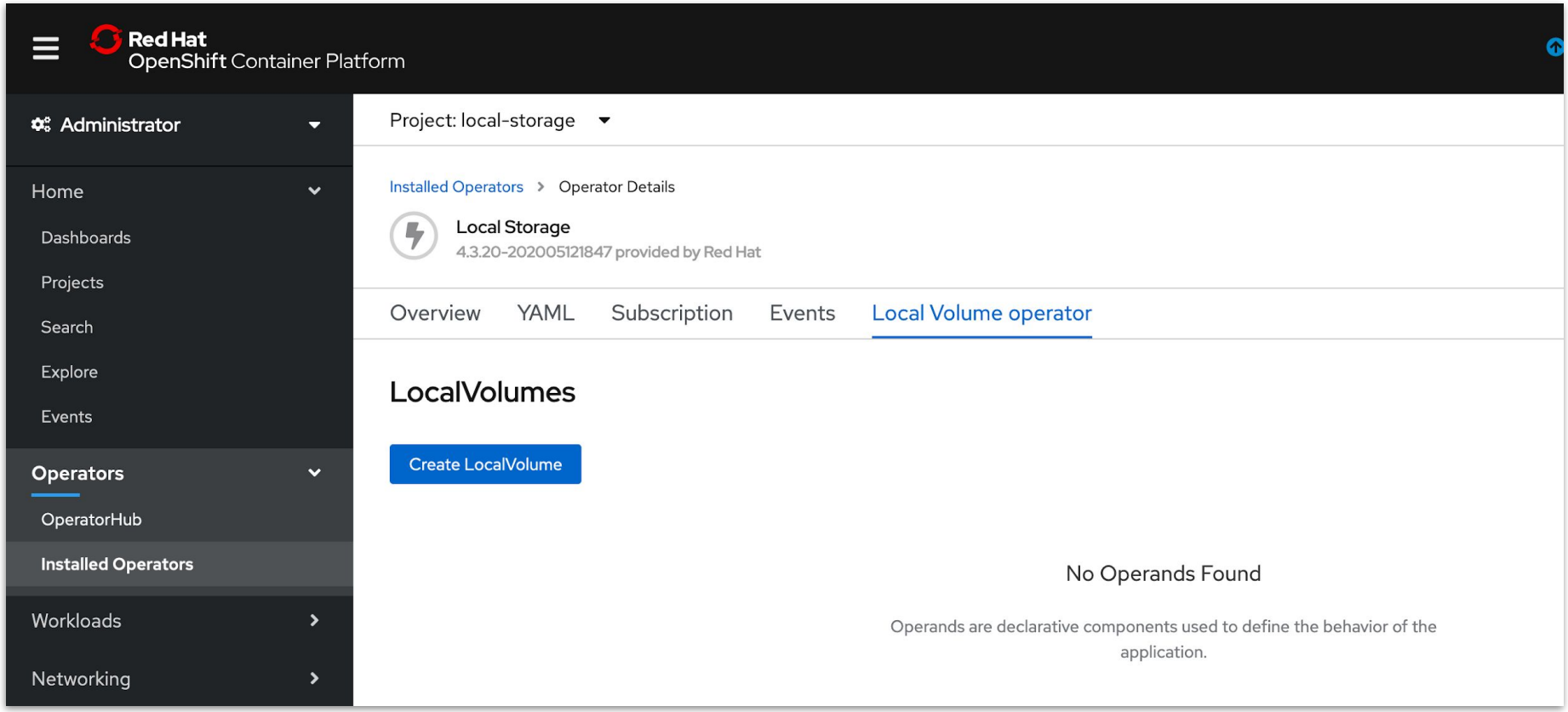



```
oc get  pods
NAME                                      READY   STATUS    RESTARTS   AGE
local-storage-operator-6b649cfbdd-jk52x   1/1     Running   0          7m24s
```


Red Hat

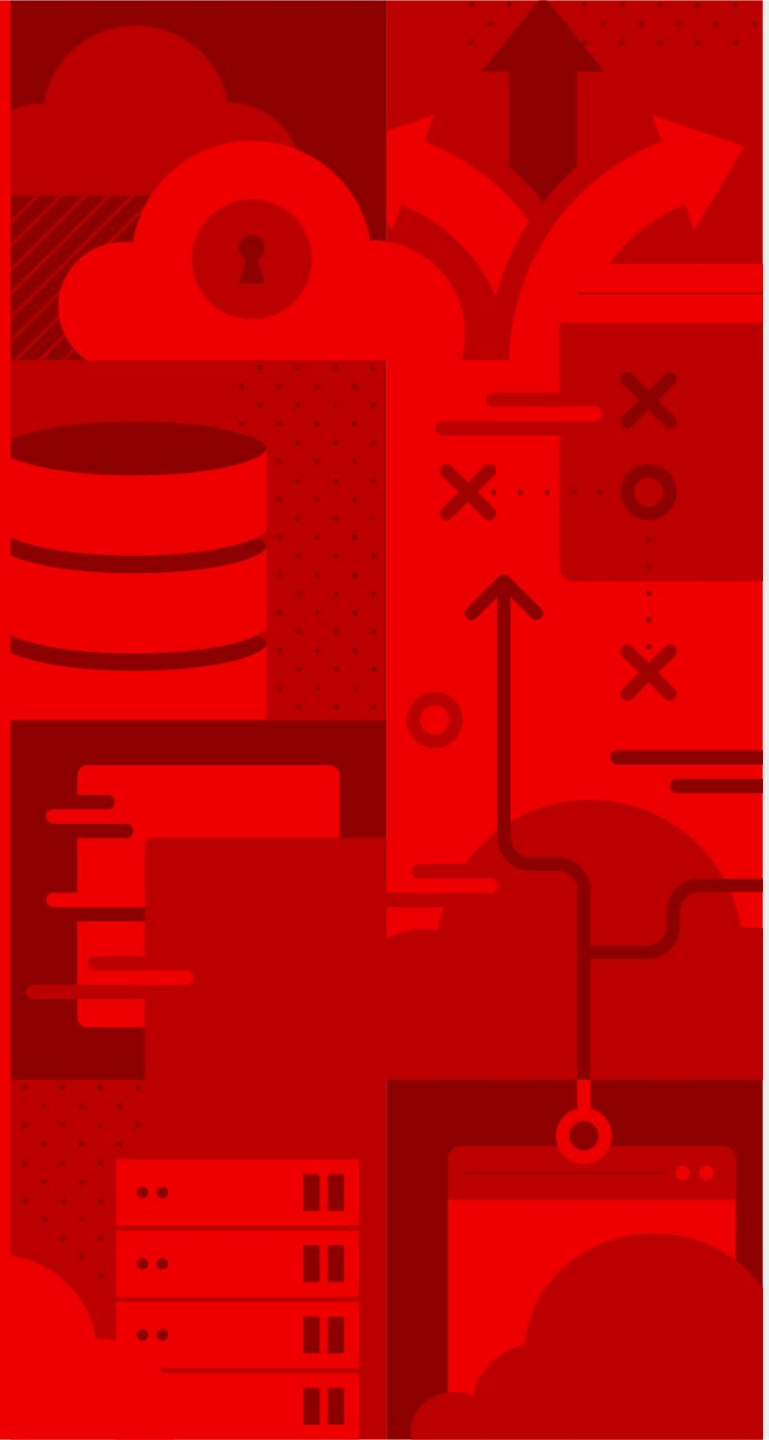
Installation Operateur Local Storage

- L'ajout de classe de stockage de type local se fera via Local Volume Opereator



Red Hat

Use Cases



Red Hat

Exemple de use case - Device Disk

- Vérifier la présence des devices

```
for i in {1..3} ; do ssh core@worker-${i}.ocp42.ssa.mbu.labs.redhat.com lsblk | egrep "^sdb.*|sdc.*$"
; done

sdb      8:16   0   10G  0 disk
sdc      8:32   0  100G  0 disk
sdb      8:16   0   10G  0 disk
sdc      8:32   0  100G  0 disk
sdb      8:16   0   10G  0 disk
sdc      8:32   0  100G  0 disk
```


https://www.openshift.com/blog/ocs-4-2-in-ocp-4-2-14-upi-installation-in-rhv

Exemple de use case - Device Disk

- Ajouter un local volume de type **volumeMode : FileSystem**
  La classe de stockage sera nommée : **local-sc**

```
[root@utility ~]# cat <<EOF >
local-storage-filesystem.yaml
apiVersion: "local.storage.openshift.io/v1"
kind: "LocalVolume"
metadata:
 name: "local-disks-fs"
 namespace: "local-storage"
spec:
 nodeSelector:
   nodeSelectorTerms:
   - matchExpressions:
       - key: kubernetes.io/hostname
         operator: In
         values:
         - worker-1.ocp42.ssa.mbu.labs.redhat.com
         - worker-2.ocp42.ssa.mbu.labs.redhat.com
         - worker-3.ocp42.ssa.mbu.labs.redhat.com
 storageClassDevices:
   - storageClassName: "local-sc"
     volumeMode: Filesystem
     devicePaths:
       - /dev/sdb
EOF
```

```
[root@utility ~]# oc get pod -n local-storage
NAME                                      READY   STATUS    RESTARTS   AGE
local-disks-fs-local-diskmaker-2bqw4      1/1     Running   0          106s
local-disks-fs-local-diskmaker-8w9rz      1/1     Running   0          106s
local-disks-fs-local-diskmaker-khhm5      1/1     Running   0          106s
local-disks-fs-local-provisioner-g5dgv    1/1     Running   0          106s
local-disks-fs-local-provisioner-hkj69    1/1     Running   0          106s
local-disks-fs-local-provisioner-vhpj8    1/1     Running   0          106s
local-storage-operator-ccbb59b45-nn7ww    1/1     Running   0          15m
[root@utility ~]# oc get sc
NAME       PROVISIONER                    AGE
local-sc   kubernetes.io/no-provisioner   109s
[root@utility ~]# oc get pv
NAME                CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM   STORAGECLASS
local-pv-68faed78   10Gi       RWO            Delete           Available           local-sc
local-pv-780afdd6   10Gi       RWO            Delete           Available           local-sc
local-pv-b640422f   10Gi       RWO            Delete           Available           local-sc
[root@utility ~]#
```

https://www.openshift.com/blog/ocs-4-2-in-ocp-4-2-14-upi-installation-in-rhv

Exemple de use case - Device Disk

- Ajouter un local volume de type **volumeMode : Block**
- La classe de stockage sera nommée : **localblock-sc**

```
[root@utility ~]# cat <<EOF >
local-storage-block.yaml
apiVersion: "local.storage.openshift.io/v1"
kind: "LocalVolume"
metadata:
 name: "local-disks"
 namespace: "local-storage"
spec:
 nodeSelector
   nodeSelectorTerms
   - matchExpressions:
       - key: kubernetes.io/hostname
         operator: I
         values:
         - worker-1.ocp42.ssa.mbu.labs.redhat.com
         - worker-2.ocp42.ssa.mbu.labs.redhat.com
         - worker-3.ocp42.ssa.mbu.labs.redhat.com
 storageClassDevices:
   - storageClassName: "localblock-sc"
     volumeMode: Block
     devicePaths:
       - /dev/sdc
EOF
```

```
[root@utility ~]# oc get pod -n local-storage
NAME                                      READY   STATUS    RESTARTS   AGE
local-disks-fs-local-diskmaker-2bqw4      1/1     Running   0          6m33s
local-disks-fs-local-diskmaker-8w9rz      1/1     Running   0          6m33s
local-disks-fs-local-diskmaker-khhm5      1/1     Running   0          6m33s
local-disks-fs-local-provisioner-g5dgv    1/1     Running   0          6m33s
local-disks-fs-local-provisioner-hkj69    1/1     Running   0          6m33s
local-disks-fs-local-provisioner-vhpj8    1/1     Running   0          6m33s
local-disks-local-diskmaker-6qpfx         1/1     Running   0          22s
local-disks-local-diskmaker-pw5ql         1/1     Running   0          22s
local-disks-local-diskmaker-rc5hr         1/1     Running   0          22s
local-disks-local-provisioner-9qprp       1/1     Running   0          22s
local-disks-local-provisioner-kkkcm       1/1     Running   0          22s
local-disks-local-provisioner-kxbnn       1/1     Running   0          22s
local-storage-operator-ccbb59b45-nn7ww    1/1     Running   0          19m
[root@utility ~]# oc get sc
NAME            PROVISIONER                    AGE
local-sc        kubernetes.io/no-provisioner   6m36s
localblock-sc   kubernetes.io/no-provisioner   25s
[root@utility ~]# oc get pv
NAME                CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM   STORAGECLASS
local-pv-5c4e718c   100Gi      RWO            Delete           Available           localblock-s
local-pv-68faed78   10Gi       RWO            Delete           Available           local-sc
local-pv-6a58375e   100Gi      RWO            Delete           Available           localblock-s
local-pv-780afdd6   10Gi       RWO            Delete           Available           local-sc
local-pv-b640422f   10Gi       RWO            Delete           Available           local-sc
local-pv-d6db37fd   100Gi      RWO            Delete           Available           localblock-s
[root@utility ~]#
```

https://www.openshift.com/blog/ocs-4-2-in-ocp-4-2-14-upi-installation-in-rhv

Exemple de use case - Disque NVME

- Ajouter un local volume de type **volumeMode : FileSystem**
  La classe de stockage sera nommée : **local-sc**

```
NAME                          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                             8:0    0   60G  0 disk
|-sda1                          8:1    0  384M  0 part /boot
|-sda2                          8:2    0  127M  0 part /boot/efi
|-sda3                          8:3    0    1M  0 part
`-sda4                          8:4    0 59.5G  0 part
  `-coreos-luks-root-nocrypt 253:0    0 59.5G  0 dm   /sysroot
sdb                             8:16   0   16G  0 disk
nvme1n1                       259:0    0  1.5T  0 disk
Warning: Permanently added '10.70.56.95' (ECDSA) to the list of known hosts.
NAME                          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                             8:0    0   60G  0 disk
|-sda1                          8:1    0  384M  0 part /boot
|-sda2                          8:2    0  127M  0 part /boot/efi
|-sda3                          8:3    0    1M  0 part
`-sda4                          8:4    0 59.5G  0 part
  `-coreos-luks-root-nocrypt 253:0    0 59.5G  0 dm   /sysroot
sdb                             8:16   0   16G  0 disk
nvme0n1                       259:0    0  1.5T  0 disk
Warning: Permanently added '10.70.56.96' (ECDSA) to the list of known hosts.
NAME                          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                             8:0    0   60G  0 disk
|-sda1                          8:1    0  384M  0 part /boot
|-sda2                          8:2    0  127M  0 part /boot/efi
|-sda3                          8:3    0    1M  0 part
`-sda4                          8:4    0 59.5G  0 part
  `-coreos-luks-root-nocrypt 253:0    0 59.5G  0 dm   /sysroot
sdb                             8:16   0   16G  0 disk
nvme0n1                       259:0    0  1.5T  0 disk
```


https://www.openshift.com/blog/blog-leveraging-local-nvme-disks-for-openshift-container-storage-on-vmware

Exemple de use case - Disque NVME

- Vérifier la présence des devices

```
→  website git:(master) ✗ cat <<EOF | oc create -n local-storage -f -
apiVersion: local.storage.openshift.io/v1
kind: LocalVolume
metadata:
  name: local-block
  namespace: local-storage
spec:
  nodeSelector:
    nodeSelectorTerms:
    - matchExpressions:
        - key: cluster.ocs.openshift.io/openshift-storage
          operator: Exists
  storageClassDevices:
    - storageClassName: local-block
      volumeMode: Block
      devicePaths:
        - /dev/nvme0n1
        - /dev/nvme1n1
EOF
```


https://www.openshift.com/blog/blog-leveraging-local-nvme-disks-for-openshift-container-storage-on-vmware

Exemple de use case - Disque NVME

- Ajouter un local volume de type **volumeMode : Block**
- La classe de stockage sera nommée : **localblock-sc**

```
→  website git:(master) ✗ cat <<EOF | oc create -n local-storage -f -
apiVersion: local.storage.openshift.io/v1
kind: LocalVolume
metadata:
  name: local-fs
  namespace: local-storage
spec:
  nodeSelector:
    nodeSelectorTerms:
    - matchExpressions:
        - key: cluster.ocs.openshift.io/openshift-storage
          operator: Exists
  storageClassDevices:
    - storageClassName: local-fs
      fsType: xfs
      volumeMode: Filesystem
      devicePaths:
        - /dev/sdb
EOF
```


https://www.openshift.com/blog/blog-leveraging-local-nvme-disks-for-openshift-container-storage-on-vmware

Exemple de use case - by Id disk

- Création d'un OpenShift Container Storage sur VMware
  - **Virtual machine disk (VMDK)**
  - **Raw device mapping (RDM)**
  - **VMDirectPath I/O**

https://access.redhat.com/documentation/en-us/red_hat_openshift_container_storage/4.3/html-single/deploying_openshift_container_storage/index#installing-local-storage-operator_rhocs

Exemple de use case - by Id disk

- Ajouter un local volume de type **volumeMode : FileSystem**
  La classe de stockage sera nommée : **local-sc**

```
apiVersion: local.storage.openshift.io/v1
kind: LocalVolume
metadata:
 name: local-file
 namespace: local-storage
spec:
 nodeSelector:
   nodeSelectorTerms:
   - matchExpressions:
       - key: cluster.ocs.openshift.io/openshift-storage
         operator: In
         values:
         - ""
 storageClassDevices:
   - storageClassName: localfile
     volumeMode: Filesystem
     devicePaths:
       - /dev/disk/by-id/scsi-36000c29520486e45a6896c58a10de97d   # <--
modify this line
        - /dev/disk/by-id/scsi-36000c29194bcdc3fbd865a67057a29ec   # <--
modify this line
        - /dev/disk/by-id/scsi-36000c2991133c4f1d6a604f3bc086967   # <--
modify this line
```


https://access.redhat.com/documentation/en-us/red_hat_openshift_container_storage/4.3/html-single/deploying_openshift_container_storage/index#installing-local-storage-operator_rhocs

Exemple de use case - by Id disk

- Ajouter un local volume de type **volumeMode : Block**
- La classe de stockage sera nommée : **localblock-sc**

```
apiVersion: local.storage.openshift.io/v1
kind: LocalVolume
metadata:
  name: local-block
  namespace: local-storage
spec:
  nodeSelector:
    nodeSelectorTerms:
    - matchExpressions:
        - key: cluster.ocs.openshift.io/openshift-storage
          operator: In
          values:
          - ""
  storageClassDevices:
    - storageClassName: localblock
      volumeMode: Block
      devicePaths:
        - /dev/disk/by-id/scsi-36000c2991c27c2e5ba7c47d1e4352de2   # <--
modify this line
        - /dev/disk/by-id/scsi-36000c29682ca9e347926406711f3dc4e   # <--
modify this line
        - /dev/disk/by-id/scsi-36000c296aaf03a9b1e4b01d086bc6348   # <--
modify this line
```


https://access.redhat.com/documentation/en-us/red_hat_openshift_container_storage/4.3/html-single/deploying_openshift_container_storage/index#installing-local-storage-operator_rhocs

Exemple - Extension du stockage

- Augmenter le stockage en ajoutant de la capacité à des nœuds OpenShift Container Storage en utilisant des périphériques de stockage locaux

4.2.2. Scaling up storage by adding capacity to your OpenShift Container Storage nodes using local storage devices

Use this procedure to add storage capacity (additional storage devices) to your configured local storage based OpenShift Container Storage worker nodes on bare metal, Amazon EC2 (storage optimized - i3en.2xlarge), and VMware infrastructures.

IMPORTANT

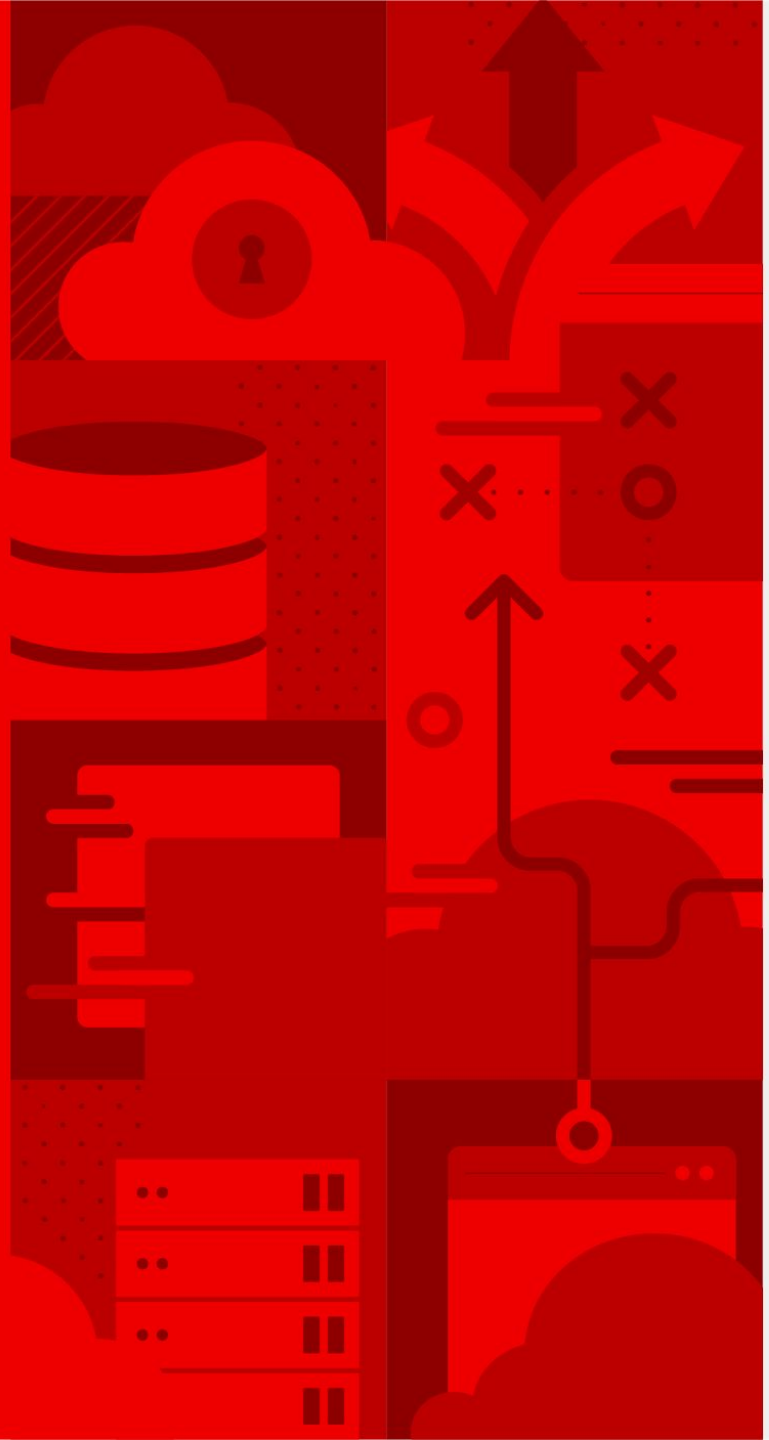
Scaling up storage on bare metal, Amazon EC2 (storage optimized - i3en.2xlarge), and VMware is a Technology Preview feature. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

NOTE

For Amazon EC2 (storage optimized - i3en.2xlarge) infrastructure, adding nodes is the only option for adding capacity, as deployment is done using both the available NVMe devices.

https://access.redhat.com/documentation/en-us/red_hat_openshift_container_storage/4.3/html-single/managing_openshift_container_storage/index#proc_scaling-up-storage-by-adding-capacity-to-your-openshift-container-storage-nodes-on-aws-vmware-infrastructure_rhocs

Bug Fix



Red Hat

Bug Fix Advisory

RHBA-2020:0581 - Bug Fix Advisory

Aperçu général | Paquets mis à jour

Synopsis

OpenShift Container Platform 4.4 image release advisory

Type / Sévérité

Bug Fix Advisory

Sujet

Red Hat OpenShift Container Platform release 4.4.3 is now available with updates to packages and images that fix several bugs and add enhancements.

BZ - 1780625 - local-storage-operator does not support LVM devices

Liang Xia 2019-12-19 06:26:37 UTC

```
Verified with local-storage-operator.4.4.0-201912170523 on 4.4.0-0.nightly-2019-12-18-064553


$ oc get sc,pv
NAME                                          PROVISIONER                    AGE
storageclass.storage.k8s.io/local-sc          kubernetes.io/no-provisioner   5m41s
storageclass.storage.k8s.io/standard (default)   kubernetes.io/cinder        3h28m

NAME                                CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM   STORAGECLASS   REASON   AGE
persistentvolume/local-pv-104b75f1   1Gi        RWO            Delete           Available           local-sc                2m20s


sh-4.4# lvdisplay
  --- Logical volume ---
  LV Path                /dev/myvg01/mylv01
  LV Name                mylv01
  VG Name                myvg01
  LV UUID                mvT0Ww-eOaz-i0Dm-4ZZo-HoPu-3EHB-dVUxpV
  LV Write Access        read/write
  LV Creation host, time lxia19-48fdq-worker-dsf5s, 2019-12-19 06:15:56 +0000
  LV Status              available
  # open                 0
  LV Size                1.00 GiB
  Current LE             256
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:1
```

https://access.redhat.com/errata/RHBA-2020:0581

Red Hat

Solutions

How to manage filesystems on RHEL CoreOS with OpenShift Container Platform
- https://access.redhat.com/solutions/4608041

OCP 4.2 and Container runtime storage configuration with iSCSI in UPI Bare Metal installations
- https://access.redhat.com/solutions/4574801

[RHCOS] Increasing partition size
- https://access.redhat.com/solutions/4608041

https://access.redhat.com/errata/RHBA-2020:0581

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat