

Rook Intro & Ceph Deep Dive

Rook maintainers:

Travis Nielsen, Red Hat

Sébastien Han, Red Hat

Blaine Gardner, Red Hat

Alexander Trost, Cloudical

November 2020



Rook is Graduated!



- CNCF graduated project!
 - Sandbox: January 2018
 - Incubation: September 2018
 - Graduation: October 2020
- Thank you!
 - Rook community, CNCF committee, and Rook maintainers





Rook is Graduated!

- Large scale production deployments
- Security review by a third party
- Project governance that is vendor neutral



Kubernetes Storage Challenges



Storage Challenges

- Kubernetes is a platform to manage distributed apps
 - Traditionally stateless
- Reliance on external storage
 - Not portable
 - Deployment burden
- Reliance on cloud provider managed services
 - Vendor lock-in
- Day 2 operations - who is managing the storage?



What is Rook?



What is Rook?

- Storage made available inside your Kubernetes cluster
- Kubernetes Operators and CRDs
- Automated management
 - Deployment, configuration, upgrading, ...
- Consume like any other K8s storage
 - Storage classes, PVCs, etc.
- Open Source (Apache 2.0)





Storage Providers

- Stable
 - Ceph
- Alpha
 - NFS, Cassandra, YugabyteDB, CockroachDB
- Deprecated
 - EdgeFS: A replacement is being developed



v1.5 released



7.8K+ Github Stars



180M+ Downloads



295+ Contributors



CNCF Graduated Project



Rook + Ceph

What is Ceph?



- Open Source
- Distributed Software-Defined Storage solution
 - Block
 - Shared File System
 - Object (S3 compliant)

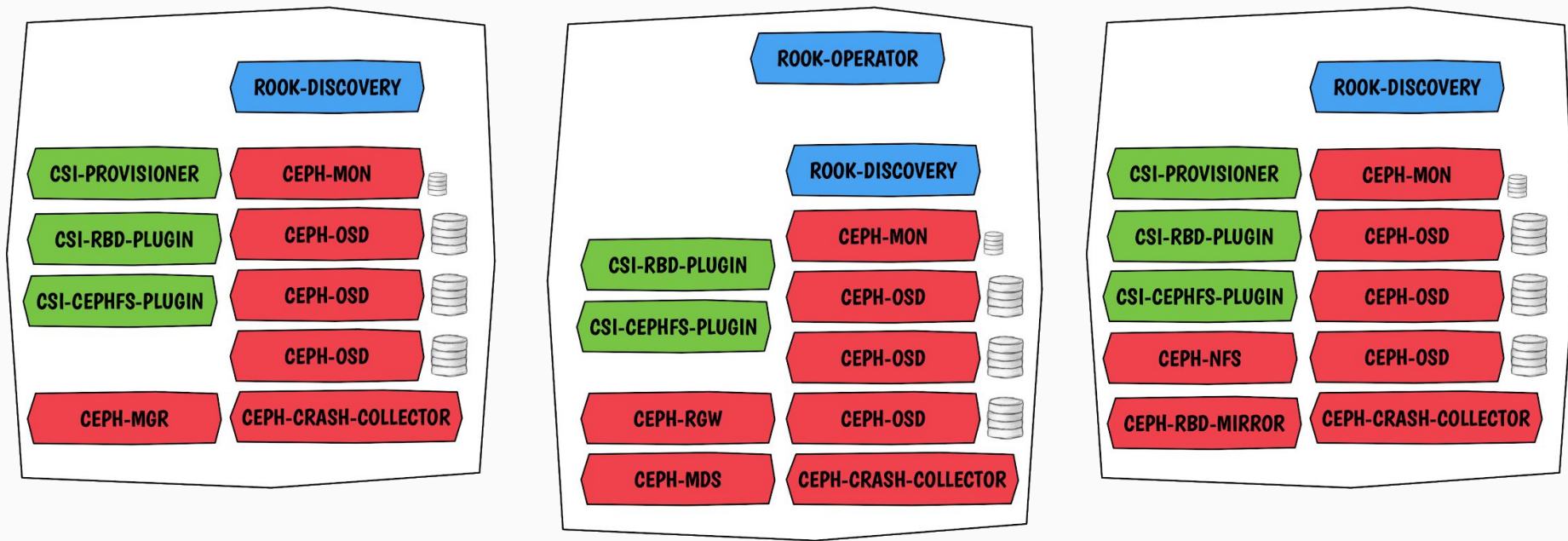


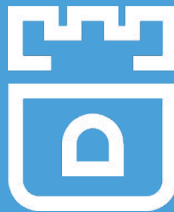
Architectural Layers

- Rook
 - The operator owns the **management** of Ceph
- Ceph-CSI
 - The CSI driver dynamically **provisions** and **mounts** storage to the app pod
- Ceph
 - **Data** layer

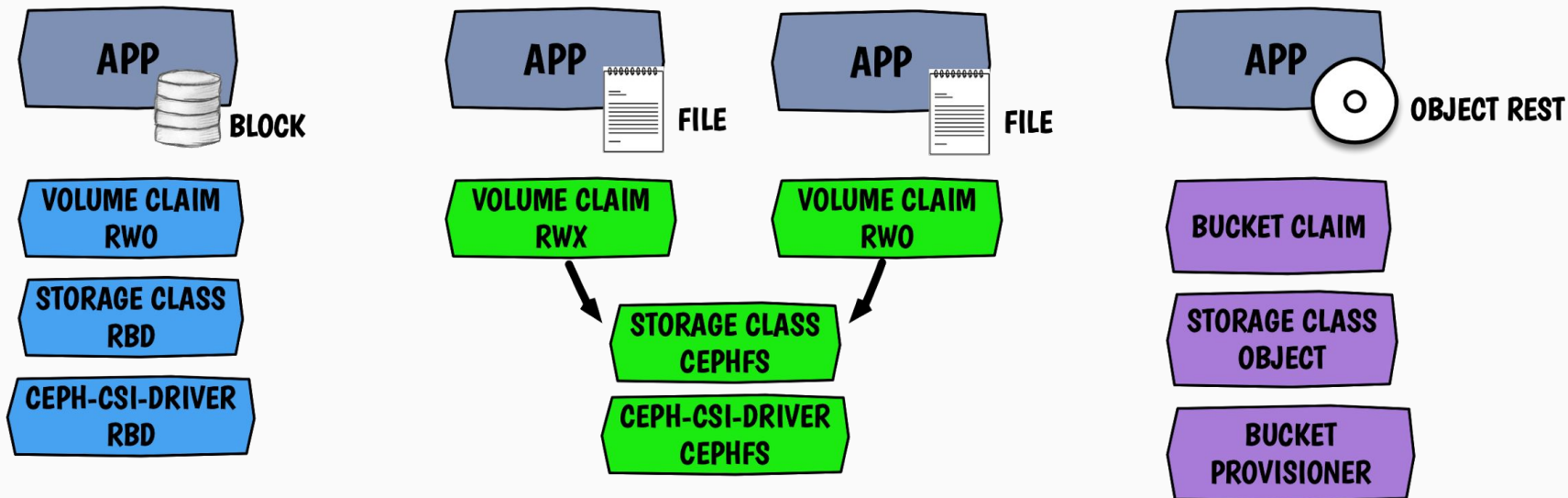


Layer 1: Rook Management



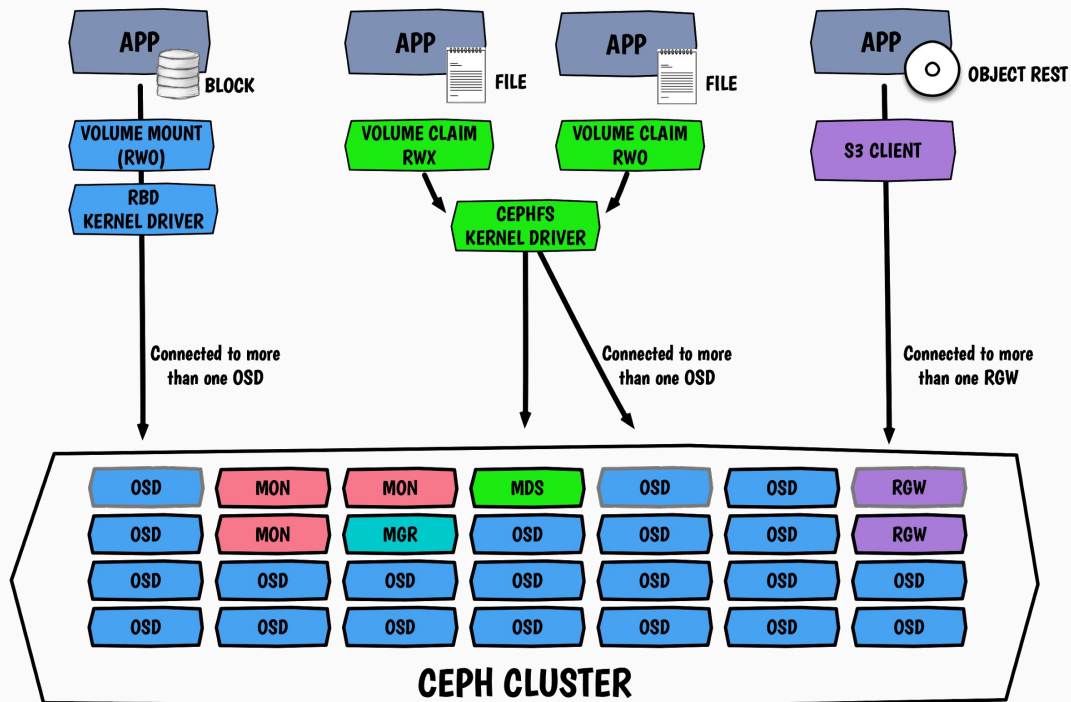


Layer 2: CSI Provisioning





Layer 3: Ceph Data Path





Getting Started



Installing Ceph is simple!

- Create the authorization (RBAC)
 - `kubectl create -f common.yaml`
- Create the Ceph Operator
 - `kubectl create -f operator.yaml`
- Create the CephCluster CR
 - `kubectl create -f cluster.yaml`

```
apiVersion: ceph.rook.io/v1
kind: CephCluster
metadata:
  name: rook-ceph
  namespace: rook-ceph
spec:
  cephVersion:
    image: ceph/ceph:v15.2.4
  dataDirHostPath: /var/lib/rook
  mon:
    count: 3
  storage:
    useAllNodes: true
    useAllDevices: true
```



Request Storage for Apps

- Admin creates a StorageClass
- Create a Persistent Volume Claim
- Create an application pod

```
apiVersion: v1
kind: Pod
metadata:
  name: csirbd-demo-pod
spec:
  containers:
    - name: web-server
      image: nginx
      volumeMounts:
        - name: mypvc
          mountPath: /var/lib/www/html
  volumes:
    - name: mypvc
      persistentVolumeClaim:
        claimName: rbd-pvc
        readOnly: false
```




Key Features

Environments



- Bare metal
 - Bring your own hardware
- Cloud providers
 - Expand cloud provider storage with Rook capabilities



Ceph in a Cloud Environment

- Consistent Storage Platform wherever K8s is deployed
- Overcome shortcomings of the cloud provider's storage
 - Storage across AZs
 - Slow failover times (seconds instead of minutes)
 - Limitations of number of PVs per node (many more than ~30)
 - Perf characteristics of large volumes
- Ceph Monitors and OSDs run on PVCs
 - No need for direct access to local devices



Configurable for Cluster Topologies

- Rook can be easily customized across cluster topologies
- Failure domains: High availability and durability
 - Spread Ceph daemons and data across failure domains
- Rook can be deployed on specific nodes if desired
 - Node affinity, taints/tolerations, etc

Ceph CSI Driver



- Ceph CSI 3.1 Driver is deployed by default with v1.5
- Dynamic provisioning of RWO/RWX/ROX modes for Block and Filesystem
- Volume expansion
- Snapshots and clones (beta)
- Flex driver is still available, but support is limited



Upgrading is automated

- To upgrade Rook, update the Operator version (operator.yaml)
 - Minor releases require steps as documented in the upgrade guide

```
image: rook/ceph:v1.5.1
```

- To upgrade Ceph, update the cluster CR version (cluster.yaml)
 - Rook handles intricacies of Ceph version upgrades

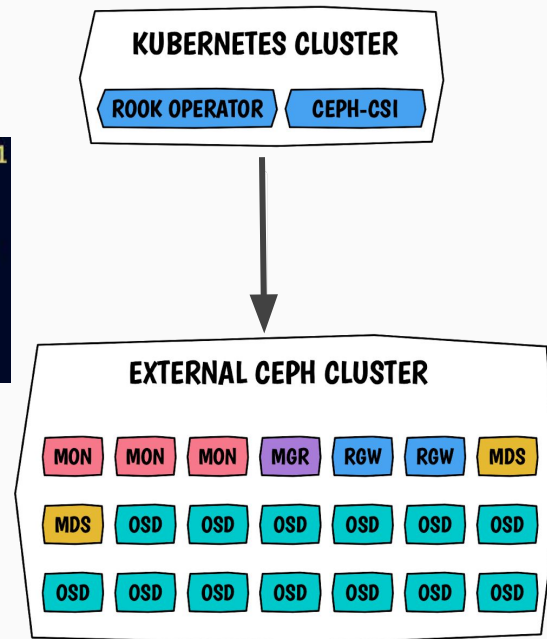
```
image: ceph/ceph:v15.2.6
```



External Cluster Connection

- Connect to a Ceph cluster that you've configured separately from Kubernetes
- Inject into Kubernetes:
 - Ceph Monitors list
 - Ceph keyring
 - Ceph cluster FSID
- Create the cluster-external CR

```
apiVersion: ceph.rook.io/v1
kind: CephCluster
metadata:
  name: rook-ceph-external
spec:
  external:
    enable: true
```





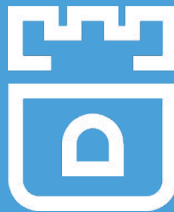
Object Bucket Provisioning

- Define a Storage Class for object storage
- Create an “object bucket claim”
 - The operator creates a bucket when requested
 - Similar pattern to a Persistent Volume Claim (PVC)
- Container Object Storage Interface (COSI)
 - Kubernetes enhancement proposal merged upstream



Rook v1.5 Features

Nov 2020



Encryption with Key Management

- Encryption supported since 1.4 using K8s secrets
- KMS HashiCorp Vault integration to store OSD encryption keys
- Token-based authentication only
- Additional KMS support planned for future releases



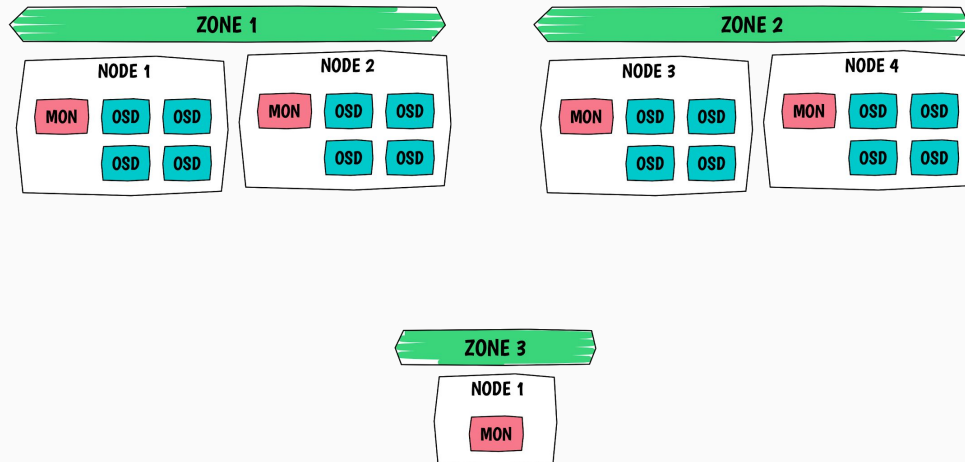
Mirroring of Block Data

- Mirror your block data to another Ceph cluster
- Application disaster recovery scenarios
- Asynchronous replication with Ceph RBD mirroring
- Automatic configuration of mirroring peers



Stretched Kubernetes Cluster

- Two zones available for storage
- Ceph has greater resiliency to network partitions



How to get involved?

Website

<https://rook.io/>

Documentation

<https://rook.io/docs/rook/v1.5/>

Slack

<https://rook-io.slack.com/>

Contributions

<https://github.com/rook/rook>

Twitter

@rook_io

Community Meeting

<https://github.com/rook/rook/community-meeting>

Thank you!

<https://rook.io/>

