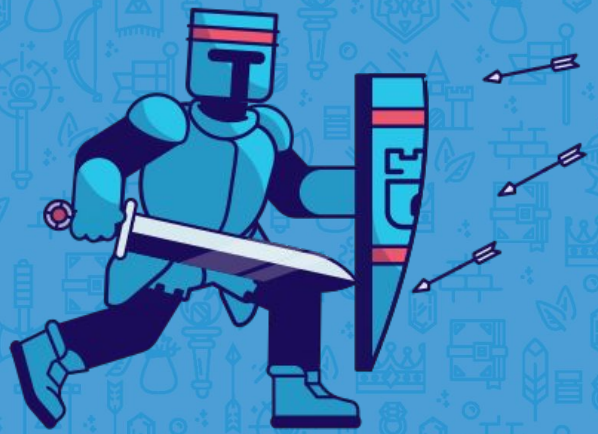
Rook Project Intro

Alexander Trost, Cloudibility
Travis Nielsen, Red Hat
Rook Maintainers

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



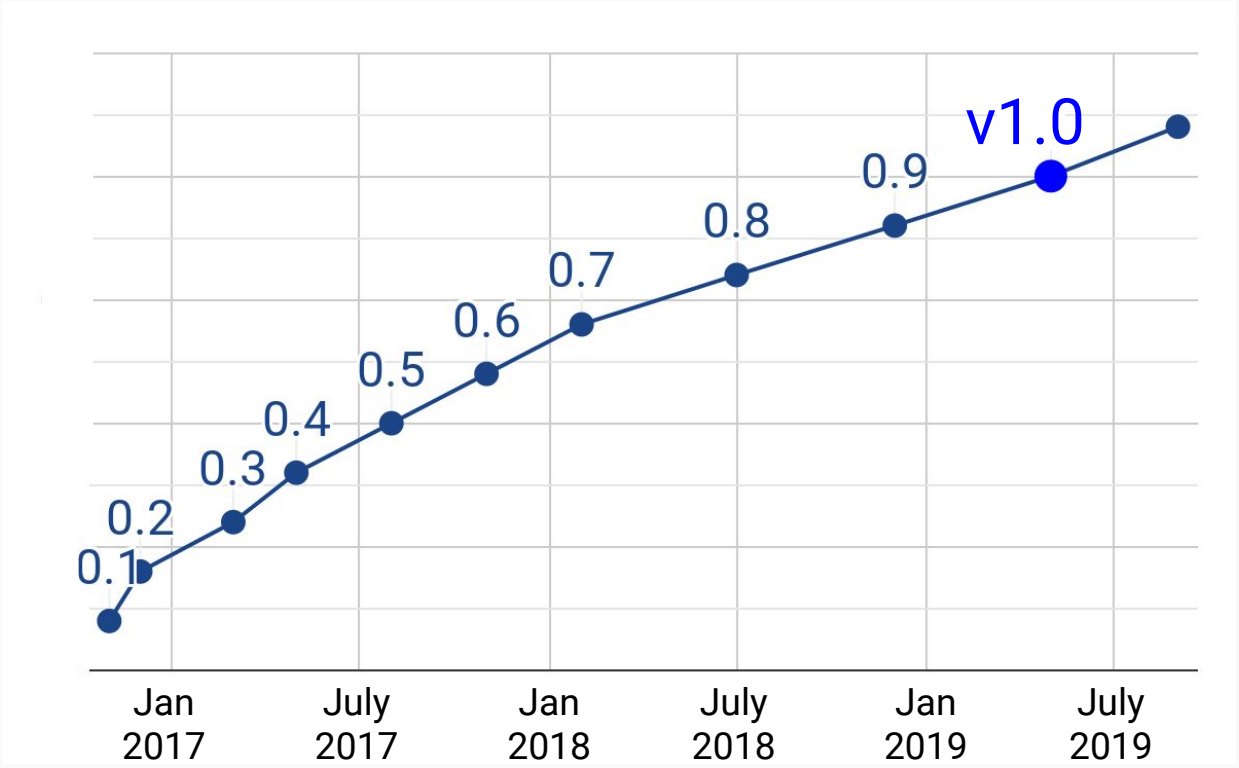
Agenda

- The Journey to v1.0
- Storage in Kubernetes
- Rook Design
- Demo
- How to Get Involved
  - More Rook sessions tomorrow!

Rook Stats

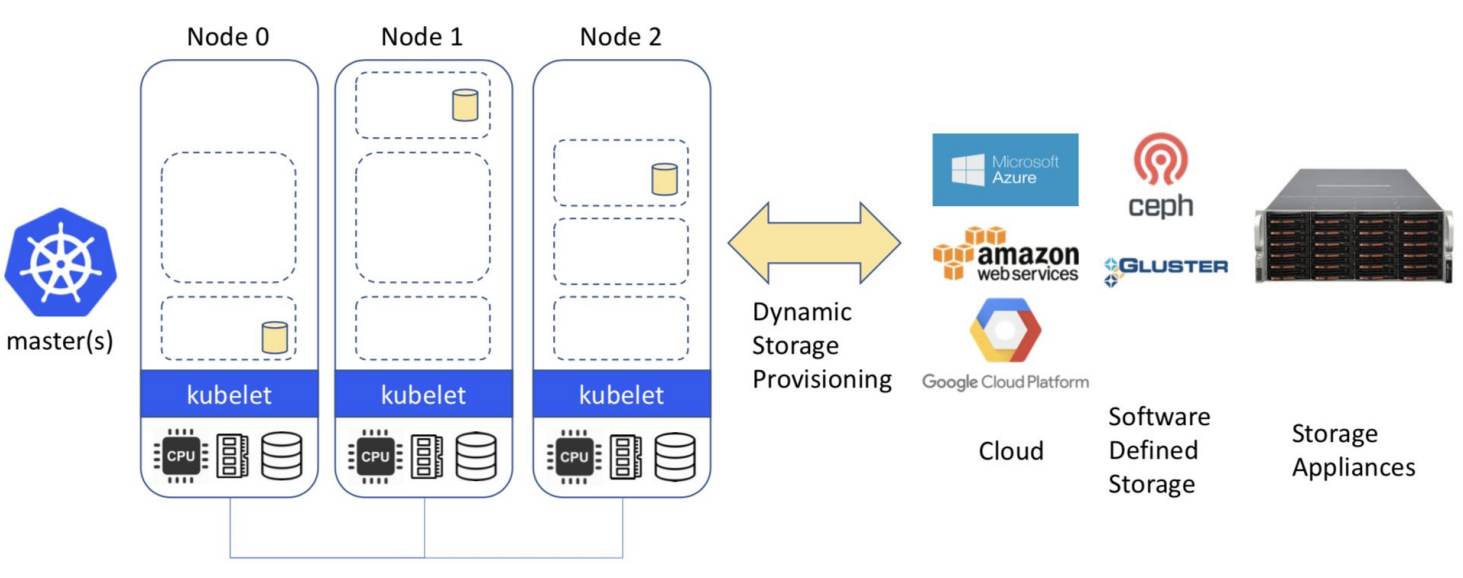
- v1.0 Release!
- 5.2K+ Github Stars
- 43M+ Downloads
- 150+ Contributors

Rook v1.0 Released!



Storage for Kubernetes

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

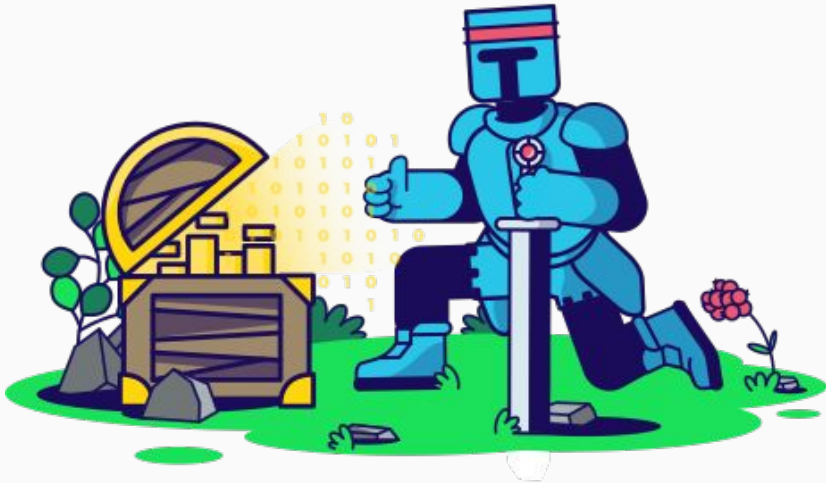


Storage Challenges

- Reliance on external storage
  - Not portable
  - Requires these services to be accessible
  - Deployment burden
- Reliance on cloud provider managed services
  - Vendor lock-in
- Day 2 operations - who is managing the storage?

What is Rook?

- Storage Operators for Kubernetes
- Automate
  - Deployment
  - Bootstrapping
  - Configuration
  - Upgrading
- Provision
  - Mount storage with PVCs

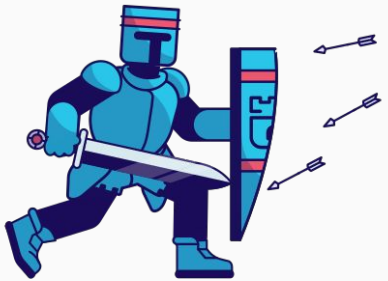


What is Rook?

- Open Source (Apache 2.0)
- Cloud-Native Computing Foundation (CNCF)
  - Incubation Project
- Extends Kubernetes with Operators and custom types
- Framework for many storage providers and solutions

Storage Providers

Ceph	Stable
EdgeFS	Beta
Cassandra	Alpha
CockroachDB	Alpha
Minio	Alpha
NFS	Alpha



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

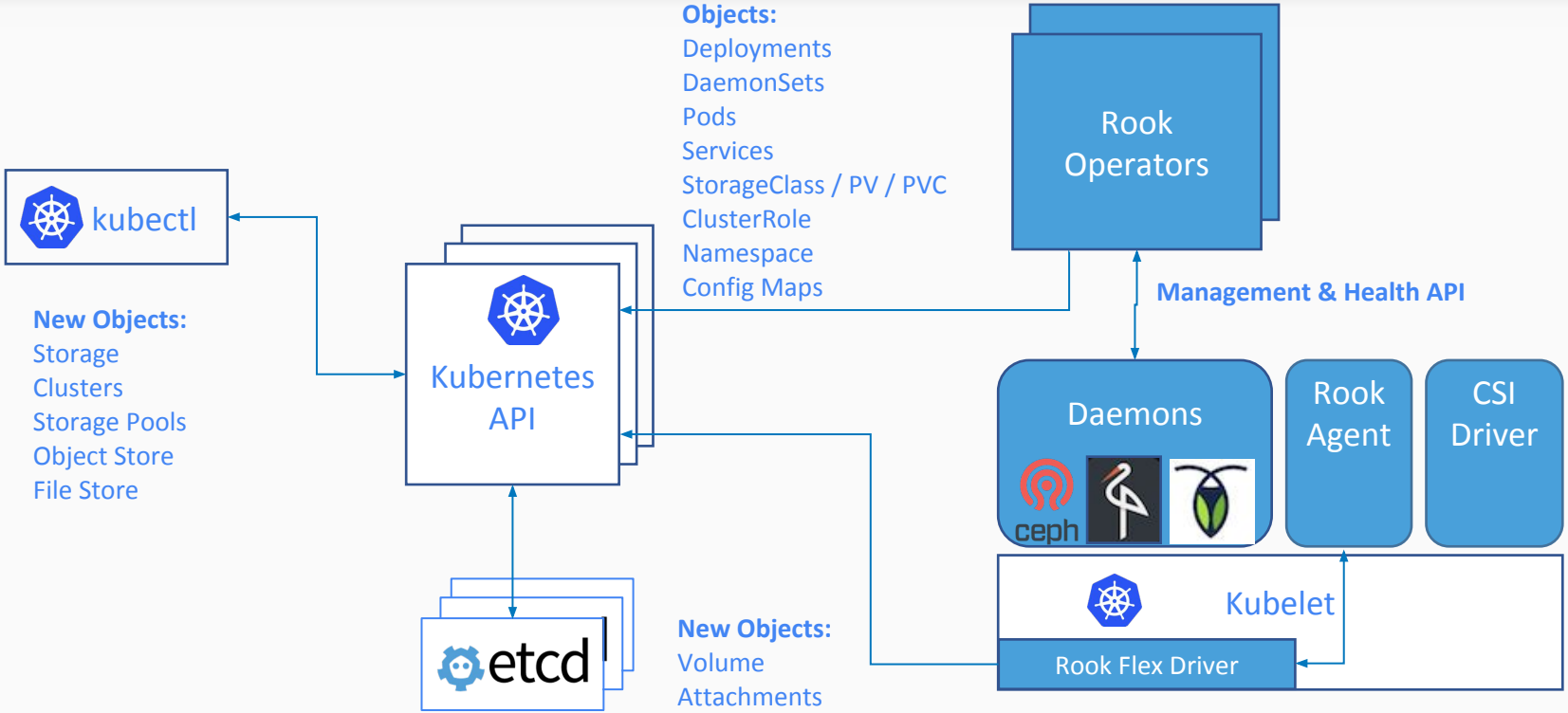
Rook Operators

- Implements the **Operator Pattern** for storage solutions
- Defines *desired state* for the storage resource
  - 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

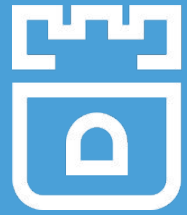
Rook Operators

- The Operators leverages the full power of K8S
  - Services, ReplicaSets, DaemonSets, Secrets, …
- Manage storage systems at scale
  - Stateful upgrades
  - Health and monitoring tasks
- Not on the data path – can be offline for minutes

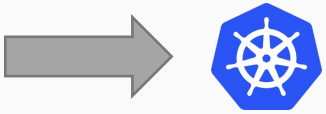
Rook Architecture

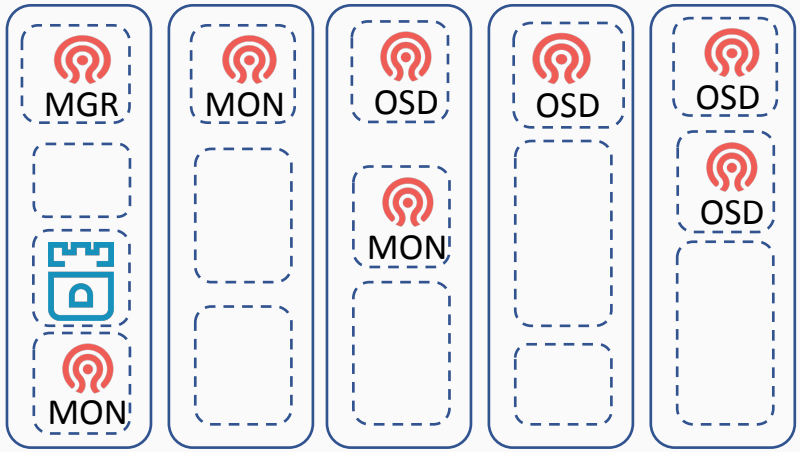


Ceph on Kubernetes with Rook



```yaml
apiVersion: ceph.rook.io/v1
kind: Cluster
metadata:
  name: rook-ceph
spec:
  cephVersion:
    image: ceph/ceph:v14
  mon:
    count: 3
  network:
    hostNetwork: false
  storage:
    useAllNodes: true
```





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, EdgeFS, ...

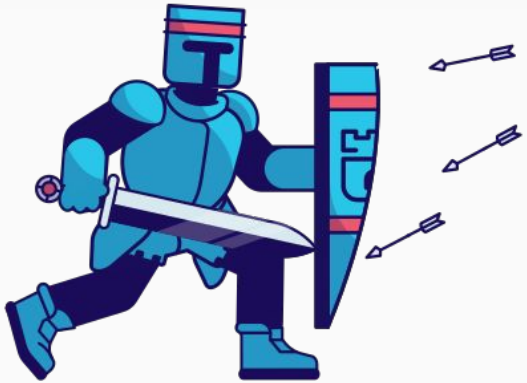
Demo

Deploying a Ceph cluster with a Stateful Application

Getting Started with Rook

Website	https://rook.io
Documentation	https://rook.io/docs/rook/v1.0/
Blog	https://blog.rook.io/
Install v1.0	https://github.com/rook/rook/releases/

How to get involved?



- Contribute to Rook, review issues and PRs
  - https://github.com/rook/rook
- Slack - https://rook-io.slack.com/
  - #conferences
- Twitter - @rook_io
- Community Meetings
- Forums: https://groups.google.com/forum/#!forum/rook-dev

Rook Sessions

- **Data Without Borders: Rook at a Global Scale**
  - Wednesday, **11:05** @ Hall 8.0 D2
- **Rook Deep Dive**
  - Wednesday, **11:55** @ Hall 8.1 G3
- **Meet the Maintainers**
  - Wednesday, **12:30** @ CNCF Answer Bar
- **Keep the Space Shuttle Flying: Writing Robust Operators**
  - Wednesday, **15:55** @ Hall 8.1 G2
- **Rook, Ceph, and ARM: A Caffeinated Tutorial**
  - Wednesday, **16:45** @ Hall 8.0 D2

Thank you!

https://github.com/rook/rook

https://rook.io/