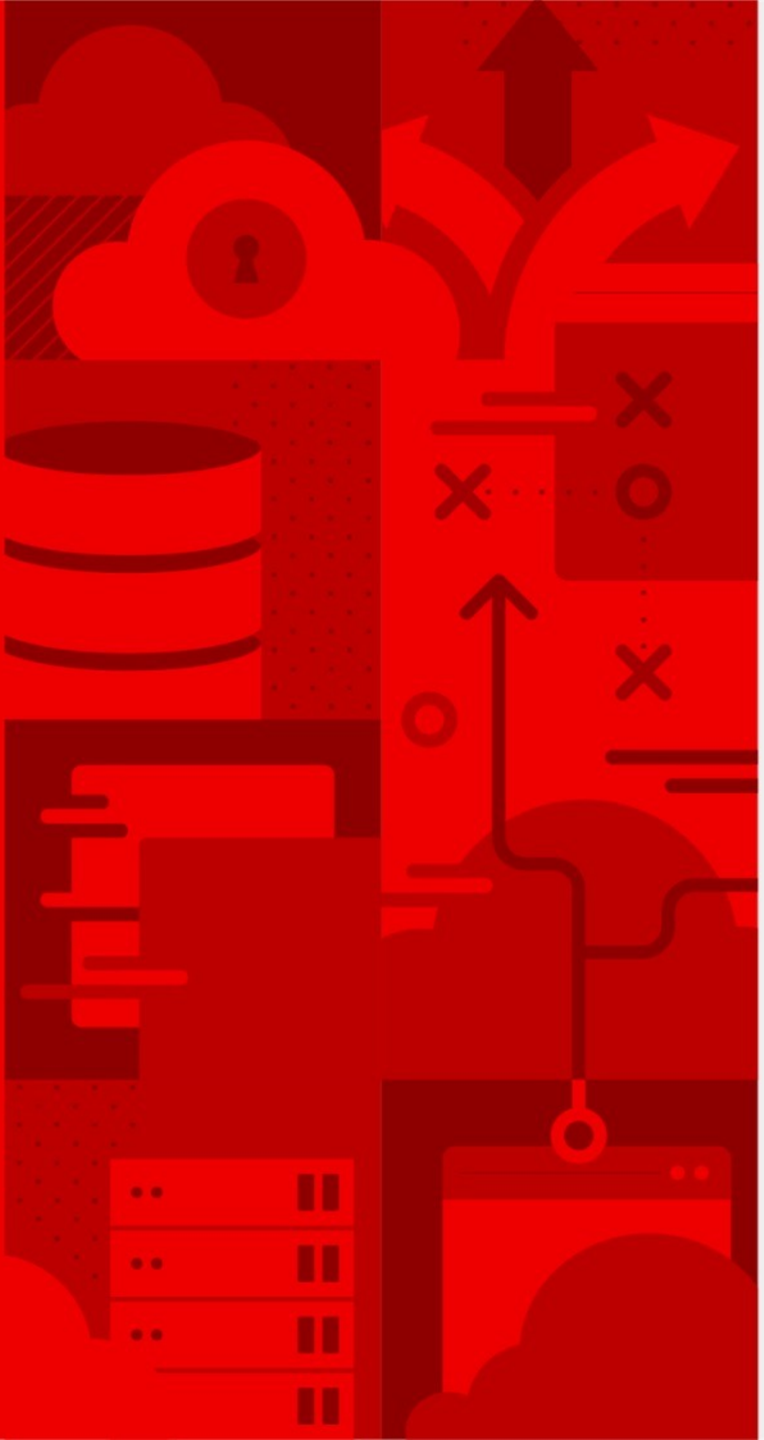
CEPH DAY NETHERLANDS, 2 JULY 2019



# Dynamic Storage Provisioning

With OpenShift, Ceph-CSI and Rook

Niels de Vos <ndevos@redhat.com> | @nixpanic
Software Engineer
Red Hat Storage

# Agenda:

Persistent Storage Basics on OpenShift and Kubernetes

Container Storage Interface (CSI)

Rook and Ceph-CSI

Upcoming enhancements to Ceph-CSI

Red Hat OpenShift

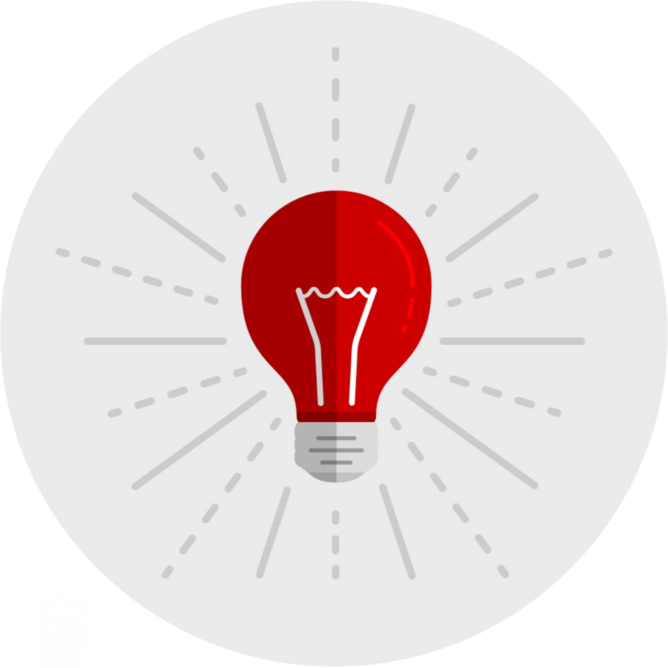
OpenShift is based on Kubernetes

Manages containers and resources

App deployments are described in YAML

Extendible with Custom Resource Definitions

# Components that are involved in provising dynamic provisioned persistent storage

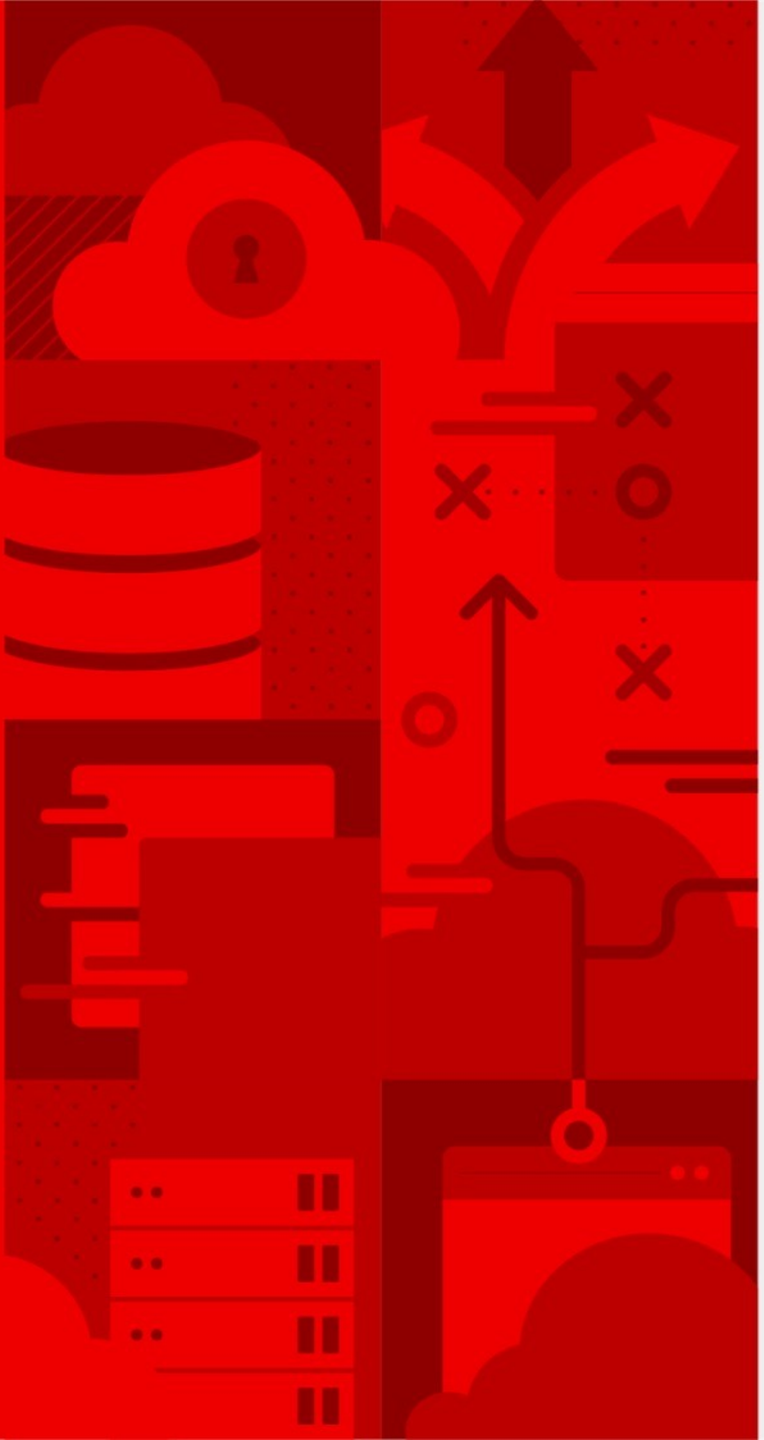


***PersistentVolumeClaim (PVC)***
Request with required resource details and reference to a StorageClass

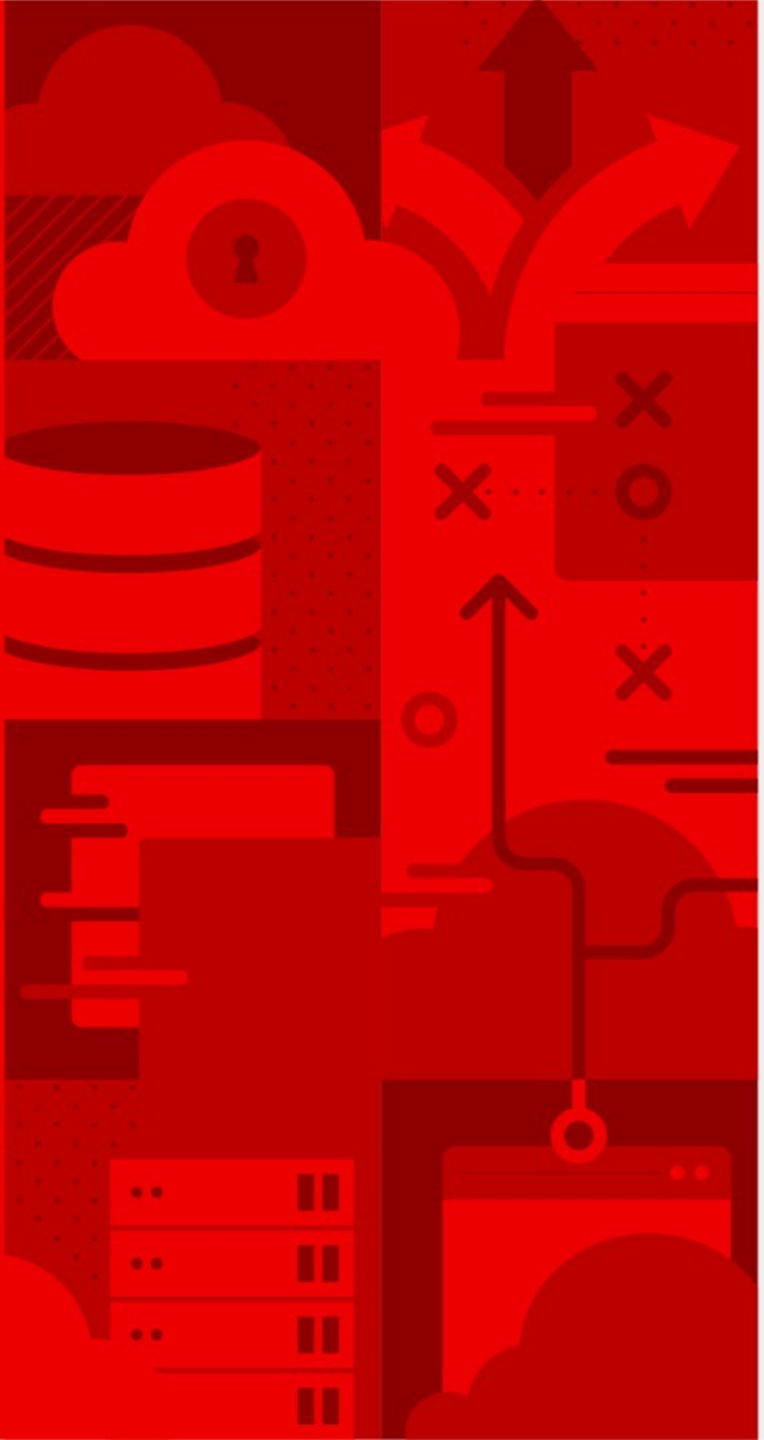
***StorageClass (SC)***
Handler for certain requests, based on storage backend with optional configuration details.

***Provisioner***
Receives the PVC with SC reference, allocates storage on the backend and returns a PV-object.

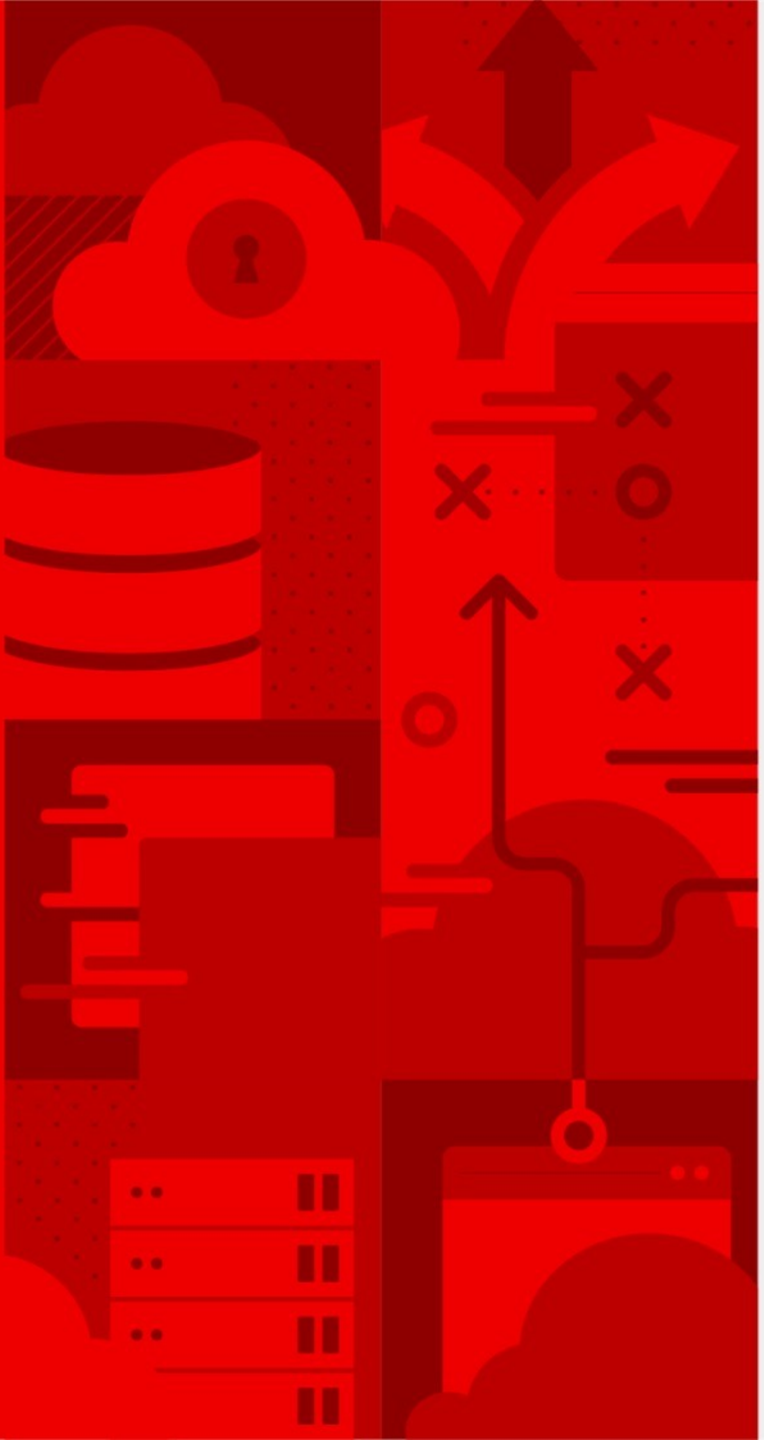
***PersistentVolume (PV)***
Reference to allocated storage on the backend. Contains details on connection information, mount options, ...



```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: rbd-pvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-rbd
```



```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
   name: csi-rbd
provisioner: rbd.csi.ceph.com
parameters:
    clusterID: rook-ceph
    pool: replicapool
    imageFormat: "2"
    imageFeatures: layering
    ...
reclaimPolicy: Delete
```



```
$ oc get pv/pvc-58fb633c-9c16-11e9-9b90-002563e4d5cb -oyaml
apiVersion: v1
kind: PersistentVolume
...
spec:
  accessModes:
  - ReadWriteOnce
  capacity:
    storage: 1Gi
  ...
  csi:
    driver: rbd.csi.ceph.com
    fsType: ext4
    ...
```

# Different Storage Provisioners that Kubernetes can use to configure/provide Persistent Storage

### *Internal*

Maintained in the main k82 repository. Changes need to go through the standard k8s review and testing process. Slow adoption of new storage features.

### *External Storage project*

External project maintained by the SIG-Storage team. Standardized interface with core k8s. Storage focussed review and testing. Release cycles not necessarily tied to k8s releases.

### *CSI*

Shared implementation of storage provisioners that can be used by different Container Orchestrators

# A specification aimed to be(come) an industry standard to enable storage vendors

A specification aimed to be(come) an industry standard that gets adopted by storage vendors and enables them to write a single plugin for a number of container orchestration systems.

CONTAINER
STORAGE
INTERFACE

# Three different services are required when implementing a CSI driver.

Both Controller and Node plugins need to provide the Identity Service. Controllers are mostly a single instance on a master. The Node Service runs on all container hosts that can consume the storage.

### *Identity Service*

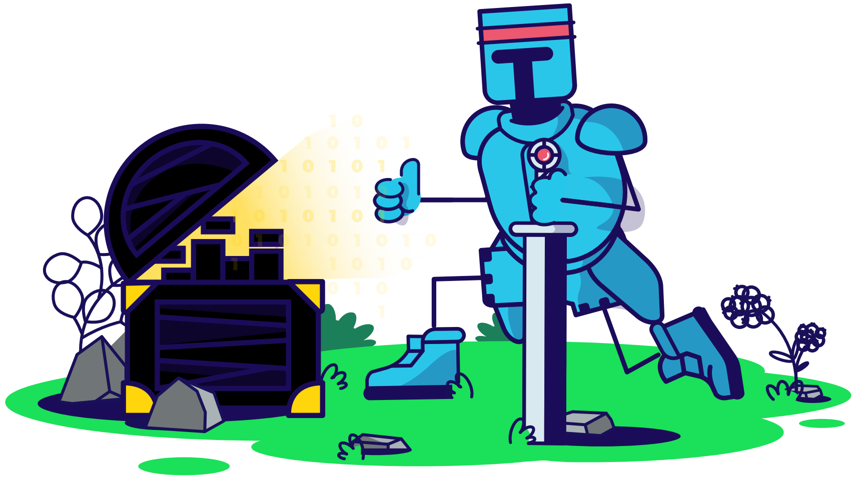
Provides details abouf the CSI driver and the features it (in combination with the backend) supports .

### *Controller Service*

Manages the creation, deletion and related operations to make a volume accessible. Actions related to the storage management that are needed on the backend are normally done by the controller.

### *Node Service*

Responsible for making the volume available on a container host. This can include configuring a (network) block devices and mounting. Also cleanup (unmounting) is a task for the Node Service.



Rook turns distributed storage systems into self-managing, self-scaling, self-healing storage services. It automates the tasks of a storage administrator: deployment, bootstrapping, configuration, provisioning, scaling, upgrading, migration, disaster recovery, monitoring, and resource management.

# Features of Rook as a Storage Operator

*Simple and reliable automated resource management*

*Hyper-scale or hyper-converge your storage clusters*

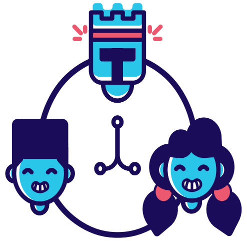
*Efficiently distribute and replicate data to minimize loss*

*Provision, file, block, and object with multiple storage providers*

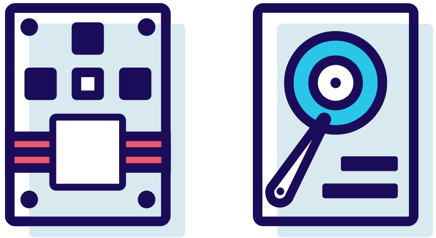
# Features of Rook as a Storage Operator

*Manage open-source storage techologies*

*Easily enable elastic storage in your datacenter*

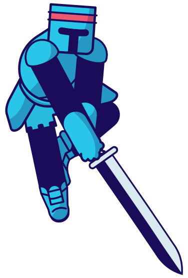


*Open source software released under the Apache 2.0 license*



*Optimize workloads on commodity hardware*

# Main components of a Rook and Ceph-CSI deployment

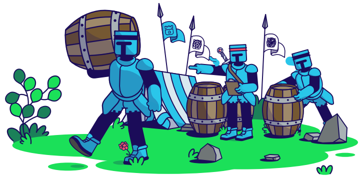


*Deployment*

Definition of the main Rook operator pod, including container, parameters and reference to selected storage backend.

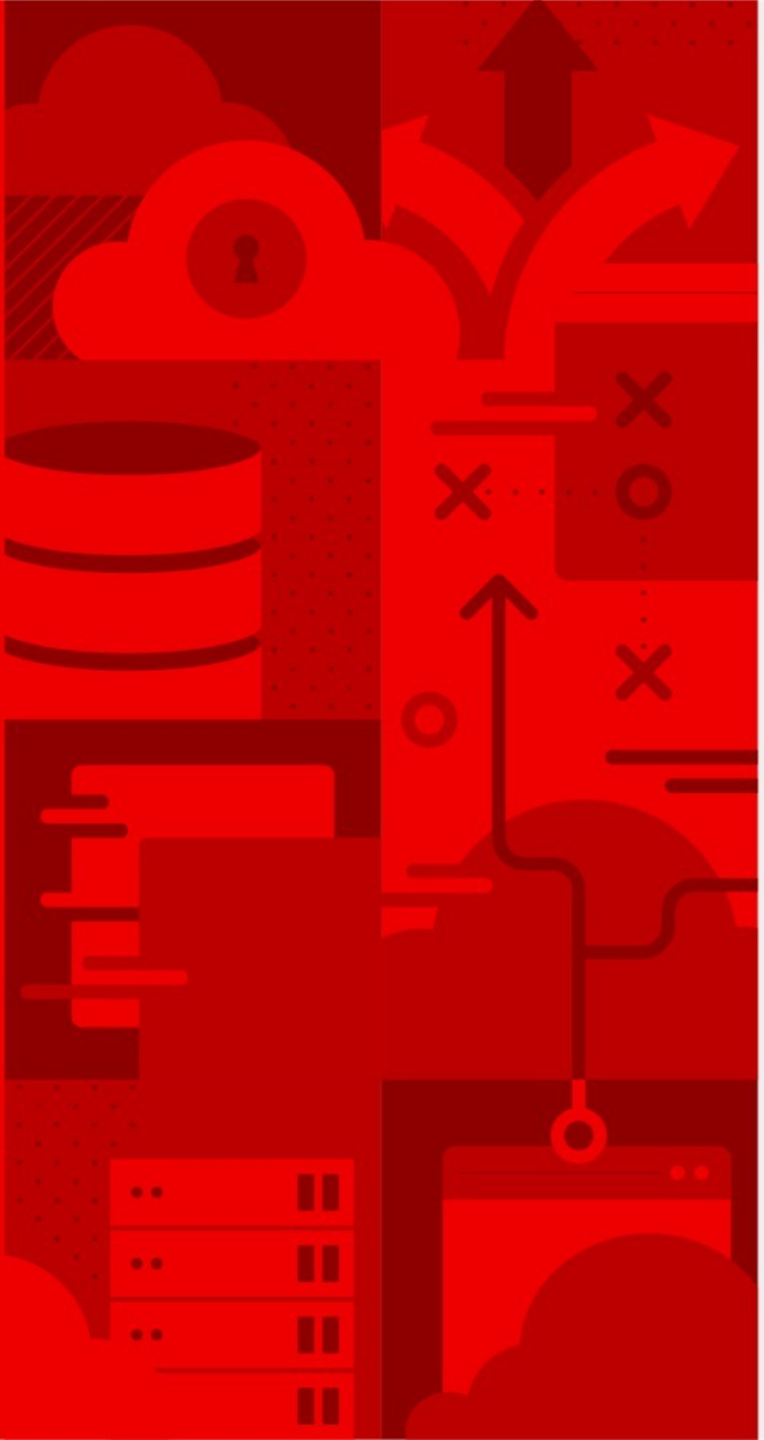
*CephCluster*

Description of the properties of the cluster, like Ceph version, storage nodes and devices to consume.

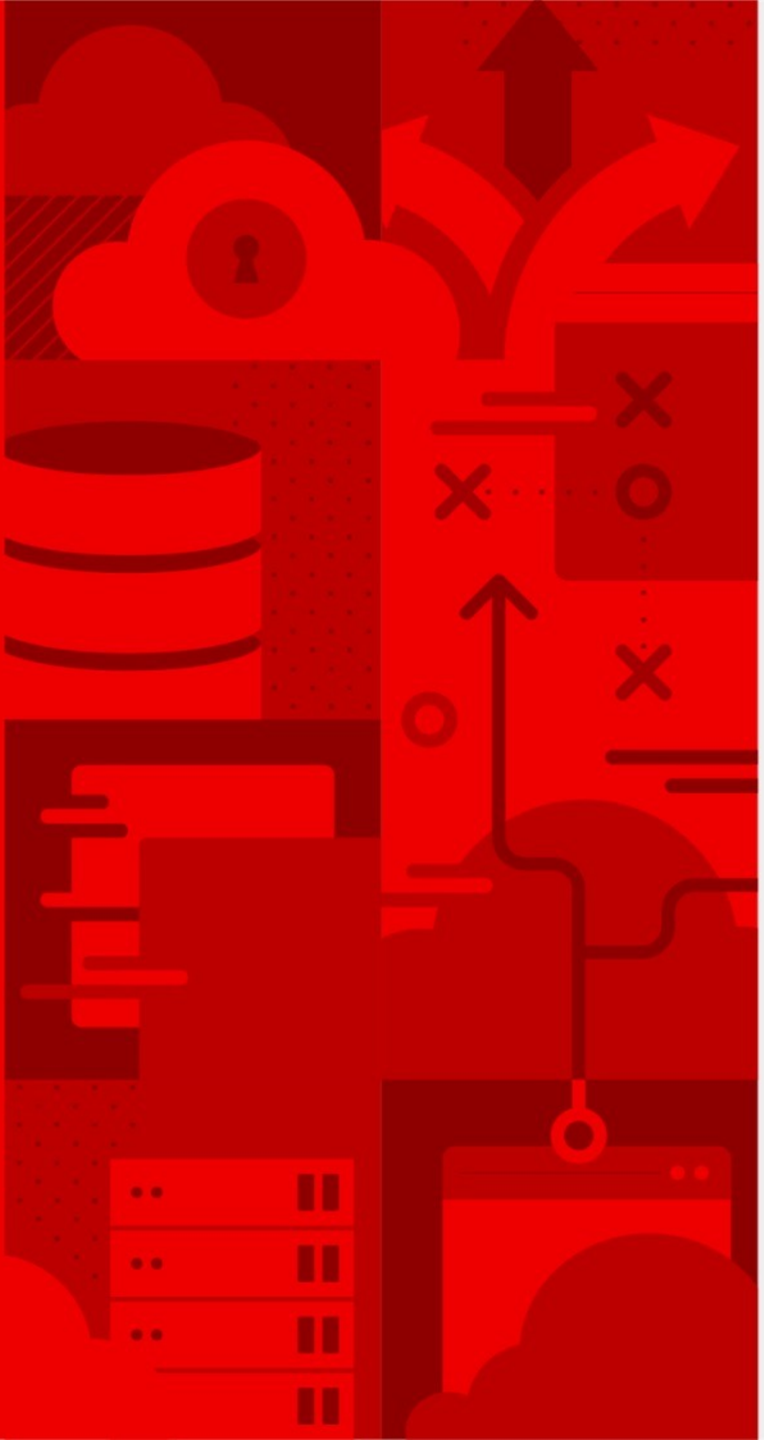


*CephFilesystem/CephBlockPool*

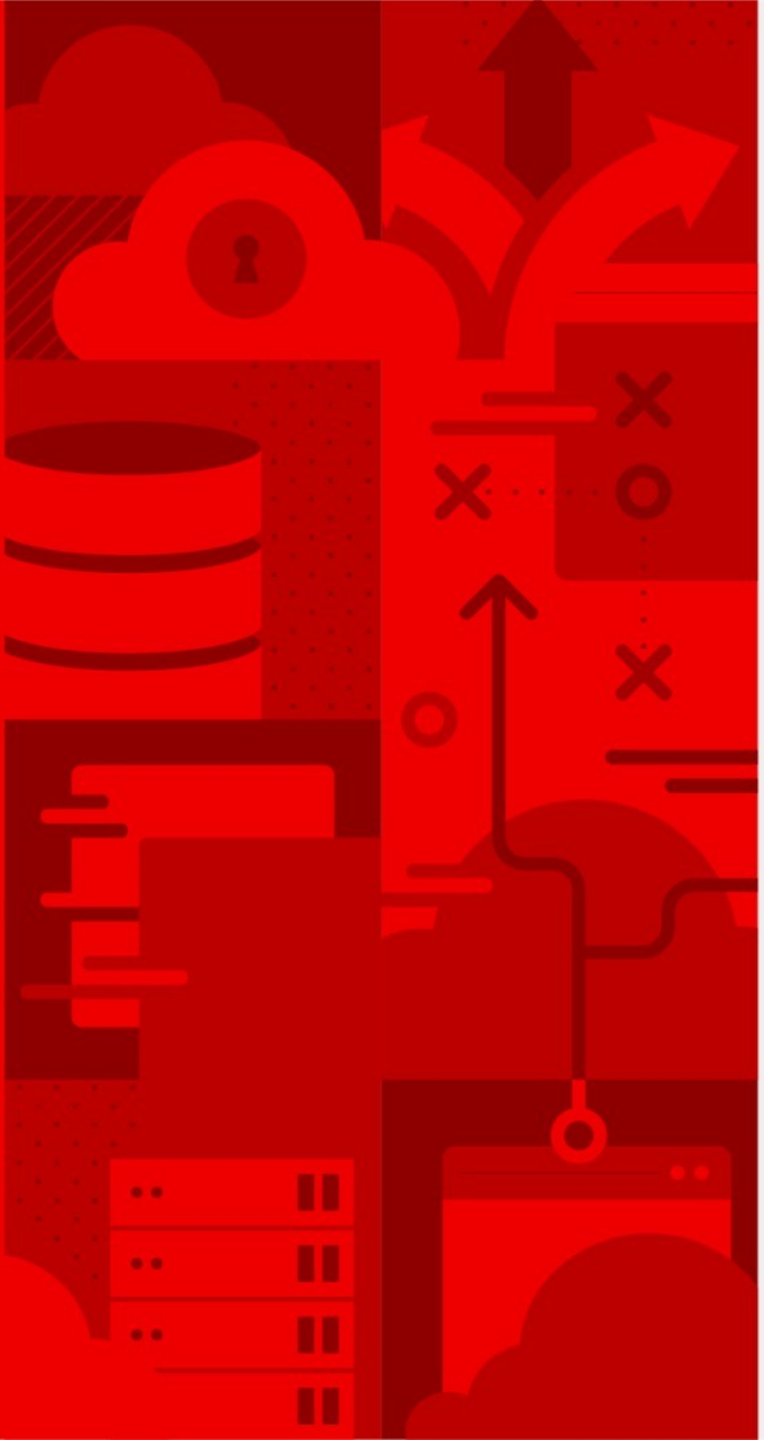
Configuration of CephFS and RBD pools, think of number of metadata servers, replication factor.



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: rook-ceph-operator
  namespace: rook-ceph
  ...
spec:
  replicas: 1
  template:
    spec:
      containers:
      - name: rook-ceph-operator
        env:
        # CSI enablement
        - name: ROOK_CSI_ENABLE_CEPHFS
          value: "true"
```



```
apiVersion: ceph.rook.io/v1
kind: CephCluster
spec:
  cephVersion:
    image: ceph/ceph:v14.2.1-20190430
  mon:
    count: 3
  dashboard:
    enabled: true
  rbdMirroring:
    ...
  ...
```



```
apiVersion: ceph.rook.io/v1
kind: CephCluster
spec:
  cephVersion:
    image: ceph/ceph:v14.2.1-20190430
  mon:
    count: 3
  dashboard:
    enabled: true
  rbdMirroring:
    ...
  ...
```

```
$ oc create -f common.yaml
$ oc create -f rbac/rbd/csi-provisioner-rbac.yaml
$ oc create -f rbac/rbd/csi-nodeplugin-rbac.yaml
$ oc create -f operator-openshift-with-csi.yaml
$ oc create -f cluster.yaml
$ oc create -f pool.yaml
$ oc create -f rbd/secret.yaml
$ oc create -f rbd/storageclass.yaml
$ oc create -f rbd/pvc.yaml
```

# Some of the notible changes that have recently been added or are currently being developed.

ceph

CONTAINER
STORAGE
INTERFACE

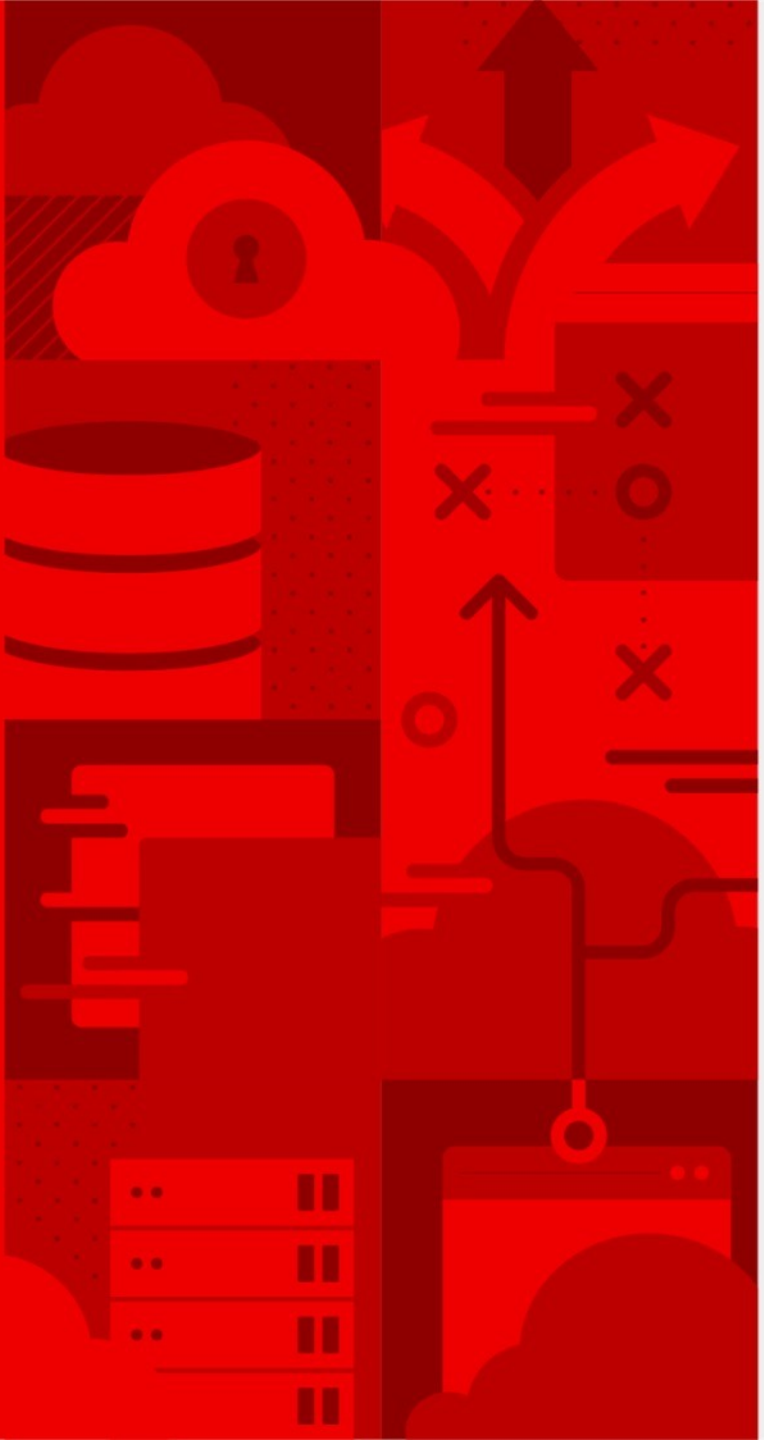
***Single Ceph-CSI container image***
All CSI related components in one image, parameterized for selecting functionality.

***Cloning & Snapshots for RBD***
Not yet fully supported by Kubernetes-CSI.

***Volume Expansion***
Not yet fully supported by Kubernetes-CSI.

***CephFS sub-volumes***
Currently subdirectories per volume, ceph-mgr enhancement to create sub-volumes is coming soon.

***Snapshots for CephFS***
Per sub-volume snapshots is in the planning.



# Thank you

ROOK

Red Hat OpenShift

ceph

CONTAINER STORAGE INTERFACE

Red Hat