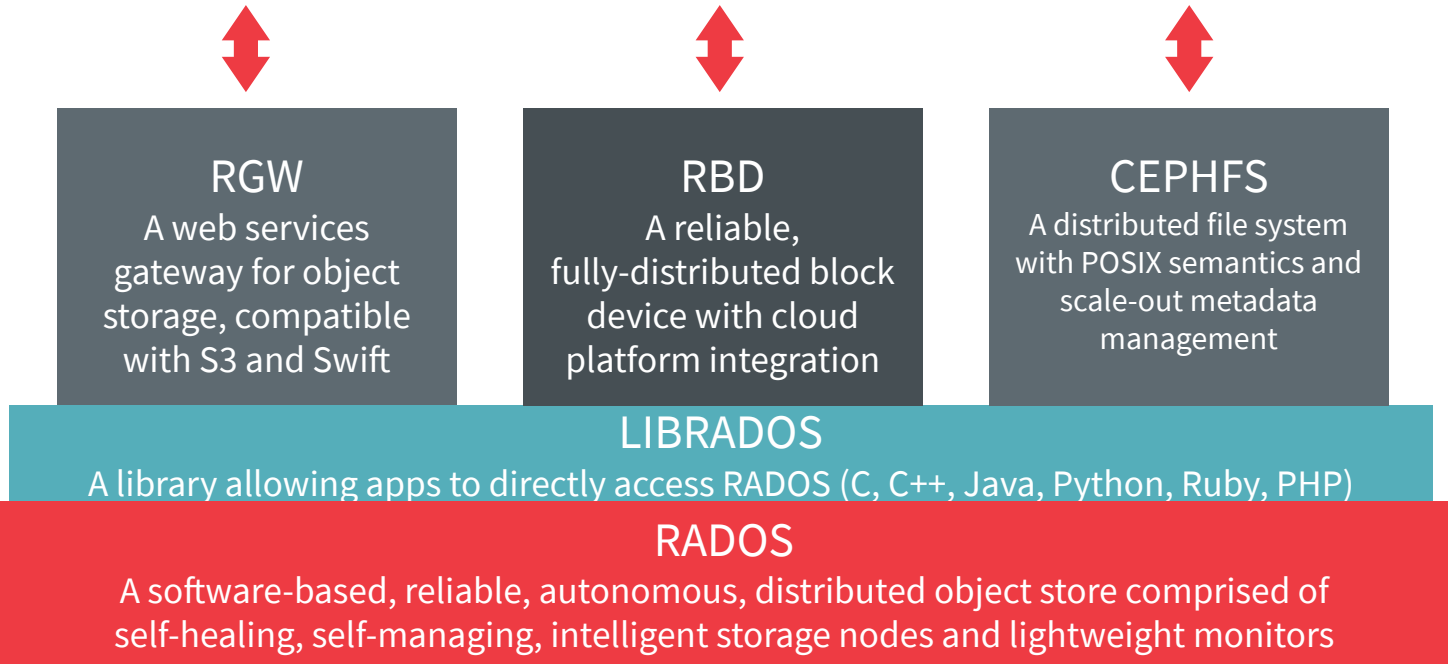
Rook, Ceph and ARM

The caffeinated tutorial

Federico Lucifredi, Sébastien Han — Red Hat


ceph

CEPH ARCHITECTURE



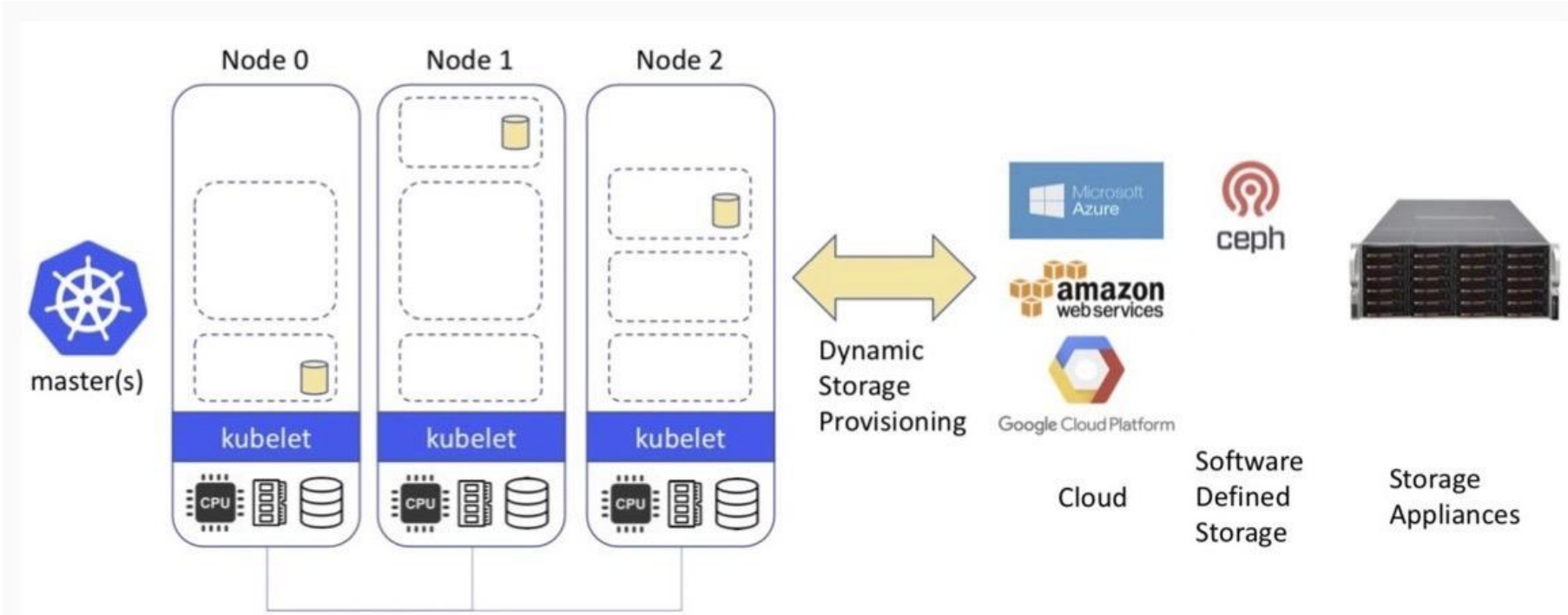
WHY IS STORAGE HARD?

STORAGE IN KUBERNETES CONTAINER ORCHESTRATION

STORAGE IN KUBERNETES

- K8s abstracts away the infrastructure it manages
- Dynamic environment
  - Balancing load
  - Rebuilding pods (healing)
- Ephemeral storage design

STORAGE FOR KUBERNETES



Volume plugins allow external storage solutions to provide storage to your apps

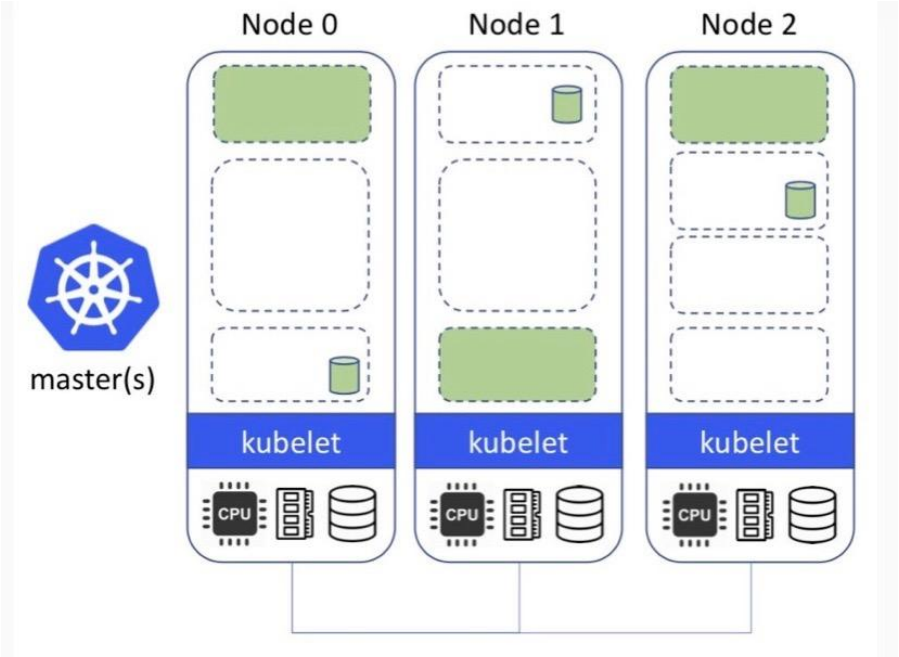
LIMITATIONS

- Not portable: requires these services to be accessible
- Deployment burden of external solutions
- Vendor lock-in due to using provider managed services

STORAGE ON KUBERNETES

Kubernetes can manage our storage solution

- Highly portable applications (including storage dependencies)
- Dedicated K8s storage cluster also possible



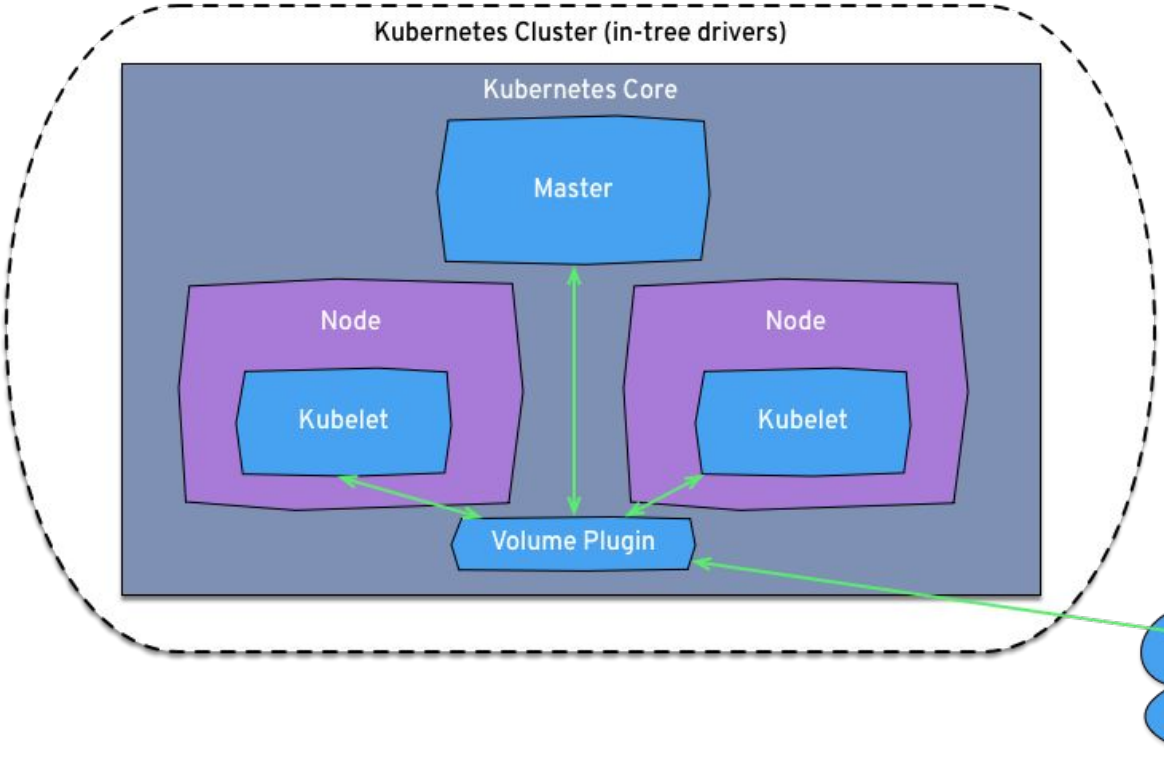
THE CONTAINER STORAGE INTERFACE

(CSI)

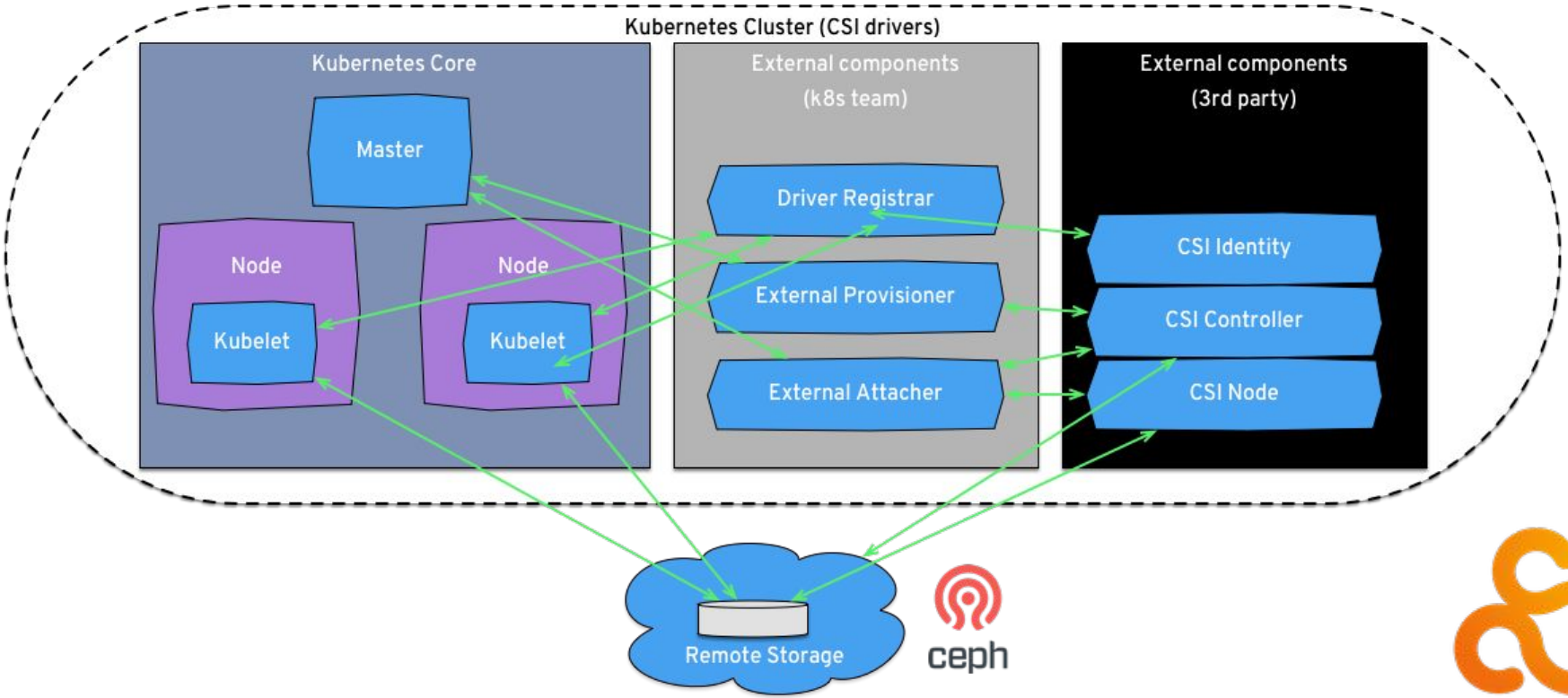
CSI

- Prior to CSI, it was challenging to add support for new volume plugins to Kubernetes.
- Volume plugins were “in-tree”, third-party storage code caused reliability and security issues in core Kubernetes binaries
- With the introduction of CSI, storage can now be treated as another workload to be containerized and deployed on a Kubernetes cluster.
- Using CSI, third-party storage providers can write and deploy plugins exposing new storage systems in Kubernetes without touching the core Kubernetes code.

CSI

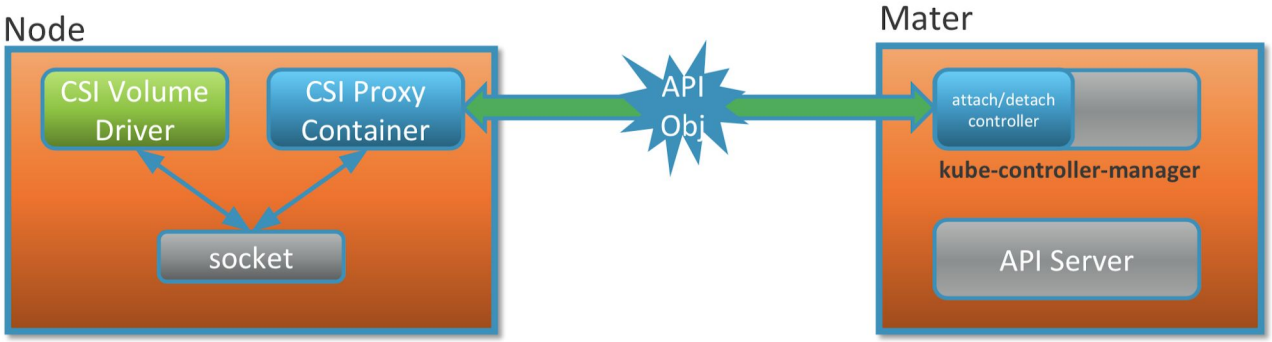

Kubernetes Cluster (in-tree drivers)
Kubernetes Core
Master
Node
Kubelet
Node
Kubelet
Volume Plugin
Remote Storage
ceph

CSI


Kubernetes Cluster (CSI drivers)
Kubernetes Core
Master
Node
Kubelet
Node
Kubelet
External components
(k8s team)
Driver Registrar
External Provisioner
External Attacher
External components
(3rd party)
CSI Identity
CSI Controller
CSI Node
Remote Storage
ceph

CSI

- A new in-tree CSI Volume plugin(K8s) + out-of-tree CSI Volume driver (3rd party)
- Communication channel via a Unix Domain Socket(UDS) created by 3rd Volume Driver



The socket file also called a 'EndPoint' in form of like:
/var/lib/kubelet/plugins/rook-ceph/csi.sock

CEPH CSI

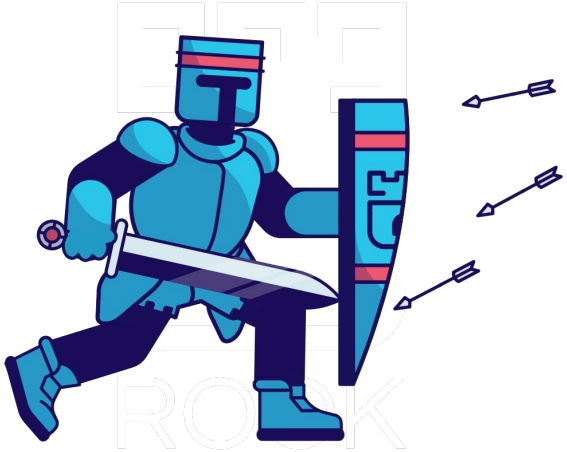
- Ceph CSI plugin allows dynamically provisioning Ceph volumes and attaching them to workloads.
- Relies on Kubernetes CSI spec (v3.0 and v1.0)
- Integrated in Rook 1.0
  - https://github.com/ceph/ceph-csi/

Storage access modes:

- RWO - ReadWriteOnce: the volume can be mounted as read-write by a single node
- ROX - ReadOnlyMany: the volume can be mounted read-only by many nodes
- RWX - ReadWriteMany: the volume can be mounted as read-write by many nodes

ROOK

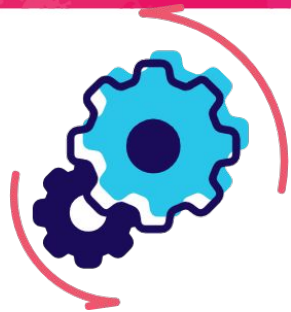
PROVIDE BEST CEPH STORAGE EXPERIENCE IN KUBERNETES



ROOK + CEPH

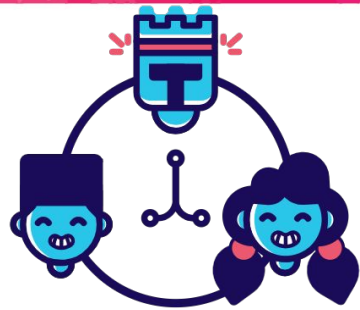
- Rook is bringing Ceph and Kubernetes together
- More than +5200 Github stars, 21M docker pools and 140+ contributors.
- Accepted as the CNCF's first storage project
- Rook has recently reached incubation stage
- Just released 1.0
- Open Source (Apache 2.0)

STORAGE FRAMEWORK



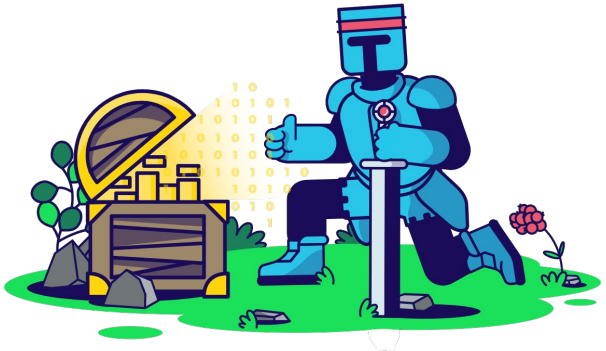
- Automated Orchestration of Ceph
  - Deployment
  - Configuration
  - Scaling
  - Upgrading
  - Recovering
  - Monitoring
- Providing persistent storage to applications
  - Ceph-CSI (Container Storage Interface)
  - Attaching / detaching volumes to pods

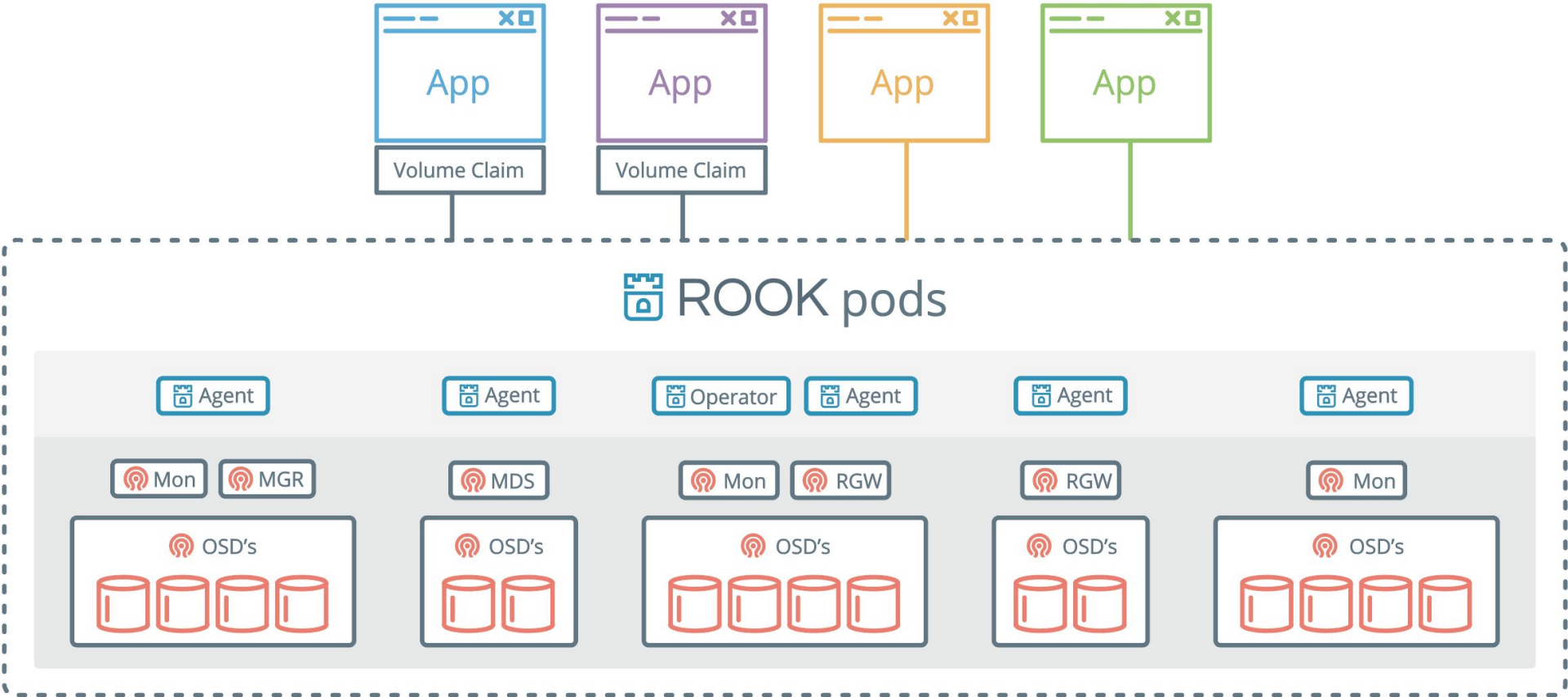
ROOK OPERATOR



- Implements the **Operator Pattern** for Ceph
  - Existed before the operator-sdk or kubebuilder
- User defines ***desired state*** for the storage cluster
  - Achieved by injecting a CR (Custom Resource)
- Operator:
  - **Observes** - Watch for changes in state and health
  - **Analyzes** - Determine differences to apply
  - **Acts** - Apply changes to the cluster

ARCHITECTURE



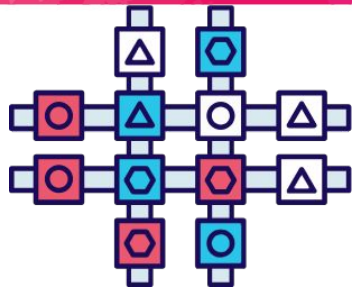

App
App
App
App
Volume Claim
Volume Claim
ROOK pods
Agent
Agent
Operator
Agent
Agent
Agent
Mon
MGR
MDS
Mon
RGW
RGW
Mon
OSD's
OSD's
OSD's
OSD's
OSD's

CLUSTER CUSTOM RESOURCE


```
apiVersion: ceph.rook.io/v1
kind: CephCluster
metadata:
 name: rook-ceph
 namespace: rook-ceph
spec:
 cephVersion:
   image: ceph/ceph:v14.2
 mon:
   count: 3
 dashboard:
   enabled: true
 storage:
   useAllNodes: true
   useAllDevices: true
```

User's desired state

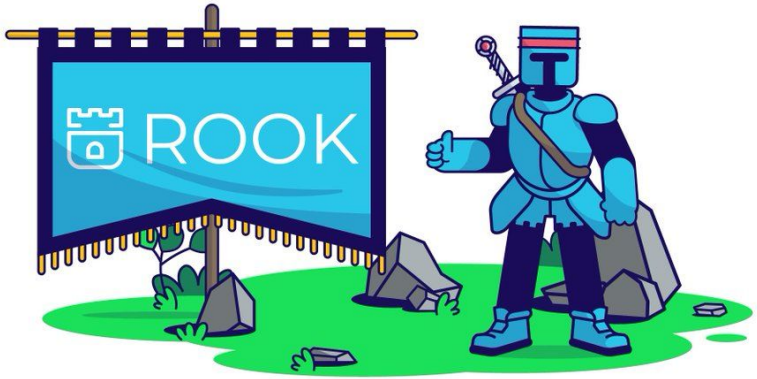
AVAILABLE CRDs



Rook's Custom Resource Definitions (CRDs):

- CephCluster: represents a Ceph Cluster
- CephBlockPool: represents a Ceph Block Pool
- CephFilesystem: represents a Ceph Filesystem interface
- CephNFS: represents a Ceph NFS interface.
- CephObjectStore: represents a Ceph Object Store.
- CephObjectStoreUser: represents a Ceph Object Store User.

DEMO TIME
MONITORS FAILOVER



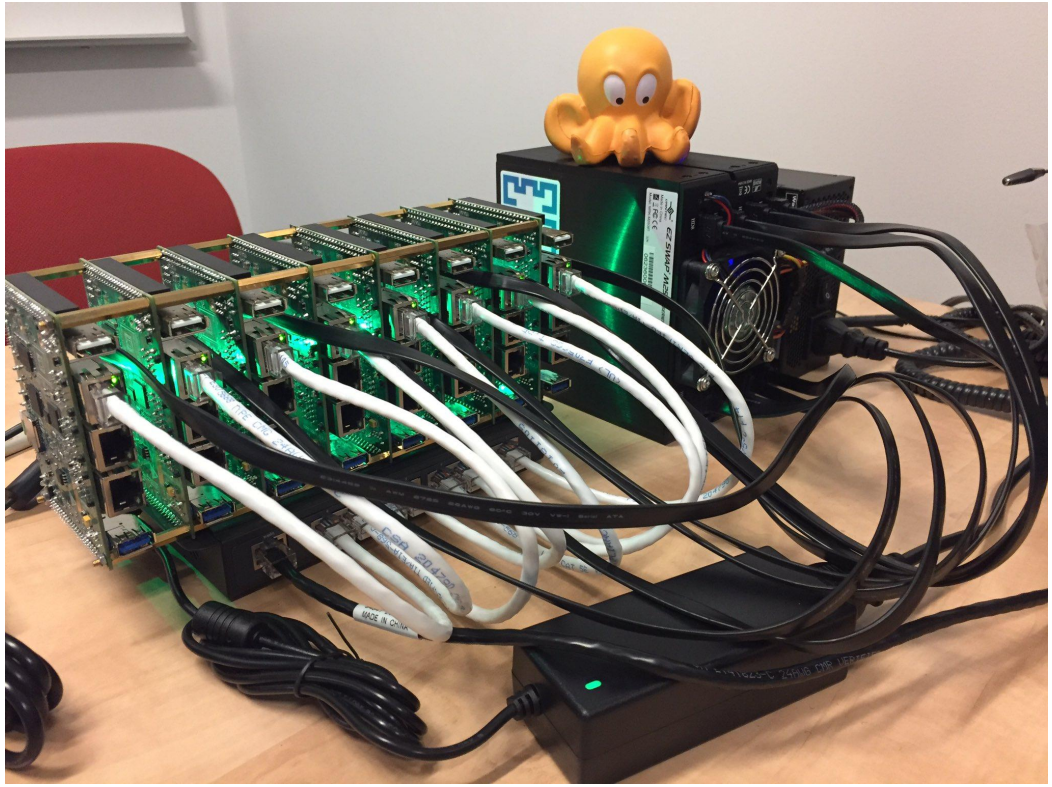
HARDWARE

Globalscale Espressobin v5

- Marvell Armada 3700LP
  - Dual-core Cortex A53
  - 1 GHz
- 1GB DDR3 RAM
- Sata Power + 3.0 port
- Mini PCIe slot

Intel SSD5 545 256GB

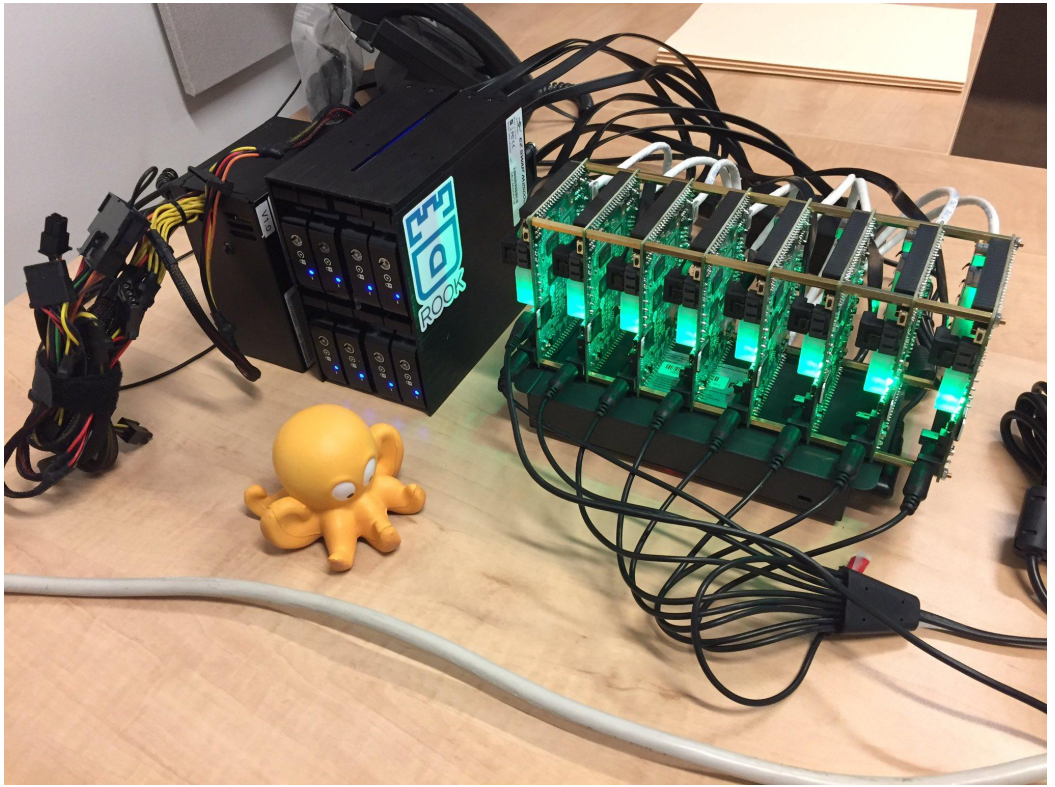
- 6Gb/s SATA
- Seq 550/500 MB/s RW (max)



SOFTWARE

uBoot
Linux 4.19.20
Docker 18.09
Kubernetes 1.14.2
… lots of pesky details.

```
$ kubectl create -f common.yaml
$ kubectl create -f operator.yaml
$ kubectl create -f cluster.yaml
```

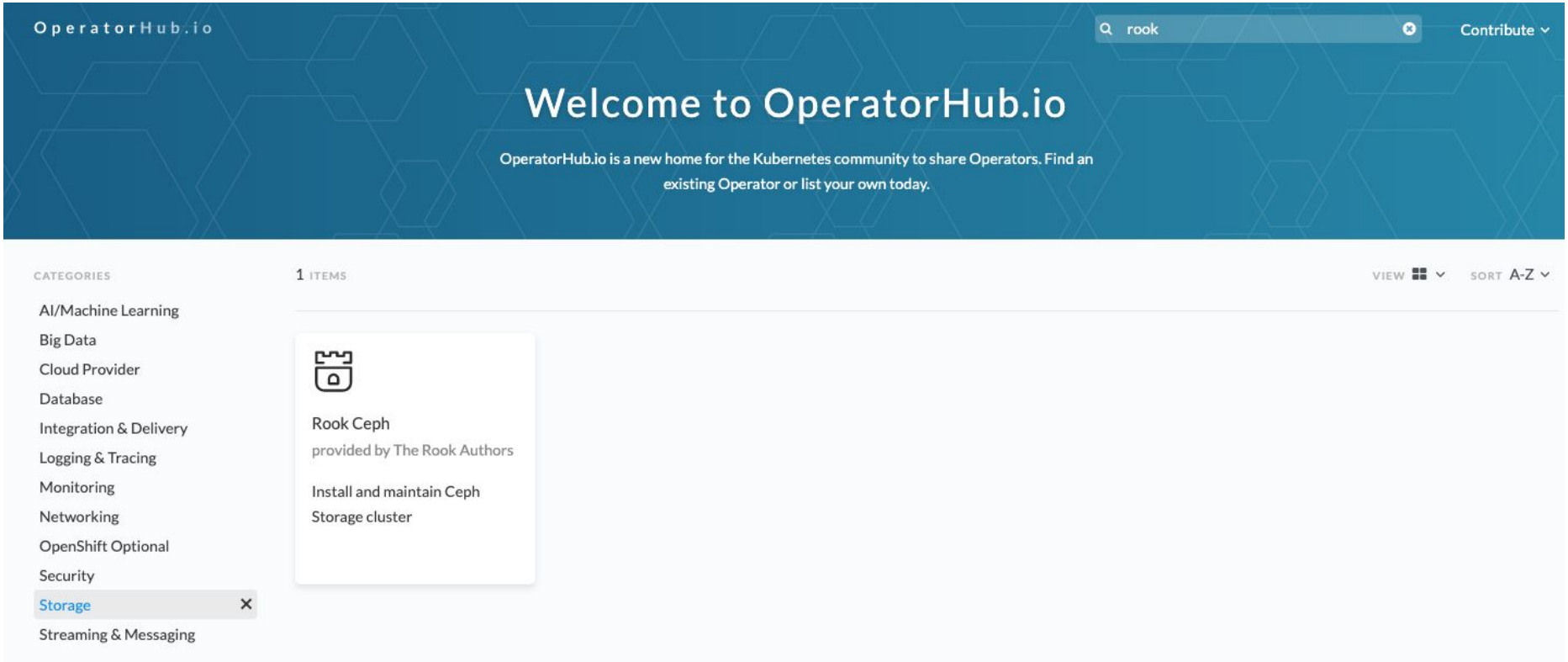


FUTURE OF ROOK

FUTURE WORK

- External cluster support
  - Consume existing Ceph storage cluster that were not deployed with Rook
  - Early steps for taking over a Ceph cluster that wasn't deployed by Rook
- Integration with Multus
  - Attach multiple physical interfaces to a pod
  - Removes the need of host Networking
  - More secure, more control
- Dynamic Volume Provisioning on Cloud provider
  - Allows smoother run Cloud platforms like AWS/GKE/AKS
- Bucket dynamic provisioner
  - `ObjectBucketClaim / ObjectBucket`

One more thing...


OperatorHub.io
rook
Contribute
Welcome to OperatorHub.io
OperatorHub.io is a new home for the Kubernetes community to share Operators. Find an existing Operator or list your own today.
CATEGORIES
AI/Machine Learning
Big Data
Cloud Provider
Database
Integration & Delivery
Logging & Tracing
Monitoring
Networking
OpenShift Optional
Security
Storage
Streaming & Messaging
1 ITEMS
VIEW
SORT A-Z
Rook Ceph
provided by The Rook Authors
Install and maintain Ceph Storage cluster

Acknowledgements

KubeCon | CloudNativeCon

Europe 2019

THANK YOU

@0xF2

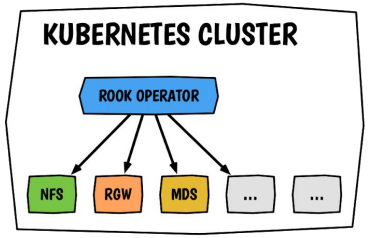
@leseb_

BACKUP SLIDES (ha!)

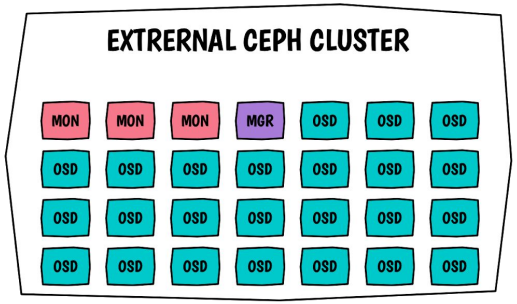
External cluster

- Ability to **consume** existing Ceph storage cluster that were not deployed with Rook
- Rook does **not** manage the cluster
- Bootstrap stateless daemons in Kubernetes but leave the rest in place on the existing cluster
- Different Storage Classes for certain clusters

External cluster




```yaml
kind: CephCluster
spec:
  external: true
  fsid: 1a394bf1-b42b-4dc9-94e7-85848fc750fd
  monitors: 192.168.0.1,192.168.0.2,192.168.0.3
  adminSecret: QVFEMGdwdGNENFNpQmhBQXZYVjJJY2lTK1p4eG9vN3E0WnlvUWc9PQo=
  name: rook-ceph-external
```



Major changes with 1.0

- Auto-scale when plugging a new disk
- Watch for new storage node and increase capacity automatically
- Upgrade mechanism enhancement
- Expose more CR’s details (Ceph health)
- More control over logging (enable/disable on the fly)
- Better maintenance mode
- Better resources control (requests and limits)

Terminology

- **CRD**: Custom Resource Definition; Schema Extension to Kubernetes API
- **CR**: Custom Resource; One record/instance/object, conforming to a CRD
- **OPERATOR**: Daemon that watches for changes to resources
- **STORAGE CLASS**: “class” of storage service
- **PVC**: Persistent Volume Claim, attach persistent storage to a pod
- **POD**: a group of one or more containers managed by Kubernetes

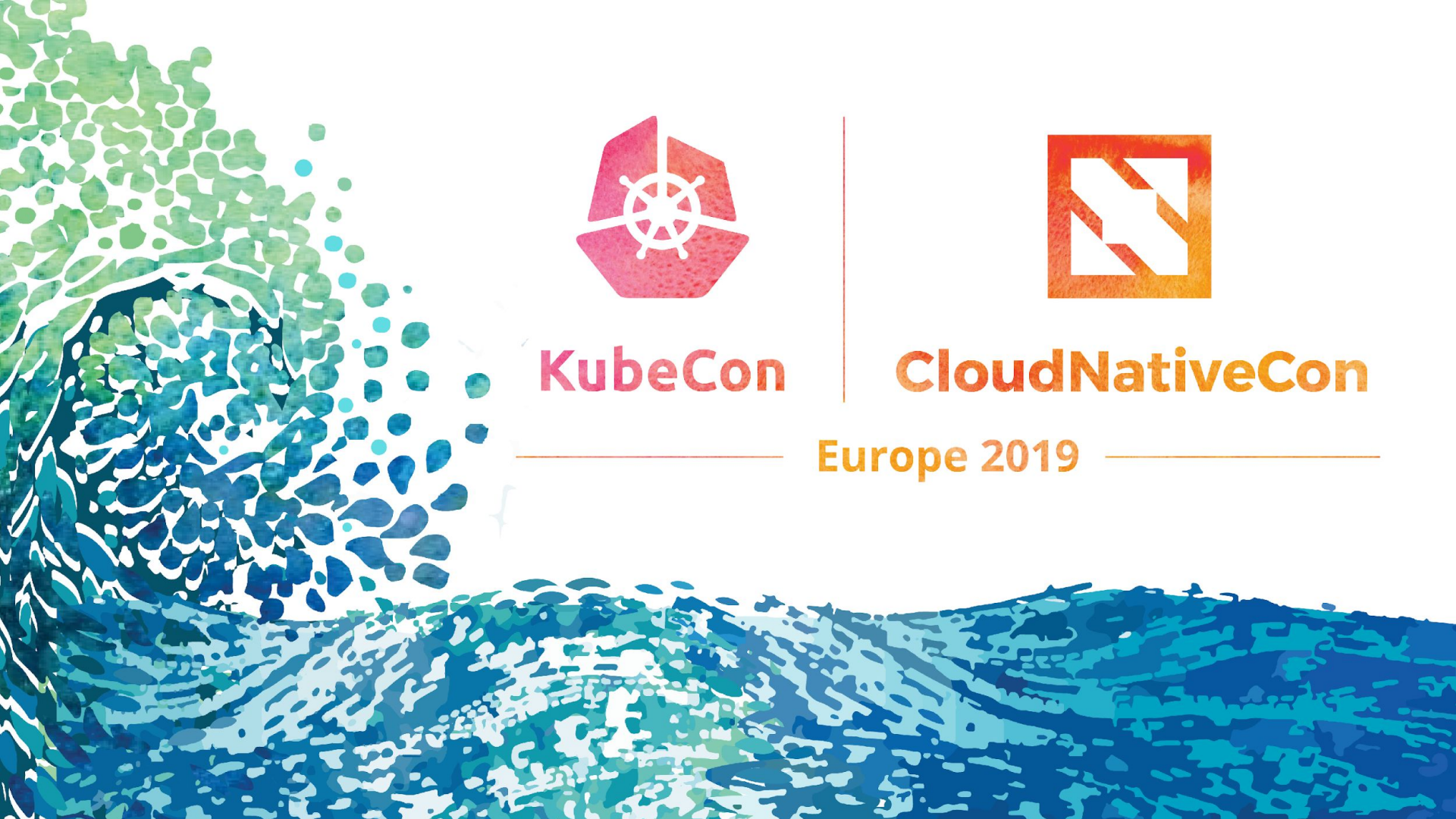
Support Matrix

VOLUME TYPE	FEATURES	CSI DRIVER VERSION
File mode, sharable or RWX Volume(CephFS)	Dynamically provision, de-provision volume	v0.3.0
	Creating and deleting snapshot	-
	Provision volumes from snapshot	-
	Provision volumes from another Volume	-
	Resize volumes	-
Block mode, sharable or RWX volumes(RBD) File/Block mode single-consumer or RWO volumes(RBD)	Dynamically provision, de-provision volume	v0.3.0,v1.0.0
	Creating and deleting snapshot	v0.3.0,v1.0.0
	Provision volumes from snapshot	v1.0.0
	Provision volumes from another Volume	-
	Resize volumes	-

Give a try!

- Download minikube

```
minikube start
git clone https://github.com/rook/rook
cd cluster/examples/kubernetes/ceph
kubectl create -f common.yaml operator.yaml
kubectl create -f cluster.yaml
```


KubeCon
CloudNativeCon
Europe 2019