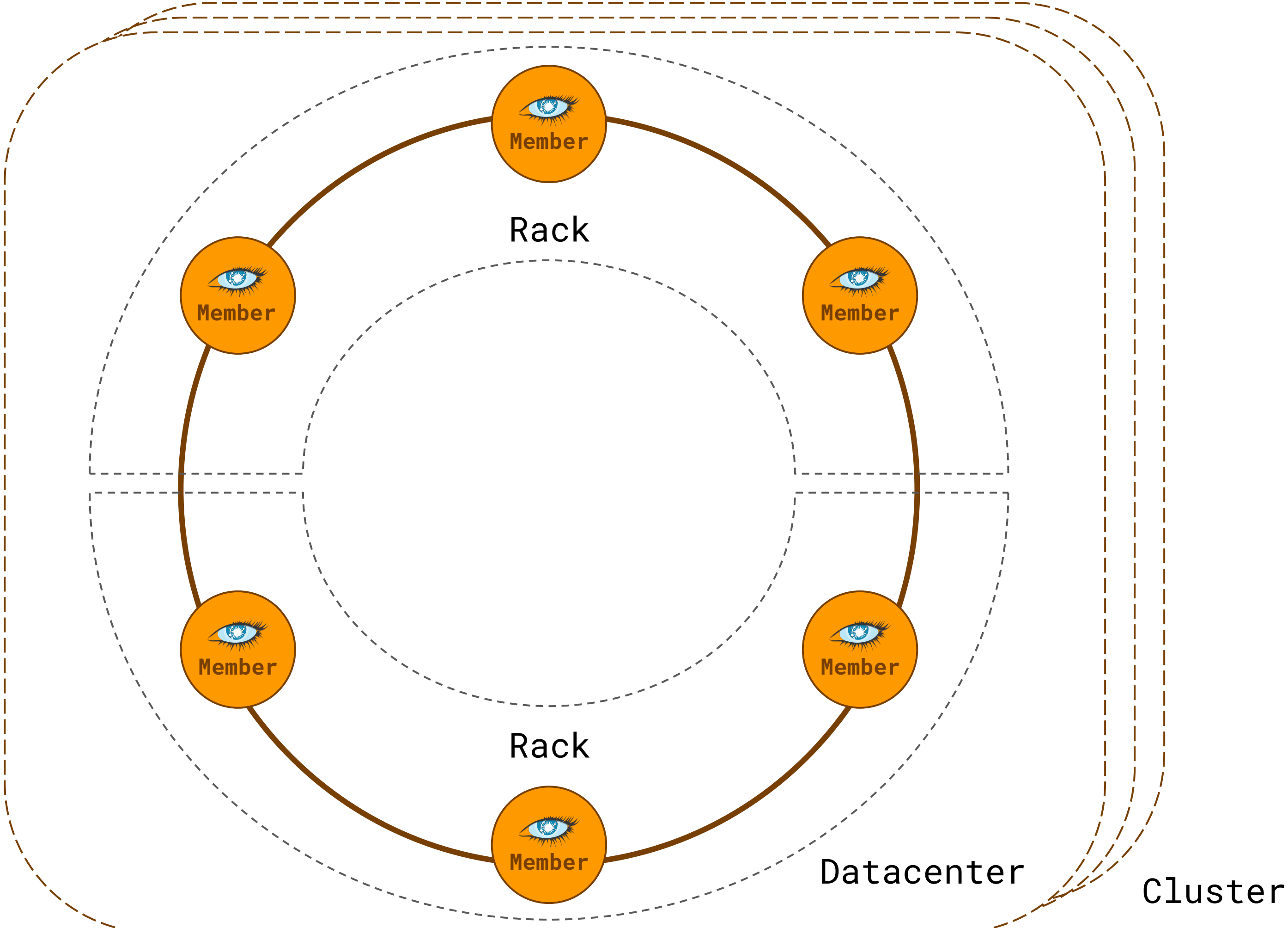
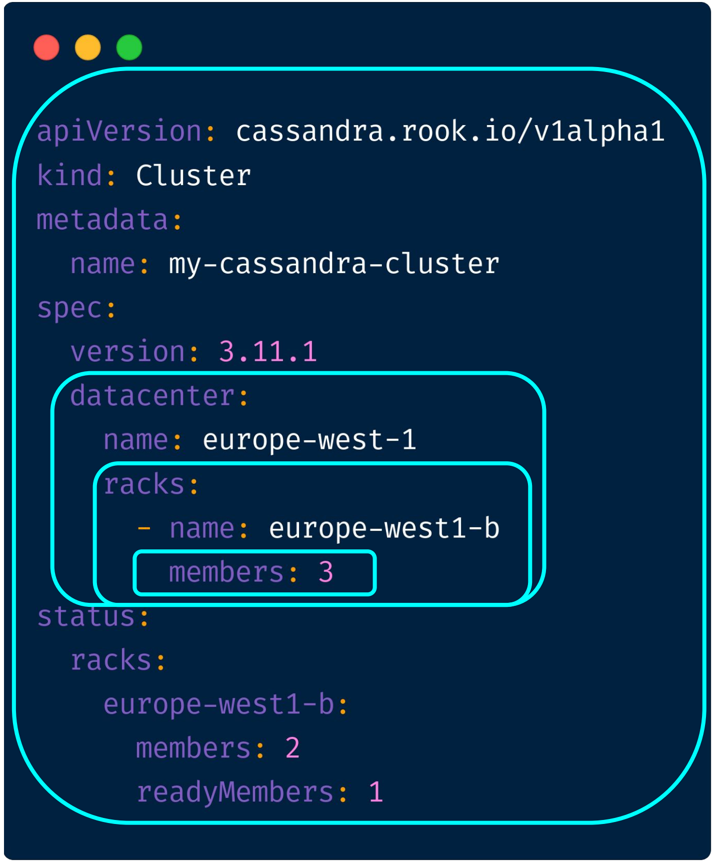
Rook Cassandra Operator

Cassandra Overview



Mapping of Abstractions

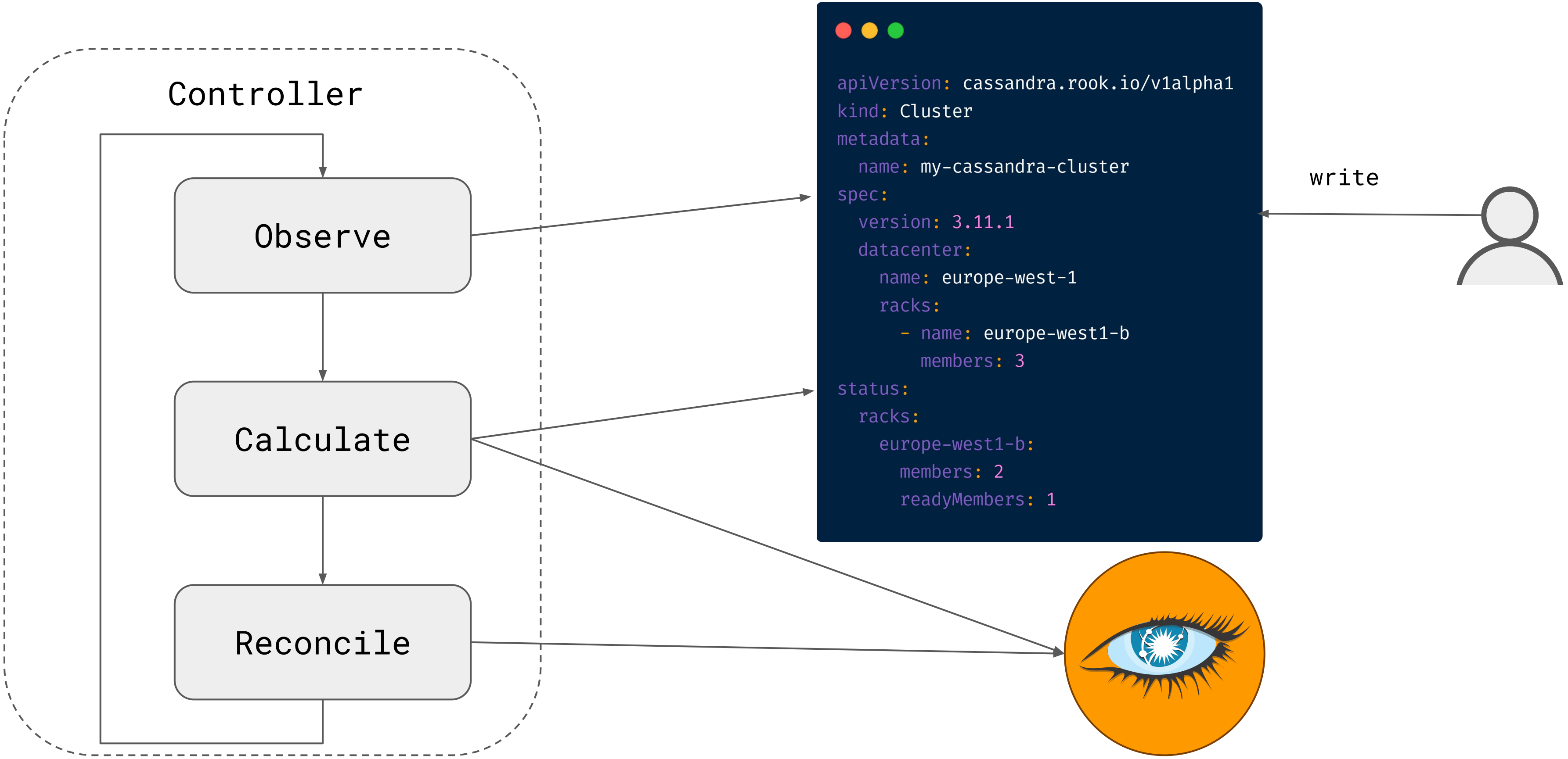
cassandra	kubernetes
Member	Pod
Rack	StatefulSet
Datacenter	StatefulSets
Cluster	Cluster Custom Resource



```
apiVersion: cassandra.rook.io/v1alpha1
kind: Cluster
metadata:
  name: my-cassandra-cluster
spec:
  version: 3.11.1
  datacenter:
    name: europe-west-1
    racks:
      - name: europe-west1-b
        members: 3
status:
  racks:
    europe-west1-b:
      members: 2
      readyMembers: 1
```

The Operator Pattern

Operator = Controller(s) + CRD(s)



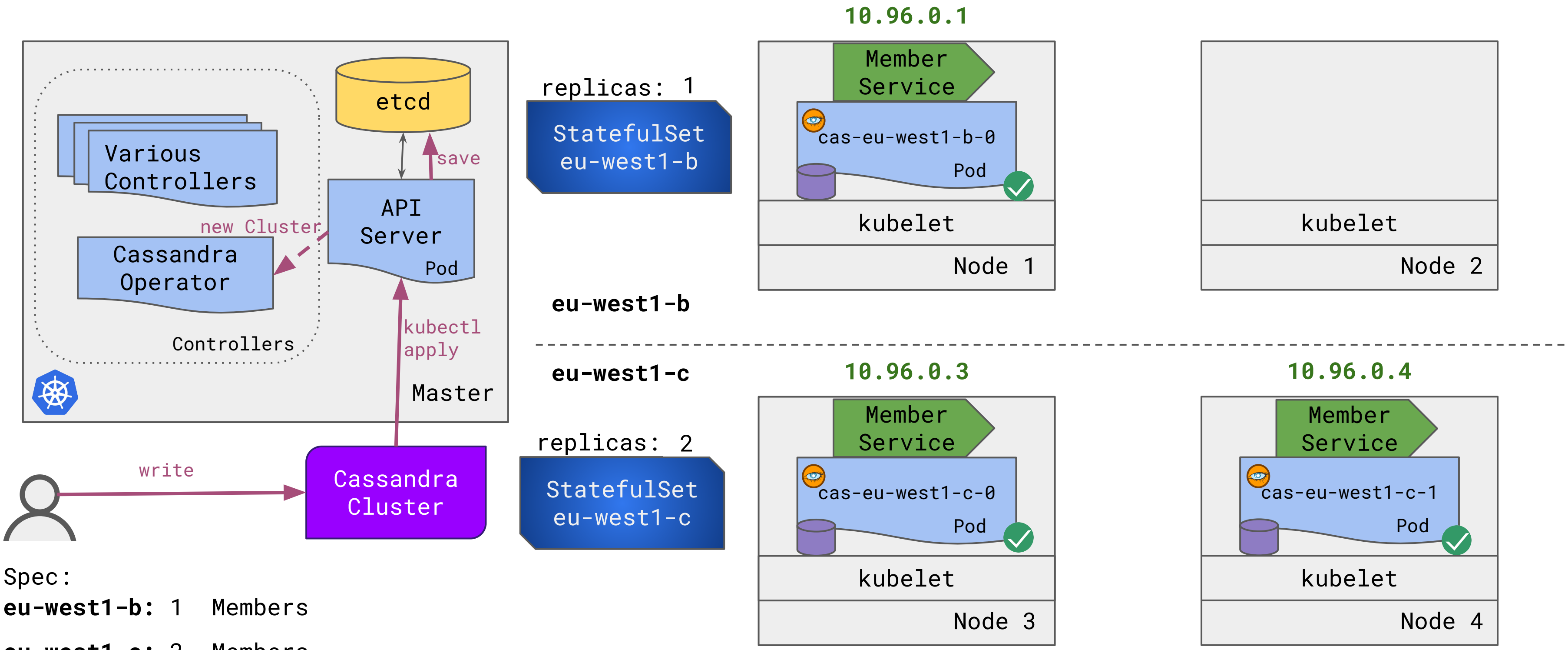
```yaml
apiVersion: cassandra.rook.io/v1alpha1
kind: Cluster
metadata:
  name: my-cassandra-cluster
spec:
  version: 3.11.1
  datacenter:
    name: europe-west-1
    racks:
      - name: europe-west1-b
        members: 3
status:
  racks:
    europe-west1-b:
      members: 2
      readyMembers: 1
```

Cassandra Operator - Current Status

- Deployment
- Scaling Up
- Multiple availability zone
- Safe Scale Down

https://github.com/rook/rook/issues/2294

Cluster Creation & Scale Up



Various Controllers

Cassandra Operator

new Cluster

Controllers

etcd

save

API Server

Pod

kubectl apply

Master

write

Cassandra Cluster

replicas: 1

StatefulSet eu-west1-b

replicas: 2

StatefulSet eu-west1-c

eu-west1-b

eu-west1-c

10.96.0.1

Member Service

cas-eu-west1-b-0

Pod

kubelet

Node 1

kubelet

Node 2

10.96.0.3

Member Service

cas-eu-west1-c-0

Pod

kubelet

Node 3

10.96.0.4

Member Service

cas-eu-west1-c-1

Pod

kubelet

Node 4

Spec:

eu-west1-b: 1 Members

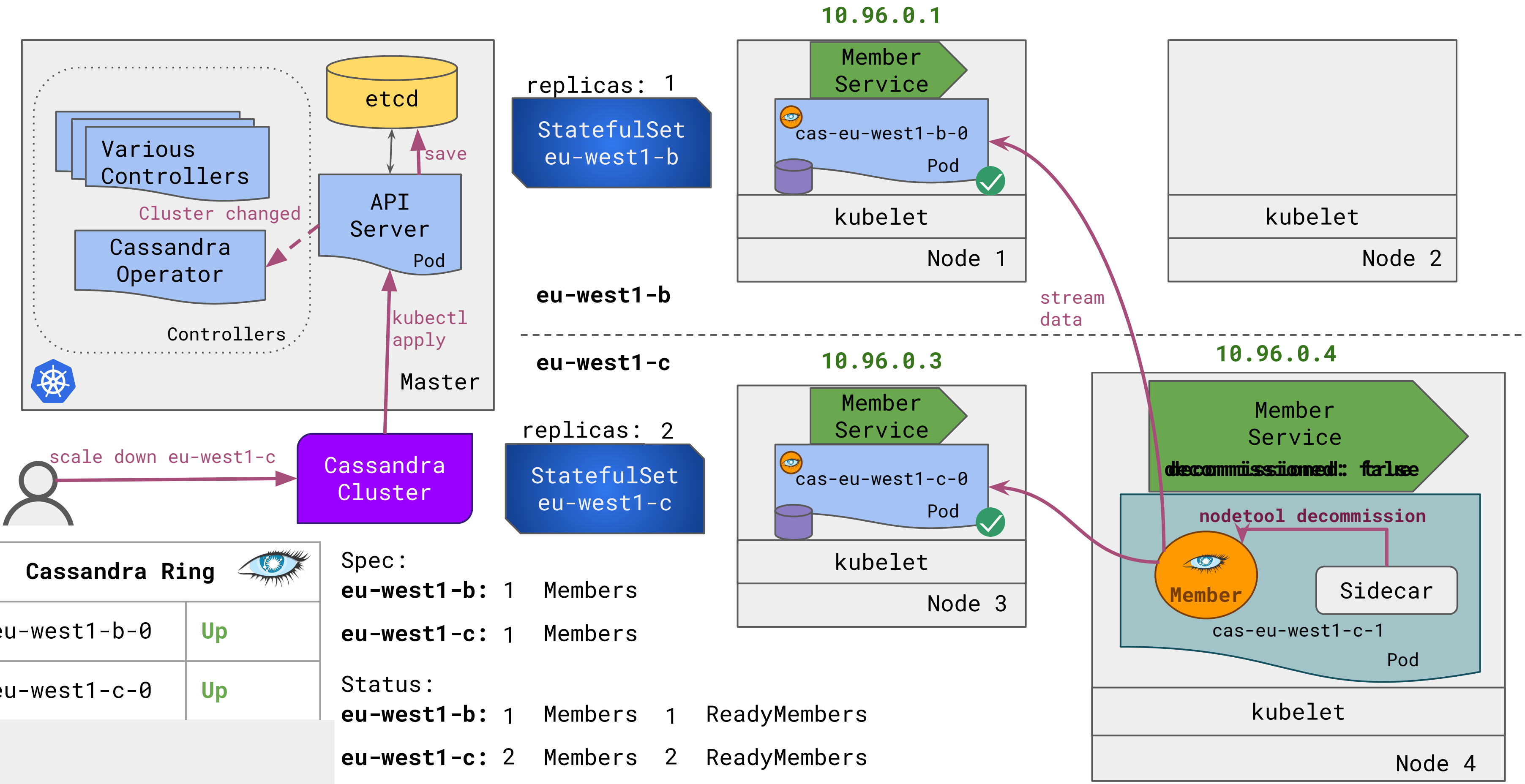
eu-west1-c: 2 Members

Status:

eu-west1-b: 1 Members 1 ReadyMembers

eu-west1-c: 2 Members 2 ReadyMembers

Scale Down



Cassandra Ring	
cas-eu-west1-b-0	Up
cas-eu-west1-c-0	Up

Spec:

eu-west1-b: 1 Members

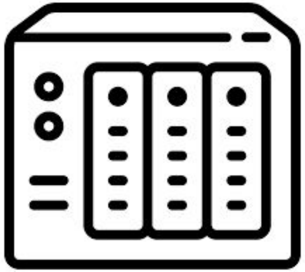
eu-west1-c: 1 Members

Status:

eu-west1-b: 1 Members 1 ReadyMembers

eu-west1-c: 2 Members 2 ReadyMembers

Local Storage vs Network Attached



Local NVME SSD

- Fast
- Ephemeral

Network Attached Storage
(AWS EBS, Google Persistent Disk)

- Slow
- Fault-tolerant

Cassandra handles replication => Use Local Storage!

v1.10: Local Persistent Volumes in Beta

Local Storage Failure Scenarios

- Disk Misbehaves
    - Block errors
    - Deteriorating performance

- Pod still runs
- Unhandled by K8s

- Disk Fails
    - Mount Point Disappears

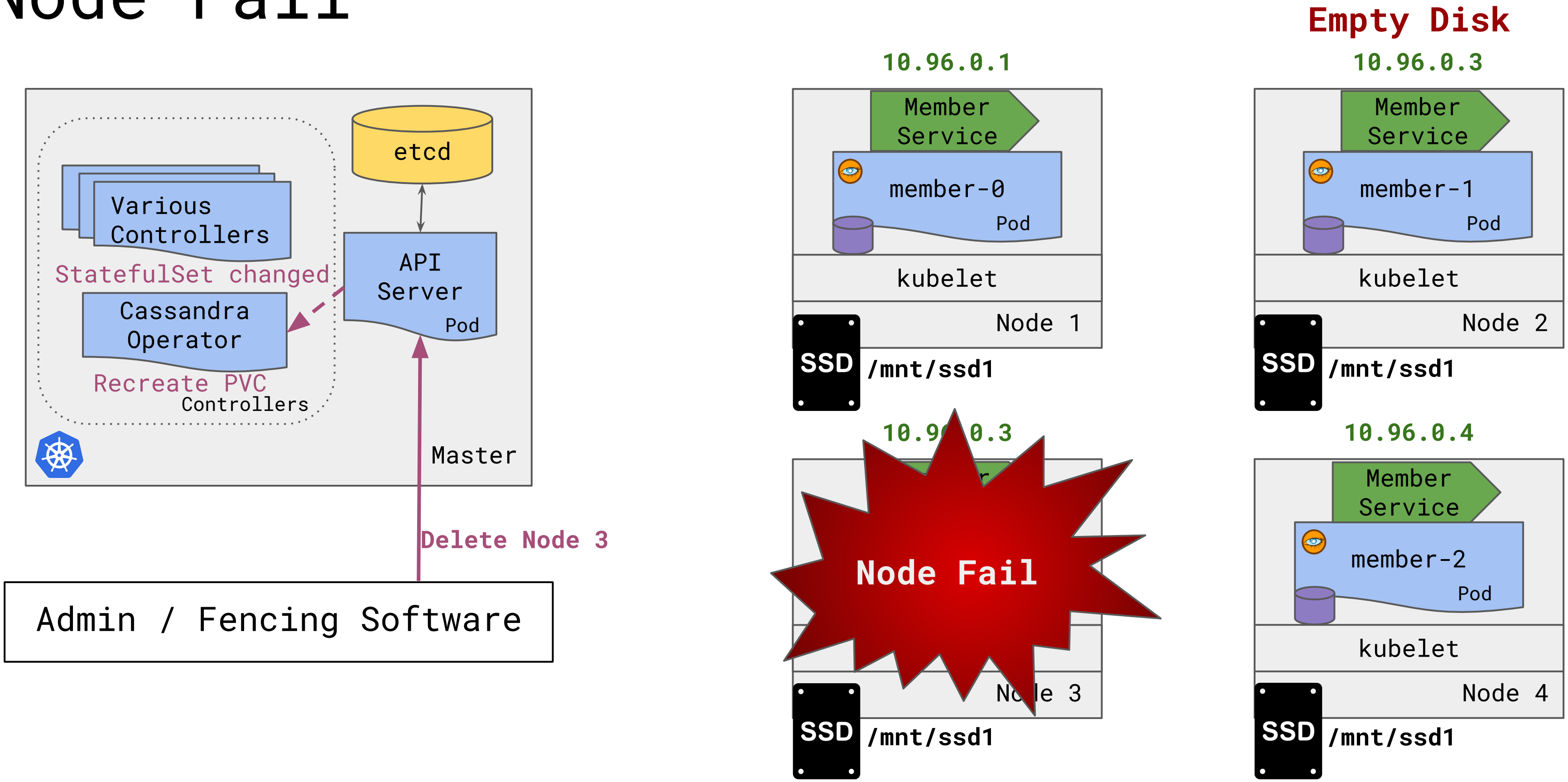
- Pod fails to start
- Unhandled by K8s

Common in the Cloud!

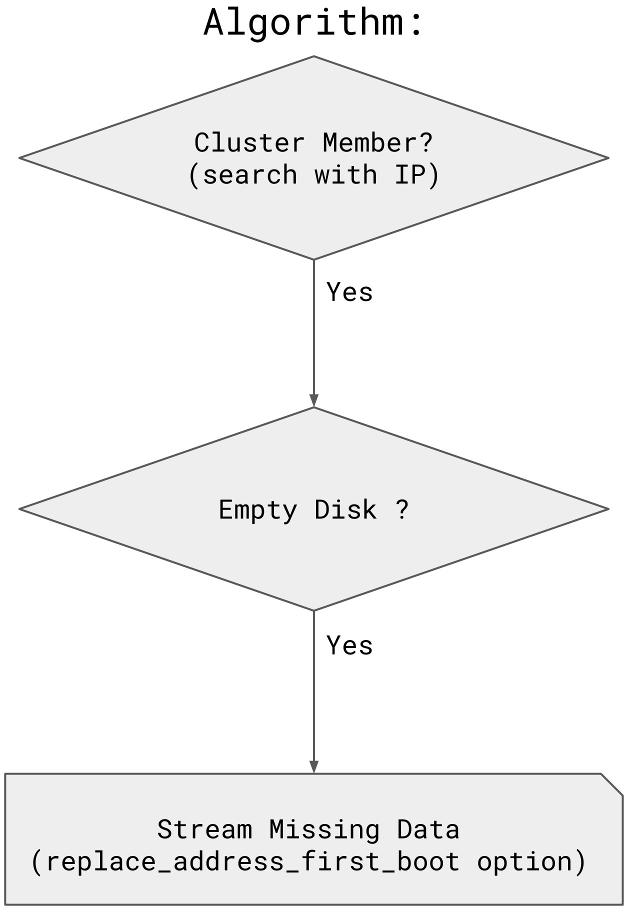
- Node Fails
    - With Disk on it

- Pod fails to be scheduled
- Unhandled by K8s

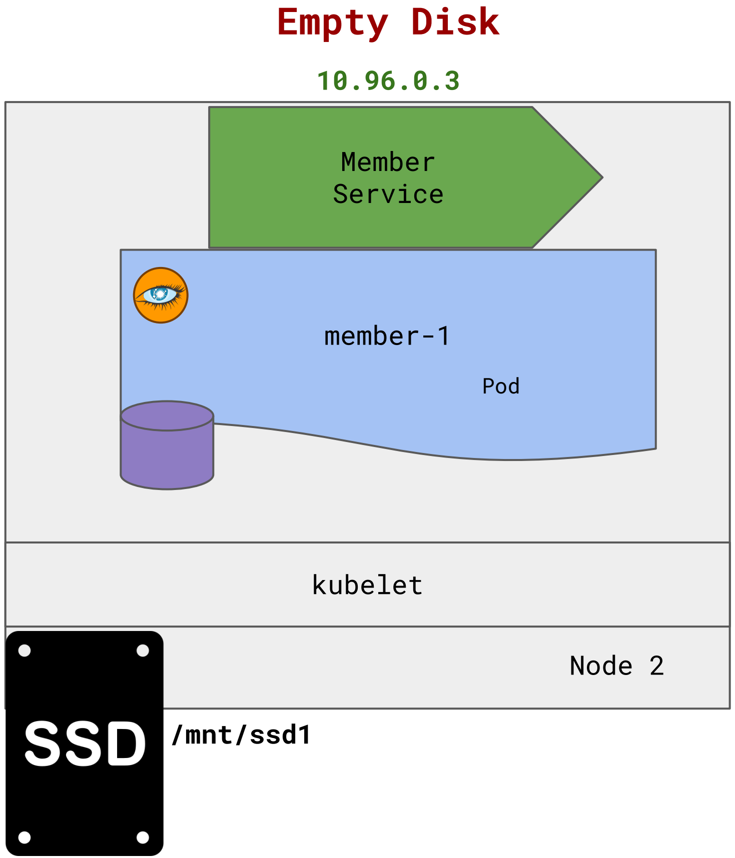
Node Fail


Empty Disk
etcd
Various Controllers
StatefulSet changed
Cassandra Operator
Recreate PVC
Controllers
API Server
Pod
Master
Delete Node 3
Admin / Fencing Software
10.96.0.1
Member Service
member-0
Pod
kubelet
Node 1
SSD
/mnt/ssd1
10.96.0.3
Member Service
member-1
Pod
kubelet
Node 2
SSD
/mnt/ssd1
10.96.0.3
Node Fail
Node 3
SSD
/mnt/ssd1
10.96.0.4
Member Service
member-2
Pod
kubelet
Node 4
SSD
/mnt/ssd1

Node Fail



Algorithm:



Monitoring

- Cassandra doesn’t expose prometheus metrics
- **Solution:** use a jmx-exporter

- Few good Grafana dashboards in the open
- **Solution:** community input and support

Pull Request in progress for v1.1.8

Repairs

- Cassandra uses repair to reduce entropy introduced by eventual consistency model
- Repair must be performed regularly and is by default a manual process

- **Solution:** integrate Cassandra Reaper, an automated repair solution by Spotify and The Last Pickle

Backups - Restores

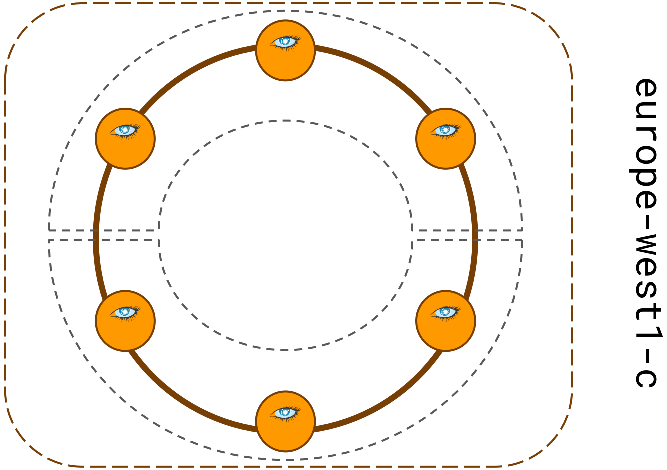
- CSI interface for snapshots/restores in K8s
- Leverage CSI to provide backup/restore of PVCs for Cassandra

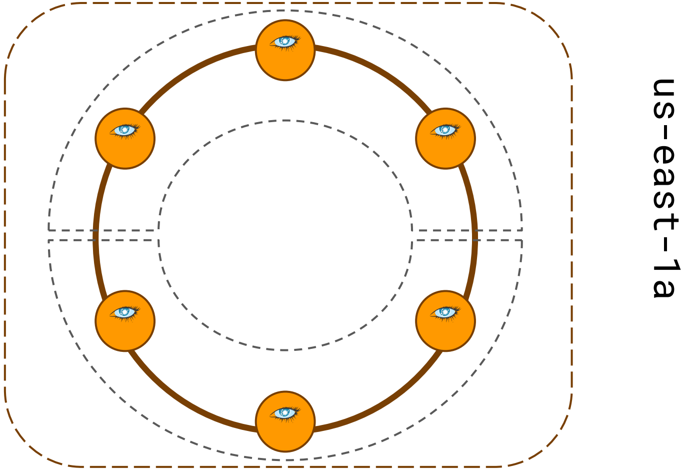
- Interesting use case: lost Node, restore from backup instead of streaming all data

- How do you do backups now?
- Tell us your experience!

Multi-Region Clusters

- Unsolved problem by most operators
- Needed for high scale Cassandra Clusters
- Experiment with designs & workarounds!





Contributors Welcome

- Want to run Cassandra on Kubernetes? Come along!

Plan:

1. Cassandra Administrators bring Cassandra expertise
2. Encode expertise and best-practices in the operator code
3. Everybody wins!