Rook Project Intro

Jared Watts
Rook Maintainer
Upbound Founding Engineer

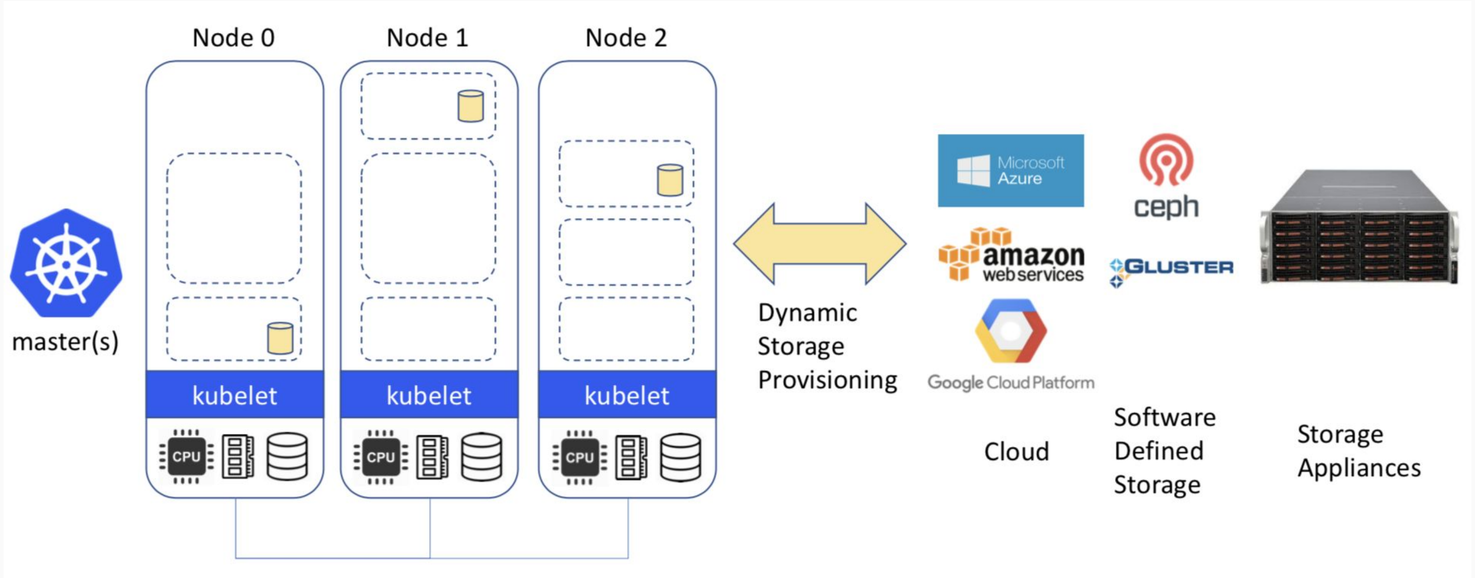
https://rook.io/
https://github.com/rook/rook

What is Rook?

- Cloud-Native Storage Orchestrator
- Extends Kubernetes with custom types and controllers
- Automates deployment, bootstrapping, configuration, provisioning, scaling, upgrading, migration, disaster recovery, monitoring, and resource management
- Framework for many storage providers and solutions
- Open Source (Apache 2.0)
- Hosted by the Cloud-Native Computing Foundation (CNCF)

Storage for Kubernetes

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

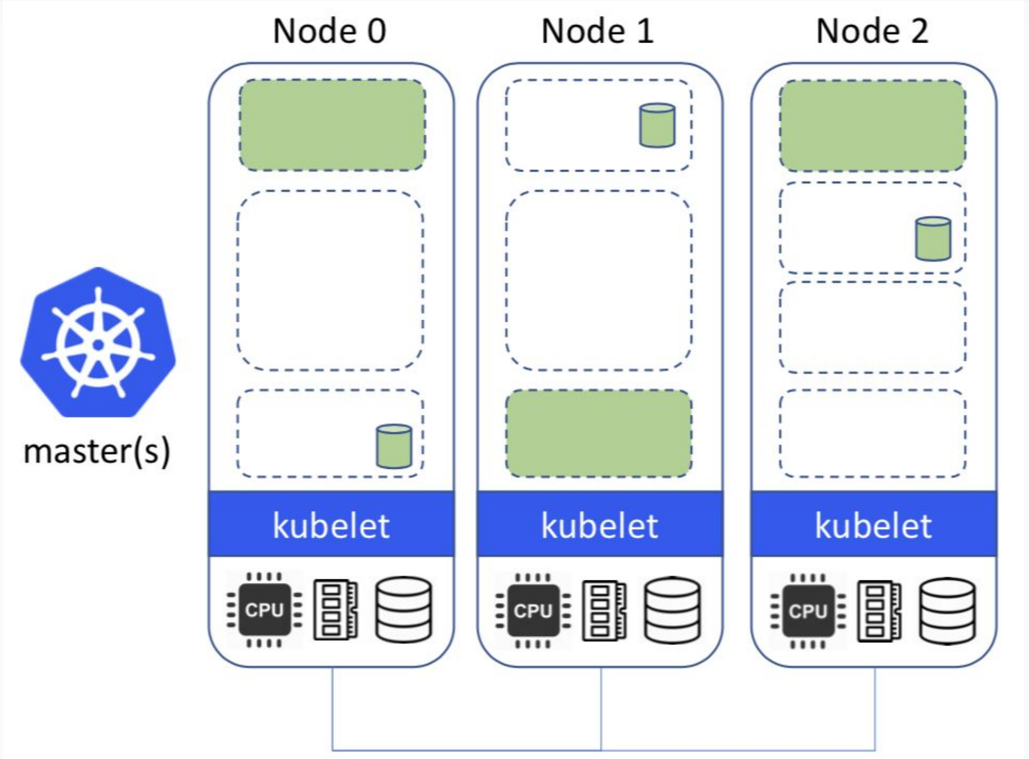


Limitations

- 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



Operator Pattern

- Codifies domain expertise to deploy and manage an application
  - Automates actions a human would normally do
- Control loop that reconciles user’s desired state and the actual system state
  - Observe - discover current actual state of cluster
  - Analyze - determine differences from desired state
  - Act - perform operations to drive actual towards desired

Custom Resource Definitions (CRDs)

- Teaches Kubernetes about new first-class objects
- Custom Resource Definition (CRDs) are arbitrary types that extend the Kubernetes API
  - look just like any other built-in object (e.g. Pod)
  - Enabled native `kubectl` experience
- A means for user to describe their desired state

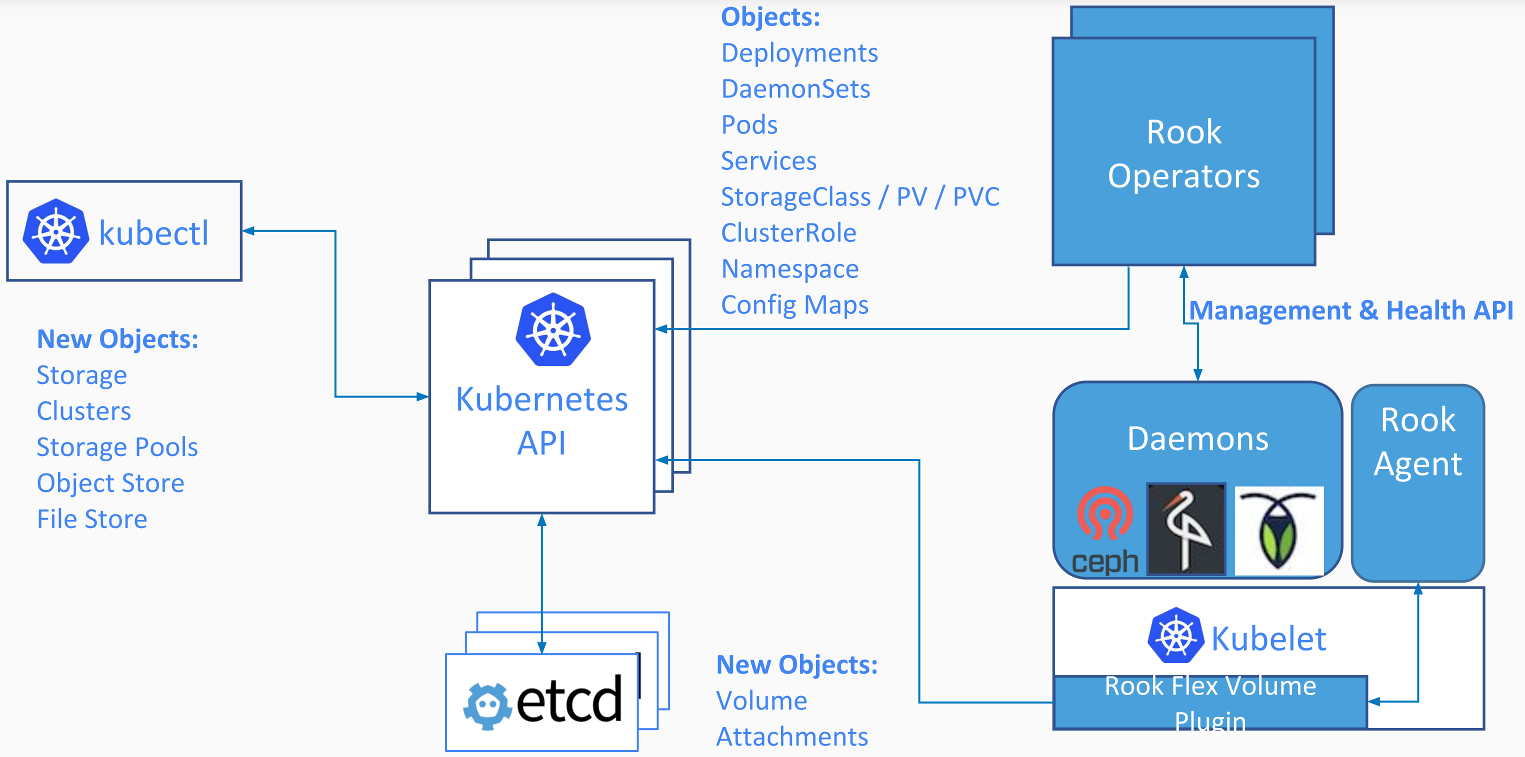
Rook Operators

- Implements the **Operator Pattern** for storage solutions
- Defines *desired state* for the storage cluster
  - Storage Cluster, Pool, Object Store, etc.
- The Operator runs reconciliation loops
  - Watches for changes in desired state
  - Watches for changes in the cluster
  - Applies changes to the cluster to make it match desired

Rook Operators

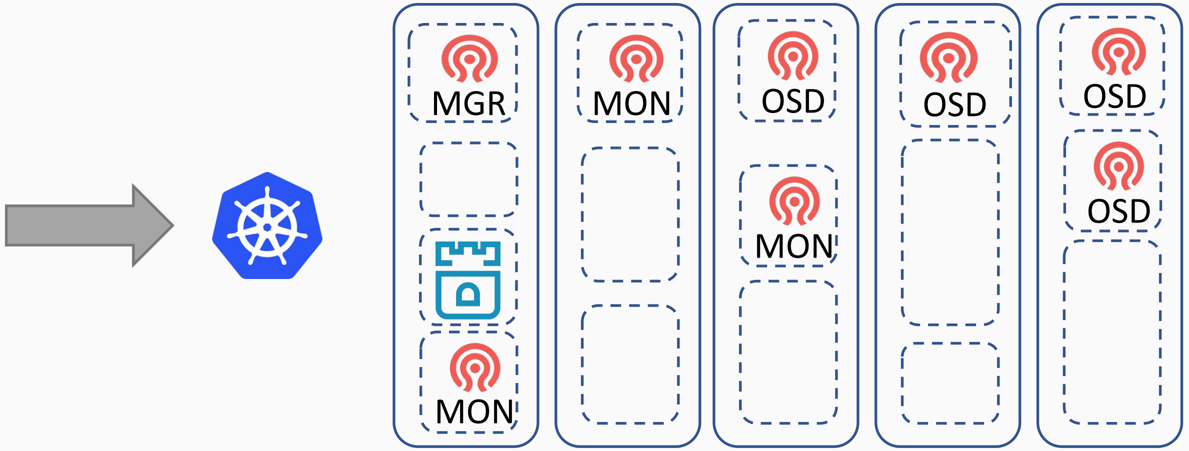
- The Operators leverages the full power of K8S
  - Services, ReplicaSets, DaemonSets, Secrets, …
- Contain all the logic to manage storage systems at scale
  - Handle stateful upgrades
  - Handle rebalancing the cluster
  - Handle health and monitoring tasks
- Not on the data path – can be offline for minutes

Rook Architecture



Ceph on Kubernetes with Rook

```
apiVersion: ceph.rook.io/v1beta1
kind: Cluster
metadata:
  name: rook-ceph
spec:
  cephVersion:
    image: ceph/ceph:v13
  mon:
    count: 3
  network:
    hostNetwork: false
  storage:
    useAllNodes: true
    deviceFilter: "^sd."
    config:
      storeType: bluestore
```



Rook Framework for Storage Solutions

- Rook is more than just a collection of Operators and CRDs
- **Framework** for storage providers to integrate their solutions into cloud-native environments
  - Storage resource normalization
  - Operator patterns/plumbing
  - Common policies, specs, logic
  - Testing effort
- Ceph, CockroachDB, Minio, NFS, Cassandra, Nexenta, and more...

Demo

Deploying a Ceph cluster with a Stateful Application

How to get involved?

- Contribute to Rook
  - https://github.com/rook/rook
  - https://rook.io/
- Slack - https://rook-io.slack.com/
  - #conferences now for Kubecon China
- Twitter - @rook_io
- Forums - https://groups.google.com/forum/#!forum/rook-dev
- Community Meetings

More Sessions

- **Meet the Rook Maintainers**
  - Chat with project leaders and ask questions
  - Starts in 30 minutes! Wed Nov 14th, 15:00 @ CNCF Booth
- **Rook Deep Dive**
  - Code & architecture specifics, storage provider integration details
  - Thurs Nov 15th, 14:20

Questions?

https://github.com/rook/rook

https://rook.io/

Thank you!

https://github.com/rook/rook

https://rook.io/