


KubeCon | CloudNativeCon

North America 2023

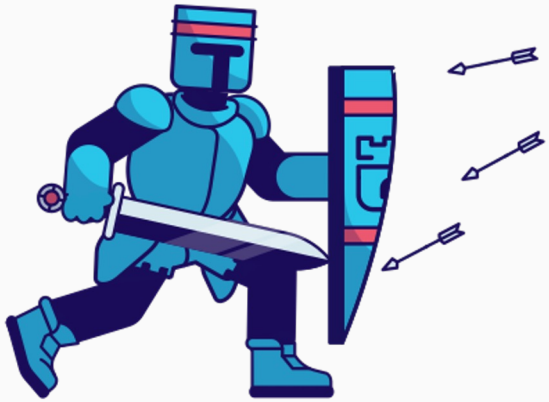
# Rook: Storage for Kubernetes

***Travis Nielsen***, *IBM Storage*
***Annette Clewett***, *IBM Storage*
***Dmitry Mishin***, *San Diego Supercomputer Center, UCSD*

*November 2023*

# Agenda

- Introduction to Rook and Ceph
- Use case: National Research Platform
- Application Disaster Recovery



# Raise your hand if…

- Are you here to learn about Rook for the first time?
- Have you heard of Ceph?
- Have you experimented with Rook?
- Have you deployed Rook in production?

# Introduction to Rook

# Questions that led to Rook

- Storage is commonly provided by cloud providers
- What about storage in your datacenter?
- Storage is traditionally not part of the cluster
  - Why should storage be external to K8s?
- Why not manage storage as any other K8s application?

# Storage Platform

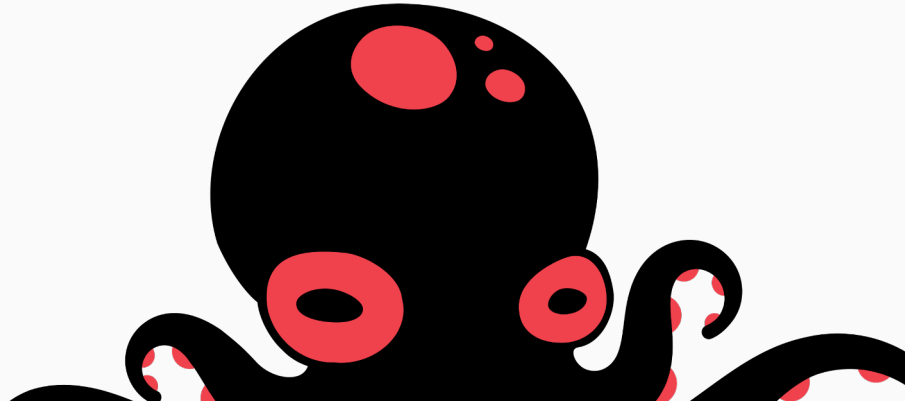
- Which storage platform to trust?
- Enterprises don’t trust a new data platform
- We didn’t want to build a new storage platform
- We made the decision to build on **Ceph**

# What is Rook?

- Makes storage available inside your Kubernetes cluster
- Manages Ceph storage with an operator and Custom Resource Definitions (CRDs)
- Automates deployment, configuration, upgrades
- Allows apps to consume storage like any other K8s storage
  - Storage Classes, Persistent Volume Claims
- Open Source (Apache 2.0)

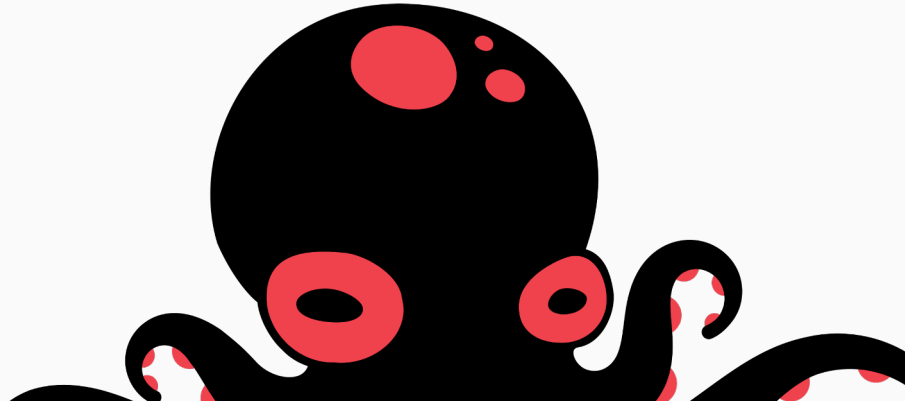
# What is Ceph?

- Distributed Software-Defined Storage solution
  - Block (RWO)
  - Shared File System (RWX)
  - Object (S3 Buckets)
- https://ceph.io/
- Open source



# Why Ceph?

- Block, Object, and File storage in a single platform
- Proven history of enterprise adoption and support
  - First release in July 2012
  - CERN’s Large Hadron Collider!



# Ceph data durability

- Ceph designed to be consistent, not eventually consistent
- Data sharded across partitions (AZs), racks, nodes, disks
- Shard replication is configurable
- Proven highly durable
- Even in extreme disasters, data can be recovered manually

# Architectural Layers

- Rook
  - Operator owns the **management** of Ceph
- CSI
  - Ceph CSI driver dynamically **provisions** and **mounts** storage to user application Pods
- Ceph
  - **Data** layer

# How do you install Rook?

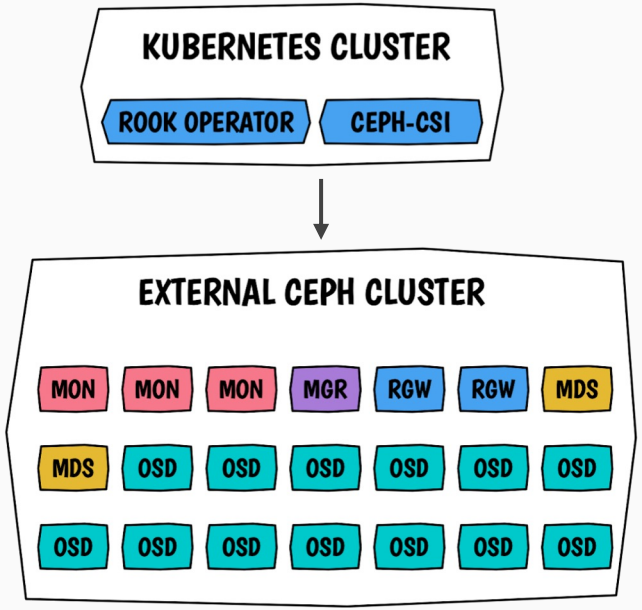
- Helm charts
- Example manifests for many configurations
- Quickstart guide
  - https://rook.io and click Get Started

# Rook installation environments

- Anywhere Kubernetes runs
  - Cloud or on-prem
  - Virtual or bare metal hardware
  - Underlying storage can be node-attached devices, cloud volumes, or loopback devices
- Rook helps enable cross-cloud support

# External Cluster Connection

- Connect to a Ceph cluster outside of the current K8s cluster
- Dynamically create Block/File/Object storage consumable by K8s applications



# Object Storage Provisioning

- Container Object Storage Interface (COSI)
  - In the latest release!
- Object Bucket Claim (OBC)
  - Similar pattern to a Persistent Volume Claim (PVC)
  - The operator creates a bucket
  - Give access via a K8s Secret

# Rook Project

# Rook Community

- Community first
- Open Source (Apache 2.0)
- Maintainers from four companies
  - Cybozu, IBM/Red Hat, Koor, Upbound
- 400+ contributors to the Github project
- 300M container downloads

# CNCF Graduated

- Rook is a CNCF graduated project!
  - Sandbox: January 2018
  - Incubation: September 2018
  - Graduation: October 2020

# Rook Stability

- Three years since CNCF graduation
- Five years since declared stable for production
- Seven years since project announced
- Many upstream users running in production
- Many downstream deployments running in production

# Release Cycle

- Minor releases are about **every 4 months**
  - v1.12 was in July
  - v1.13 planned in early December
- Regular patch releases
  - Biweekly unless there is a critical need

# Real world findings: National Research Platform


NATIONAL RESEARCH PLATFORM
Designed for Growth & Inclusion
NRP is a partnership of more than 50 institutions, led by researchers and cyberinfrastructure professionals at UC San Diego, supported in part by awards from National Science Foundation

# What is National Research platform (NRP)

NRP NATIONAL RESEARCH PLATFORM

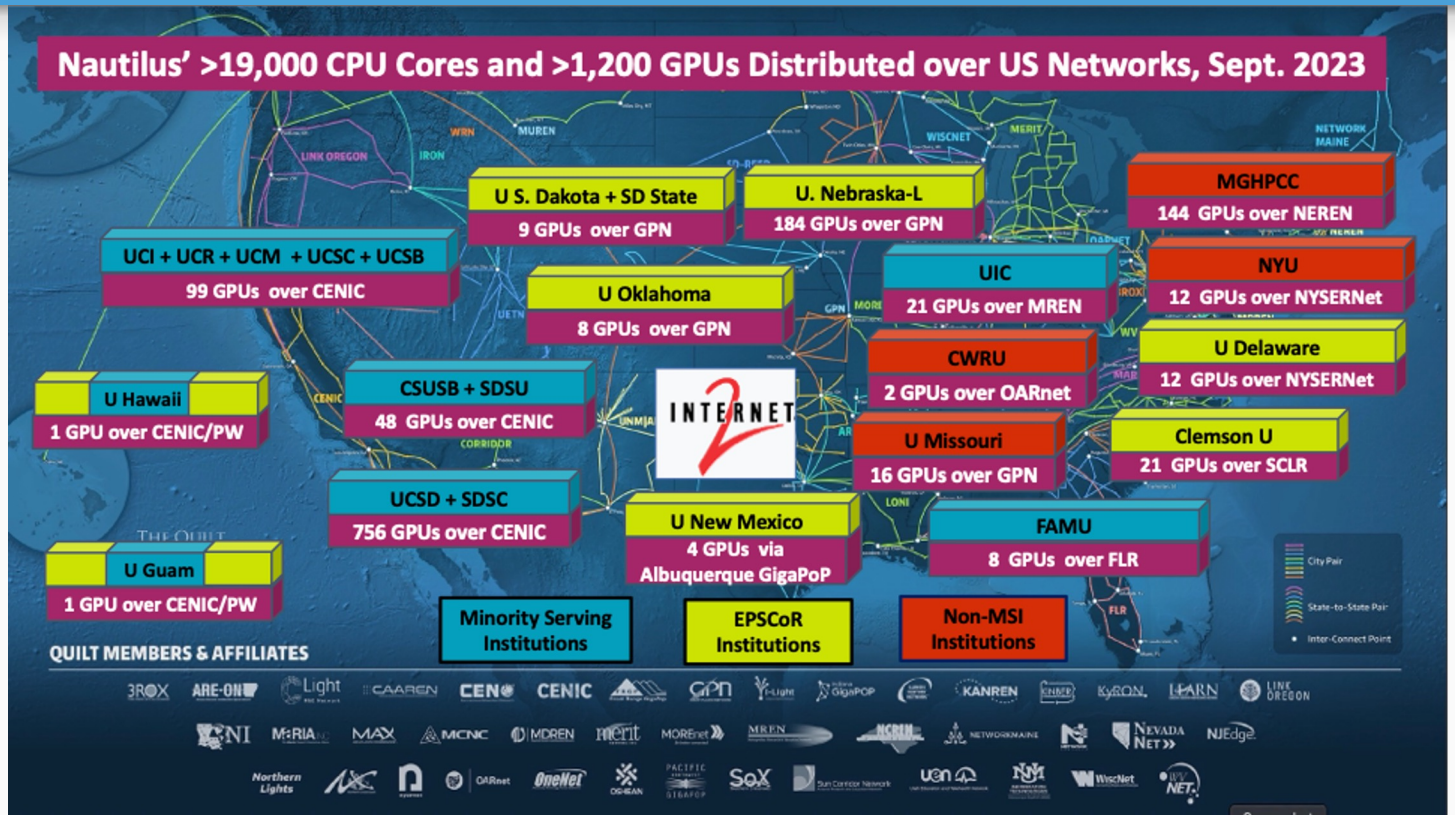
Nautilus

- Global kubernetes cluster (Nautilus)
- 10-100Gbit ScienceDMZ connections with jumbo frames
- Includes 6 local and regional ceph clusters
- Can mount from any cluster node across the world

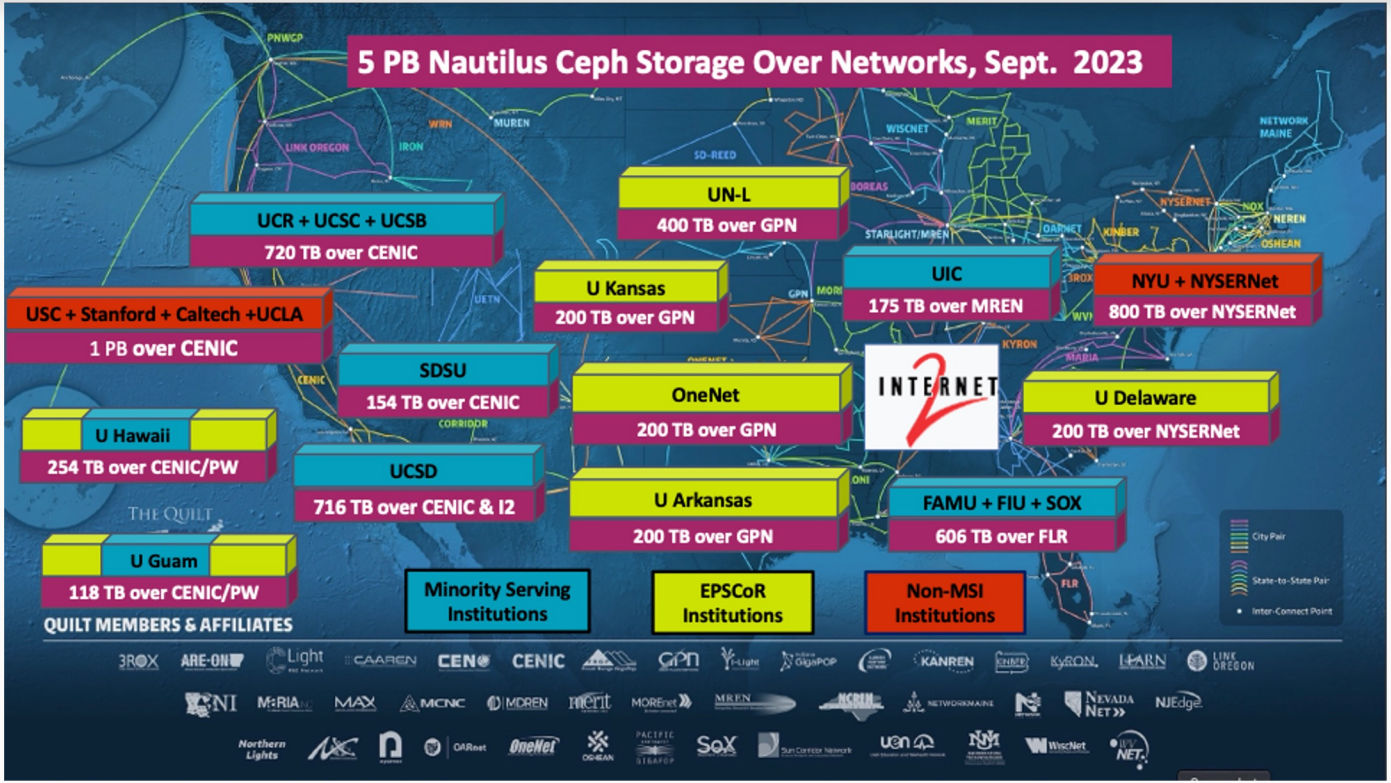
# Bring Your Own resources

- Researchers and providers can **add their own** resources
- Researchers provide hardware maintenance, power, cooling, networking
- **NRP** admins support the **OS and above**
- 5 minutes and a single ansible command from bare Ubuntu OS to Nautilus node
- Storage drives management and volumes creation in kubernetes

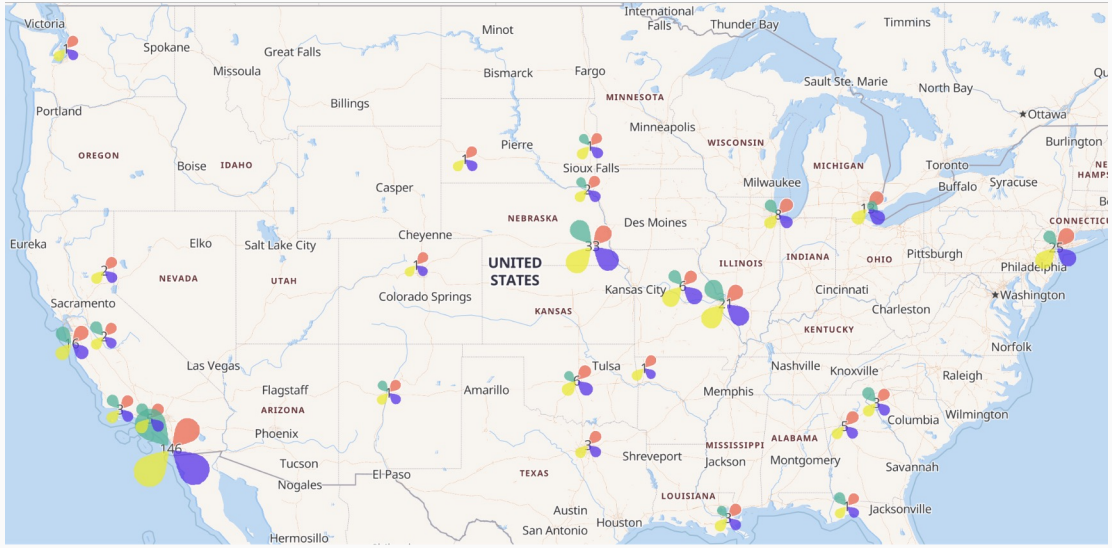
# GPU distribution

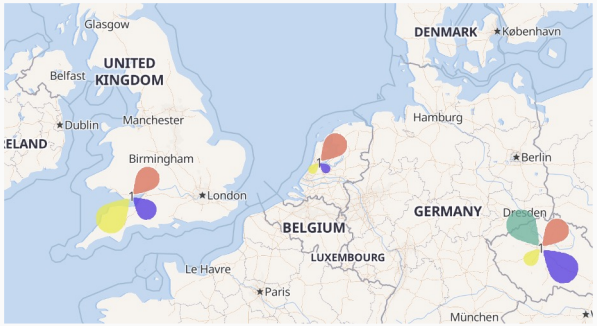


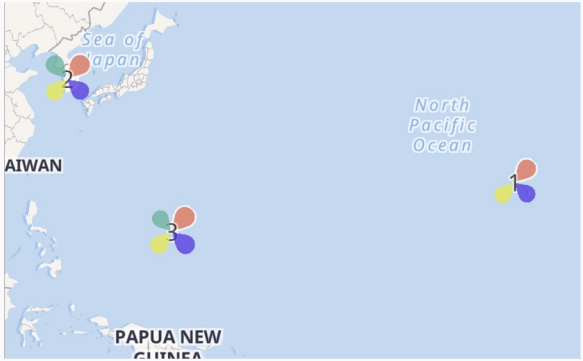
# Ceph by Rook storage distribution

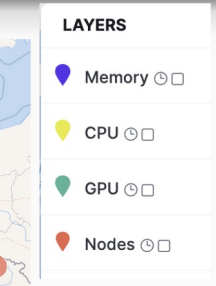


# BYOR status

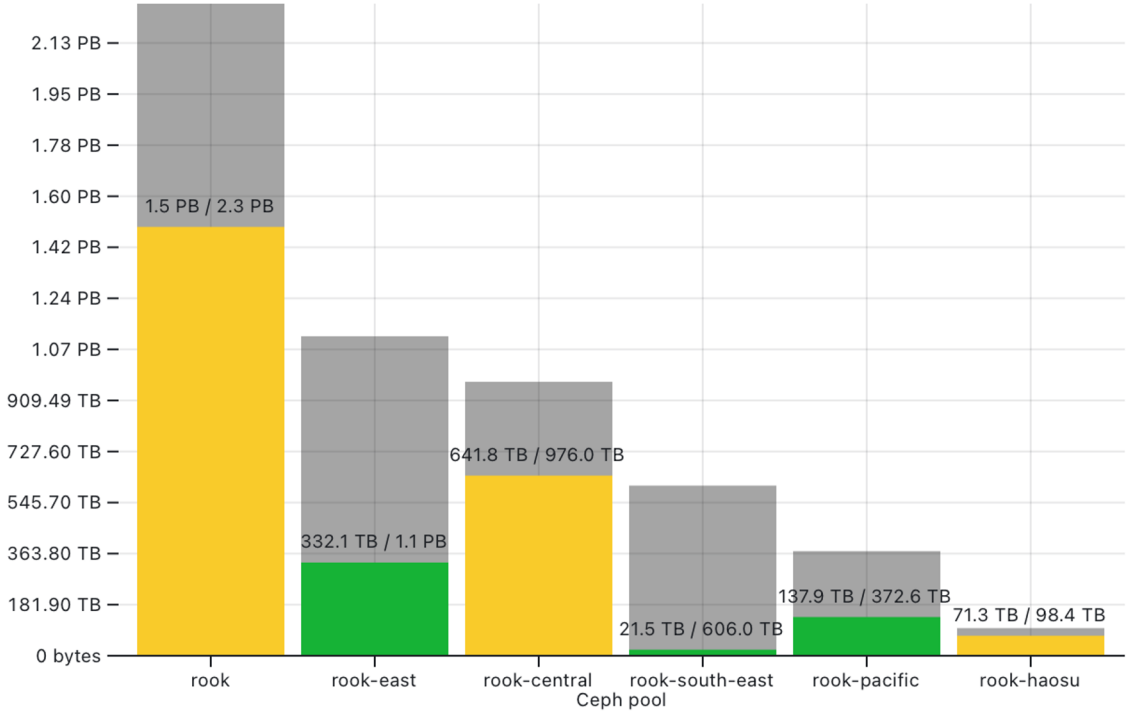

Victoria
Spokane
Great Falls
Missoula
Minot
International Falls
Thunder Bay
Timmins
Bismarck
Fargo
Sault Ste. Marie
North Bay
Portland
Billings
MINNESOTA
Minneapolis
Ottawa
Pierre
WISCONSIN
Burlington
OREGON
Boise
IDAHO
Sioux Falls
MICHIGAN
Toronto
Buffalo
Syracuse
Casper
Milwaukee
NEBRASKA
Des Moines
Cheyenne
Eureka
Elko
Salt Lake City
UNITED STATES
ILLINOIS
INDIANA
OHIO
Pittsburgh
Philadelphia
NEVADA
UTAH
Kansas City
Cincinnati
Colorado Springs
Washington
Sacramento
KANSAS
Charleston
KENTUCKY
Norfolk
Las Vegas
Tulsa
Nashville
Knoxville
Raleigh
Flagstaff
Amarillo
Memphis
ARIZONA
Columbia
Wilmington
Phoenix
Shreveport
MISSISSIPPI
ALABAMA
Tucson
Jackson
Montgomery
Savannah
Nogales
El Paso
TEXAS
LOUISIANA
Austin
Houston
Jacksonville
San Antonio
Hermosillo


Glasgow
DENMARK
København
UNITED KINGDOM
Belfast
Dublin
Manchester
Hamburg
Birmingham
Berlin
London
GERMANY
Dresden
BELGIUM
LUXEMBOURG
Le Havre
Paris
München


Sea of Japan
North Pacific Ocean
AIWAN
PAPUA NEW


LAYERS
Memory
CPU
GPU
Nodes

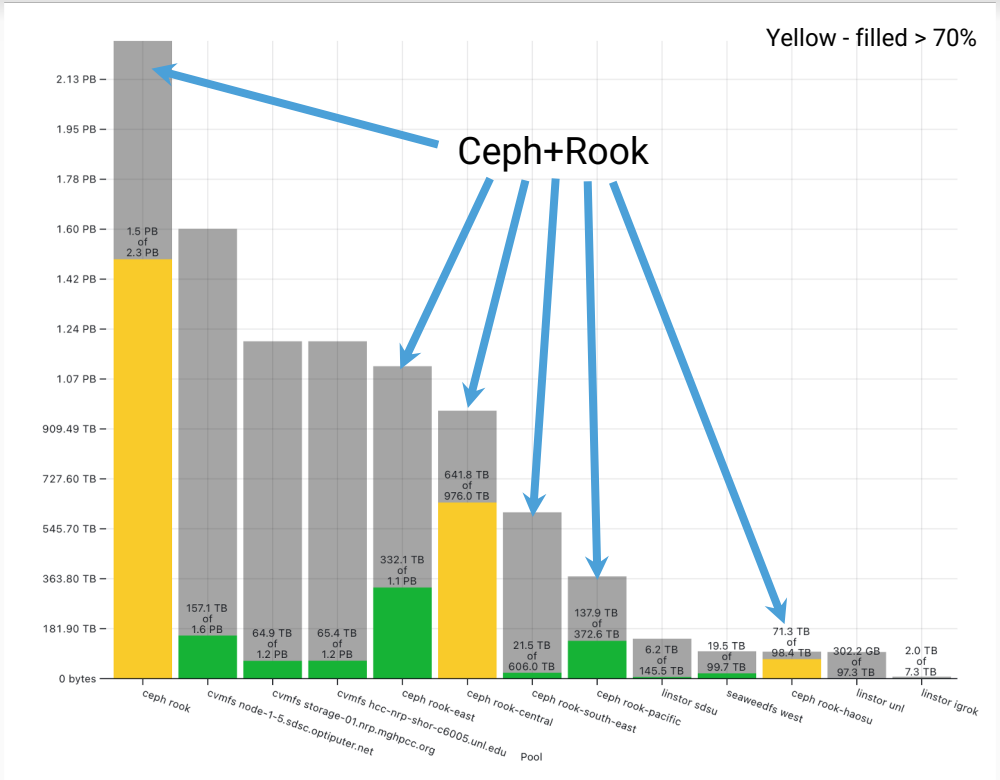
# Ceph clusters on Nautilus




Yellow - filled > 70%

# Monitoring with prometheus operator

# Other storages in the cluster

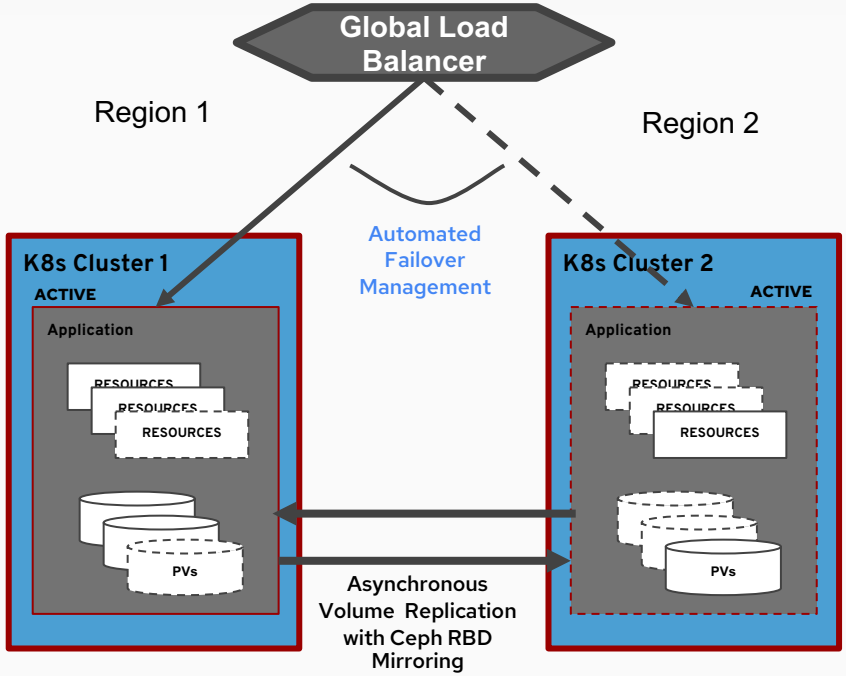



- Most users in the cluster are using **ceph/rook**
- Provides the most universal storage (block, shared FS, object)
- Most capacity managed (>5PB)

- Other storages used:
  - Linstor - block storage via DRBD
  - SeaweedFS - object storage
  - CVMFS+Xrootd - Read-only object storage (OSG origins + stashcaches)

# Application Disaster Recovery and Resiliency

# DR: Protection against Geographic Scale Disasters



# Disaster Recovery goals & objectives

Disaster recovery is the ability to recover and continue business critical applications from natural or human created disasters. Recovery goals are usually expressed as Recovery Point Objective (RPO) and Recovery Time Objective (RTO).

- RPO is a measure of how frequently you take backups or snapshots of persistent data. In practice, the RPO indicates the amount of data that will be lost or need to be re-entered after an outage. For synchronous storage replication the amount of data loss will be zero. For asynchronous storage replication the amount of data loss is related to the replication interval (e.g., 5 minutes of maximum data loss).
- RTO is the amount of downtime an application or service can tolerate. The RTO answers the question, “How long can it take for our system to recover after we were notified of a business disruption?”.

# Rook Application Disaster Recovery

RBD mirroring is an asynchronous replication of RBD images between peer Ceph clusters.

- RBD Mirroring CRD
  - CephRBDMirror
    - Creates rbd-mirror daemon
- Volume Replication CRDs
  - VolumeReplicaton
  - VolumeReplicationClass
    - Provides extended APIs for storage disaster recovery

# Rook Application Disaster Recovery

To achieve RBD Mirroring, csi-omap-generator and csi-addons containers need to be deployed in the RBD provisioner pods, which are not enabled by default.

- **Volume Replication Operator**: Volume Replication Operator is a kubernetes operator that provides common and reusable APIs for storage disaster recovery. The volume replication operation is supported by the CSIAddons.
- **Omap Generator**: Omap generator is a sidecar container that when deployed with the CSI provisioner pod, generates the internal CSI omaps between the PV and the RBD image.
- Edit the rook-ceph-operator-config configmap to enable these containers.

# Asynchronous DR Failover and Failback

Rook comes with the volume replication support, which allows users to perform disaster recovery and planned migration of applications.

- Application Failover (disaster recovery)
  - Communication with Primary site cluster where application is deployed is not required.
  - The Volume Replication operator automatically sends request to forcefully mark the RBD image as primary on the Secondary site cluster.
- Application Failback (planned migration)
  - Requires both Primary and Secondary sites are online and healthy.
  - Application scaled down so RPO=0 after recovery on alternate cluster.

# Open Cluster Management

Open Cluster Management is a community-driven project focused on multi-cluster and multicloud scenarios for Kubernetes apps.

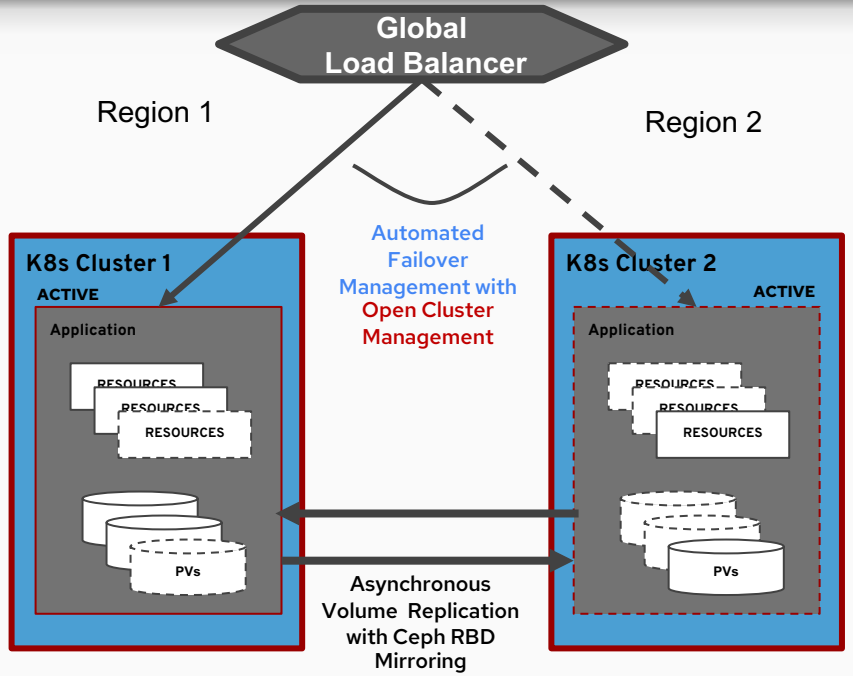
- Provides:
  - Cluster registry, Work distribution, Content placement, Vendor neutral APIs
- Leveraged for:
  - Cluster configuration
  - Application lifecycle management
  - Application placement using OCM Placement CRDs, which determine cluster(s) to deploy the application to
- Upstream Project:
  - https://github.com/open-cluster-management-io

# RamenDR

Kubernetes orchestrator that provides "Instant Cloud-Native Workload Recovery and Relocation Across Kubernetes Clusters".

- Orchestrates workload placement and PVC replication across k8s clusters
  - Enhances OCM Placement scheduling for DR workflows
  - Groups PVCs in an application and orchestrates their replication, leveraging VolumeReplication and VolumeReplicationClass
- Ramen CRDs:
  - DRPolicy, DRClusters, DRPlacementControl are used on the hub cluster VolumeReplicationGroup is on managed cluster and used for DR actions
- Upstream Project:
  - https://github.com/RamenDR/ramen

# Regional-DR (RDR) Architecture



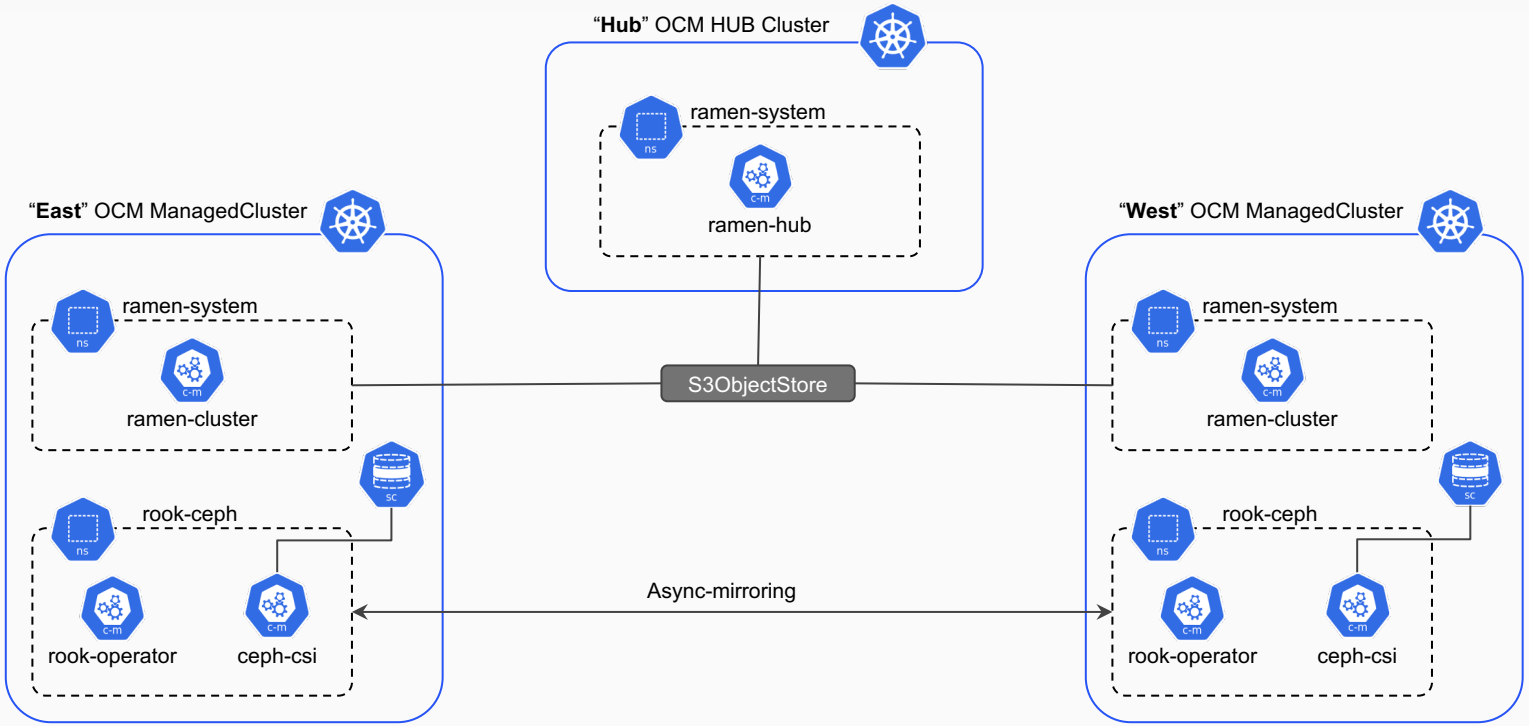

- Asynchronous Volume Replication => low RPO
  - Cross cluster replication of block volumes with replication intervals as low as 1 min defined in VolumeReplicationClass(s) and used in VolumeReplication CRs..
  - Ramen DR operators synchronizes both volume persistent data and kubernetes metadata for PVs
  - ***No distance limitations between peer clusters***
- Open Cluster Management (OCM) Failover Management => low RTO
  - OCM and Ramen DR operators enables failover and failback automation at application granularity
  - Both clusters remain active with Apps distributed and protected by the alternate cluster

Protection against Geographic Scale Disasters

# RamenDR drenv testing tool

- Required resources
  - Machine with 8 CPU and 32 GB mem
  - Linux - tested on Fedora 37, 38
- Required packages and tool
  - @virtualization group (nested virtualization enabled), minikube, kubectl, podman, cluteradm, and other tools (see the docs)
- We can create the env with one command:
  - **drenv start envs/regional-dr.yaml**
  - Creates minikube hub and 2 managed clusters for testing Regional DR using rbd mirroring.
- For more info see:
  - https://github.com/RamenDR/ramen/blob/main/docs/user-quick-start.md
  - https://www.youtube.com/embed/8-fpChSWzeo

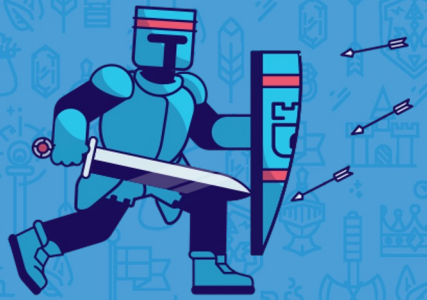
# Ramen, OCM & Rook Setup



# Questions?

## Come talk to us at the Rook booth in the Project Pavilion!

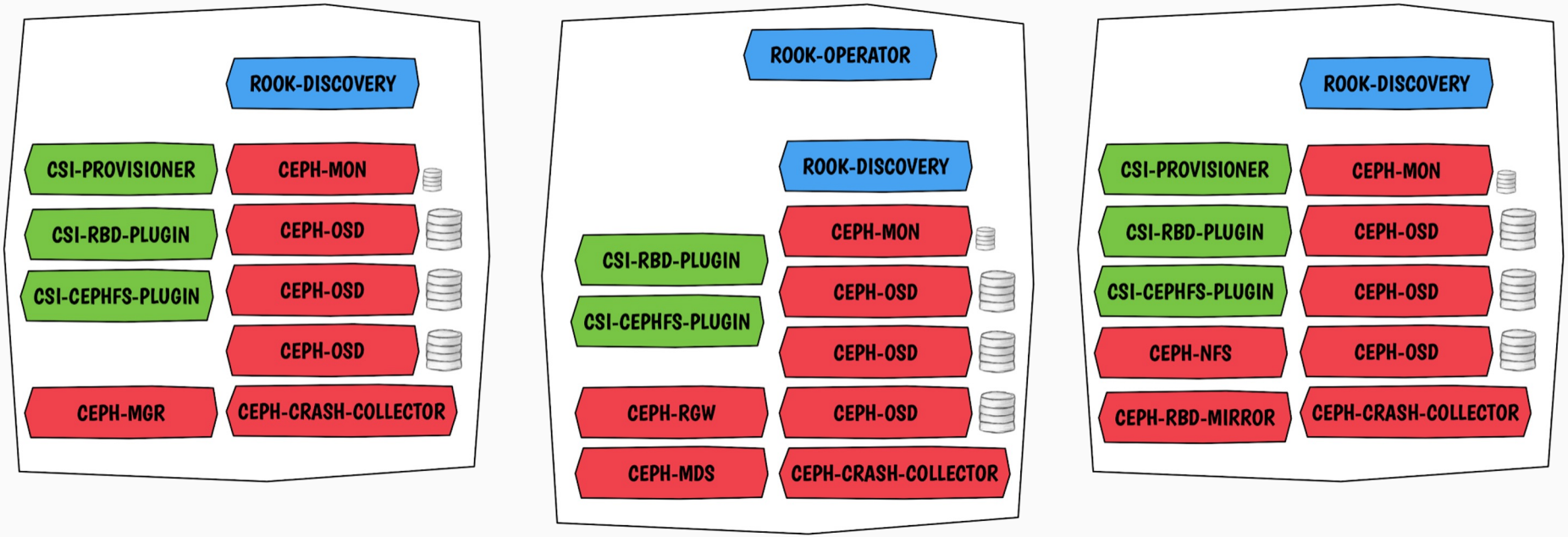
| | |
|---|---|
| Website and Documentation | https://rook.io/ |
| Slack | https://slack.rook.io/ |
| Twitter | @rook_io |



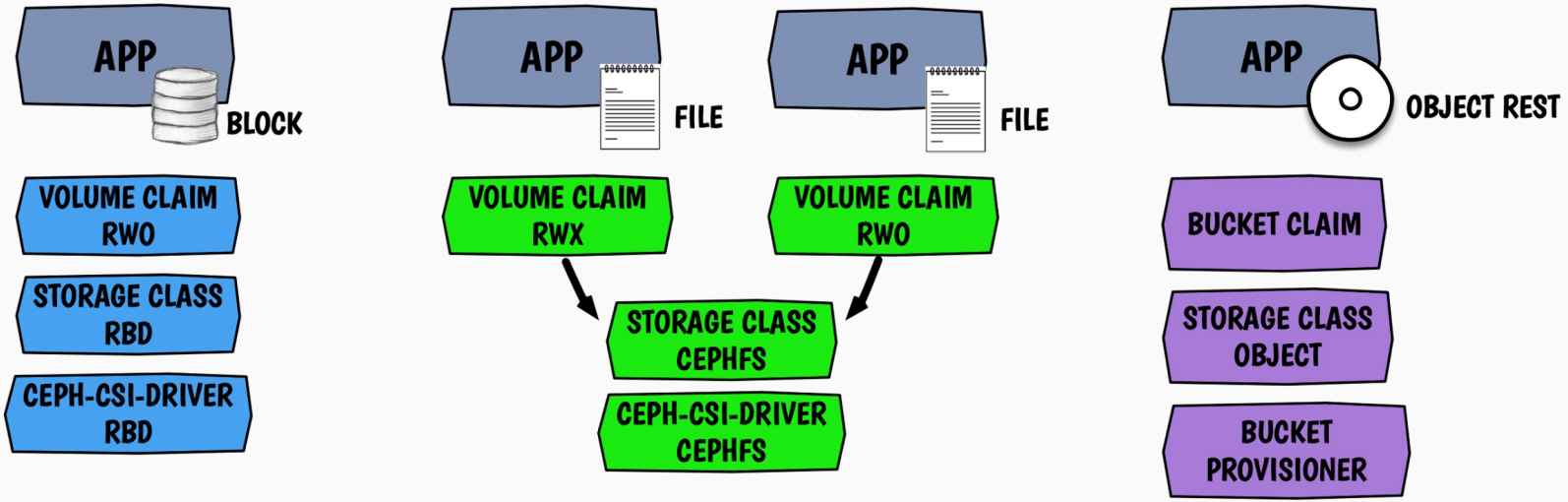

KubeCon
CloudNativeCon
North America 2023

# Appendix

# Rook Pods


ROOK-DISCOVERY
CSI-PROVISIONER
CEPH-MON
CSI-RBD-PLUGIN
CEPH-OSD
CSI-CEPHFS-PLUGIN
CEPH-OSD
CEPH-OSD
CEPH-MGR
CEPH-CRASH-COLLECTOR
ROOK-OPERATOR
ROOK-DISCOVERY
CSI-RBD-PLUGIN
CEPH-MON
CSI-CEPHFS-PLUGIN
CEPH-OSD
CEPH-OSD
CEPH-RGW
CEPH-OSD
CEPH-MDS
CEPH-CRASH-COLLECTOR
ROOK-DISCOVERY
CSI-PROVISIONER
CEPH-MON
CSI-RBD-PLUGIN
CEPH-OSD
CSI-CEPHFS-PLUGIN
CEPH-OSD
CEPH-NFS
CEPH-OSD
CEPH-RBD-MIRROR
CEPH-CRASH-COLLECTOR

# Layer 2: CSI Provisioning



# Layer 3: Ceph Data Path