



kubernetes



KubeCon



CloudNativeCon

Europe 2020

Deploy Cassandra on Kubernetes with Datastax Cass Operator



#CloudNativeLon



About me @clunven



Director of Developer Advocacy



Creator and Maintainer of FF4j
Java and Spring Dinosaur



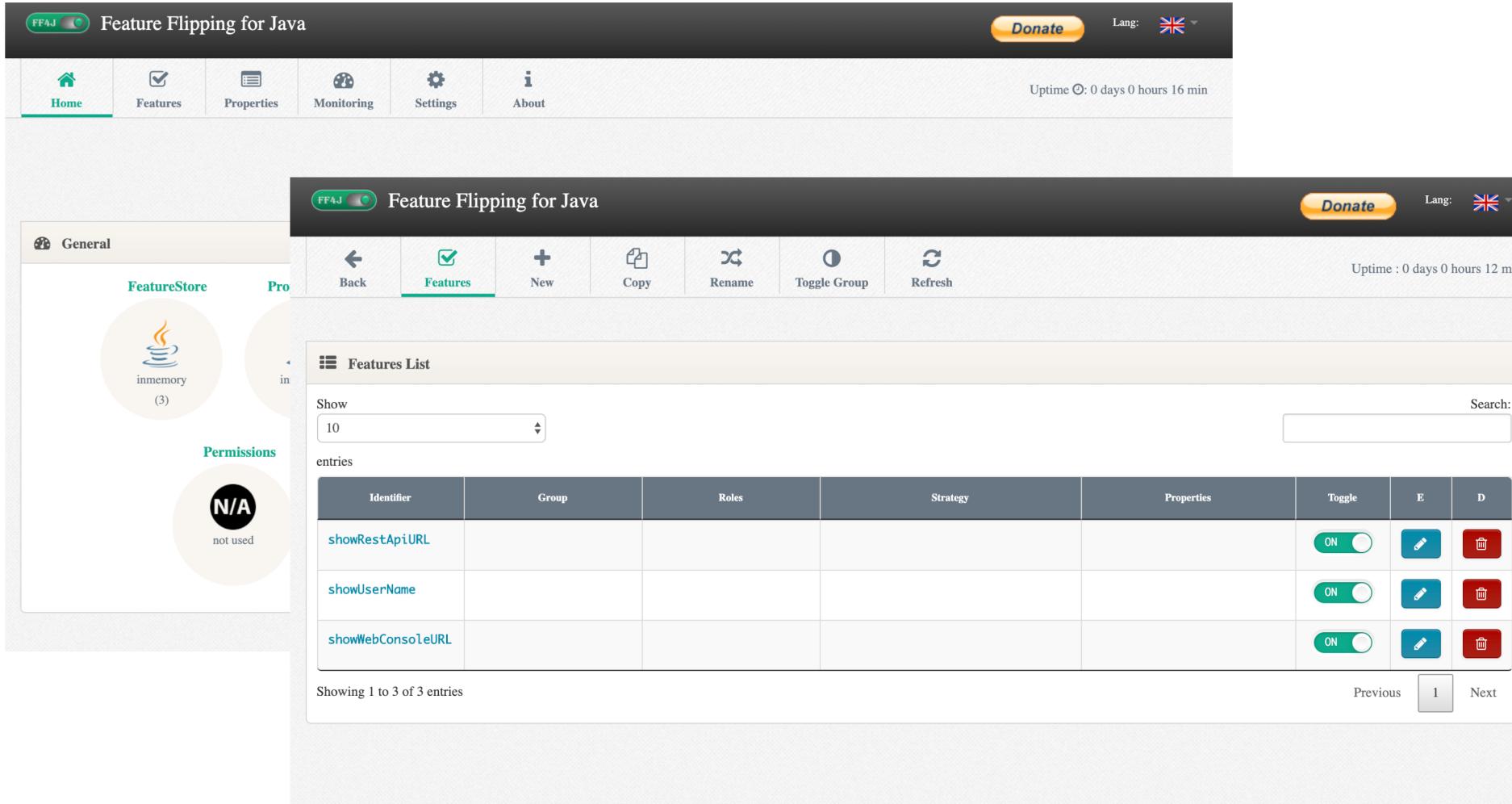
#CloudNativeLon | June 3rd | Deploy Cassandra on Kubernetes



@clunven

 FF4J

Feature Flags 4 Java



The screenshot shows the FF4J web application interface. At the top, there's a navigation bar with links for Home, Features, Properties, Monitoring, Settings, and About. A 'Donate' button and language selection (Lang: EN) are also present. Below the navigation, a status bar shows 'Uptime: 0 days 0 hours 16 min'. The main content area is titled 'Feature Flipping for Java' and displays a 'Features List'. On the left, there are two circular cards: one for 'inmemory' (3 entries) and another for 'Permissions' (N/A, not used). The 'Features' tab is selected in the navigation bar. The 'Features List' table has columns for Identifier, Group, Roles, Strategy, Properties, Toggle, and edit/delete icons. Three entries are listed: 'showRestApiURL', 'showUserName', and 'showWebConsoleURL', all currently set to 'ON'.

Identifier	Group	Roles	Strategy	Properties	Toggle	E	D
showRestApiURL					<input checked="" type="checkbox"/>		
showUserName					<input checked="" type="checkbox"/>		
showWebConsoleURL					<input checked="" type="checkbox"/>		

ff4j.org

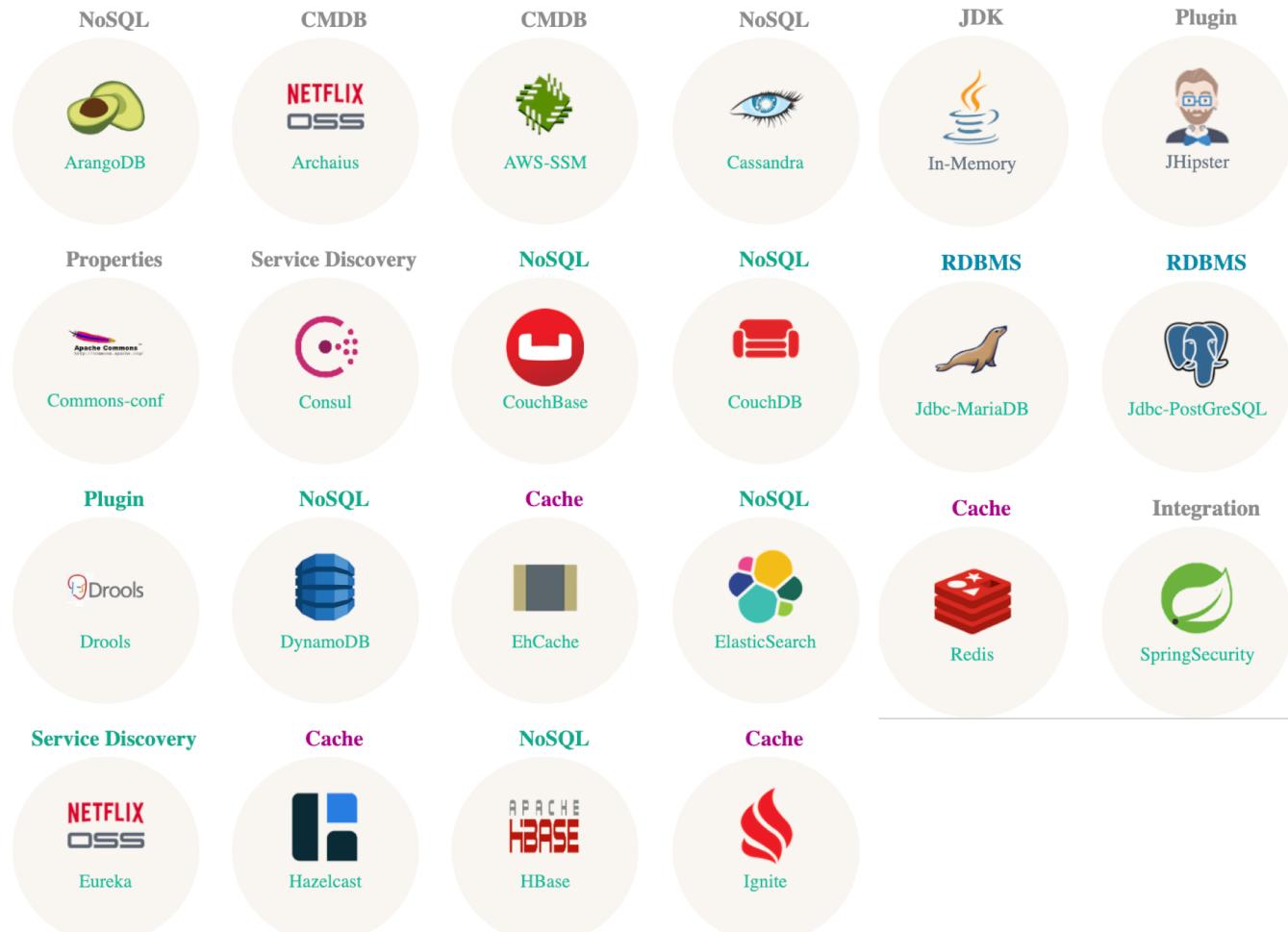


#CloudNativeLon | June 3rd | Deploy Cassandra on Kubernetes

 DataStax

@clunven

Feature Flags 4 Java



Apache Cassandra™ with Kubernetes

1

2

3

4

Reminders on Apache Cassandra™

Cassandra and Containers

Kubernetes and Stateful Applications

Cass Operator



Apache Cassandra™ with Kubernetes

1

2

3

4

Reminders on Apache Cassandra™

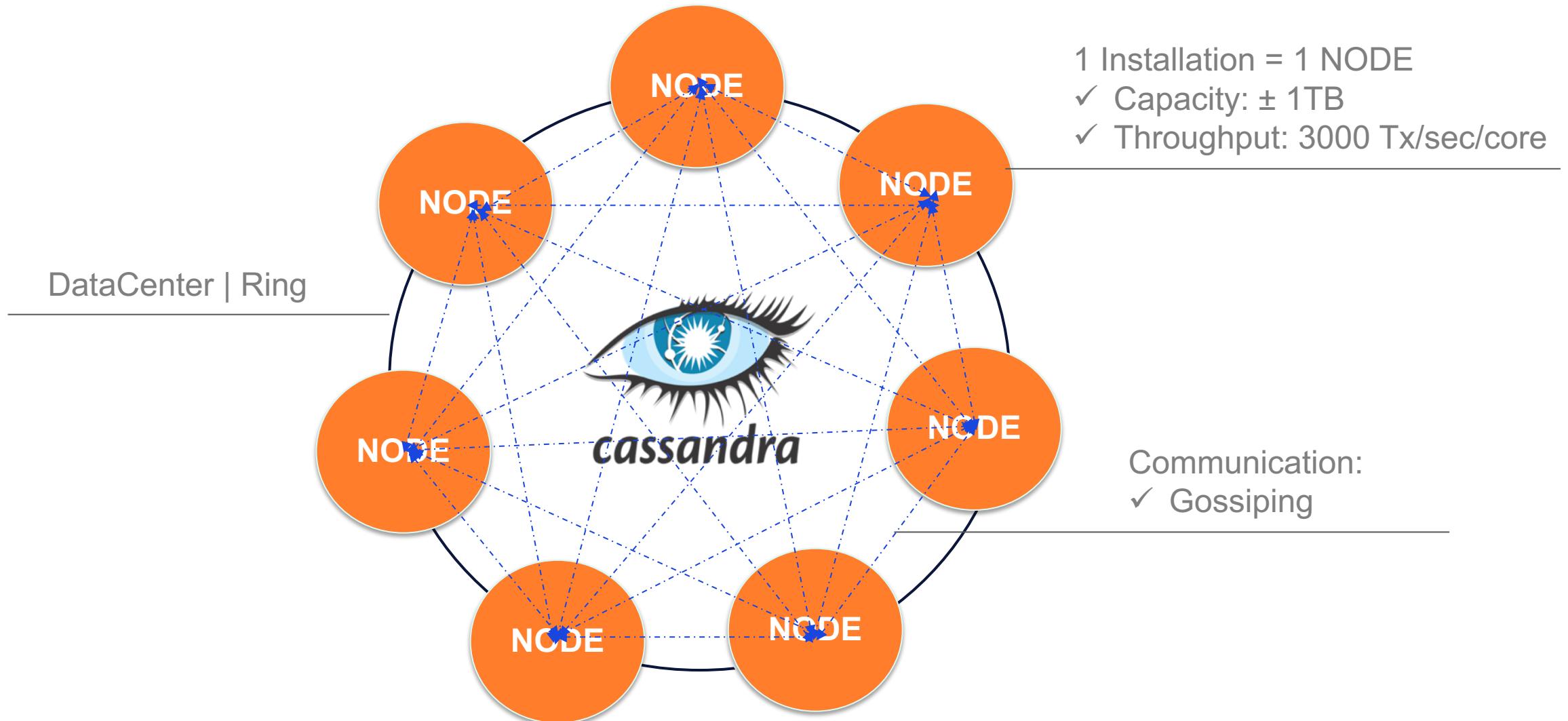
Cassandra and Containers

Kubernetes and Stateful Applications

Cass Operator



Apache Cassandra™ = NoSQL Distributed Database

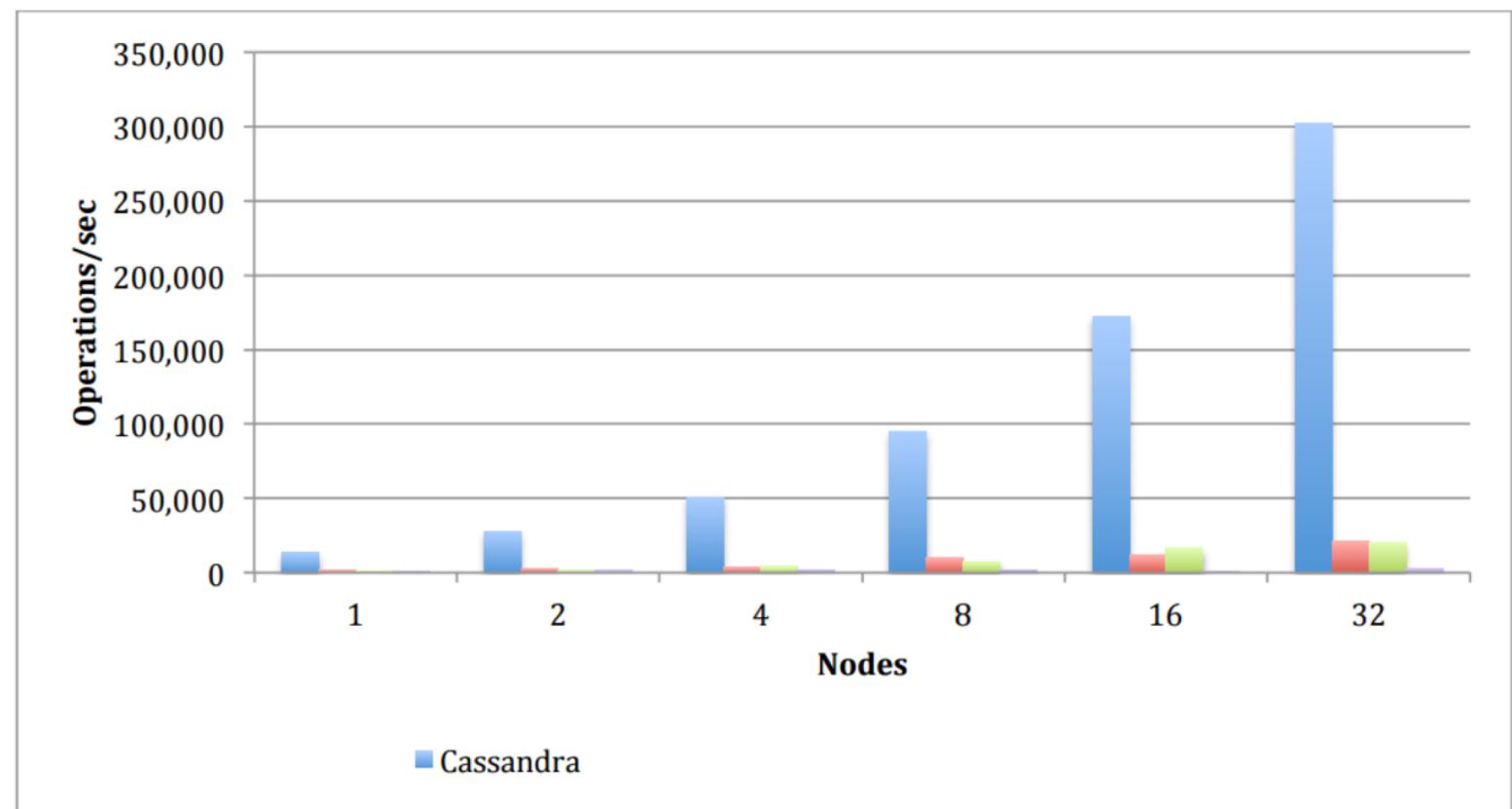




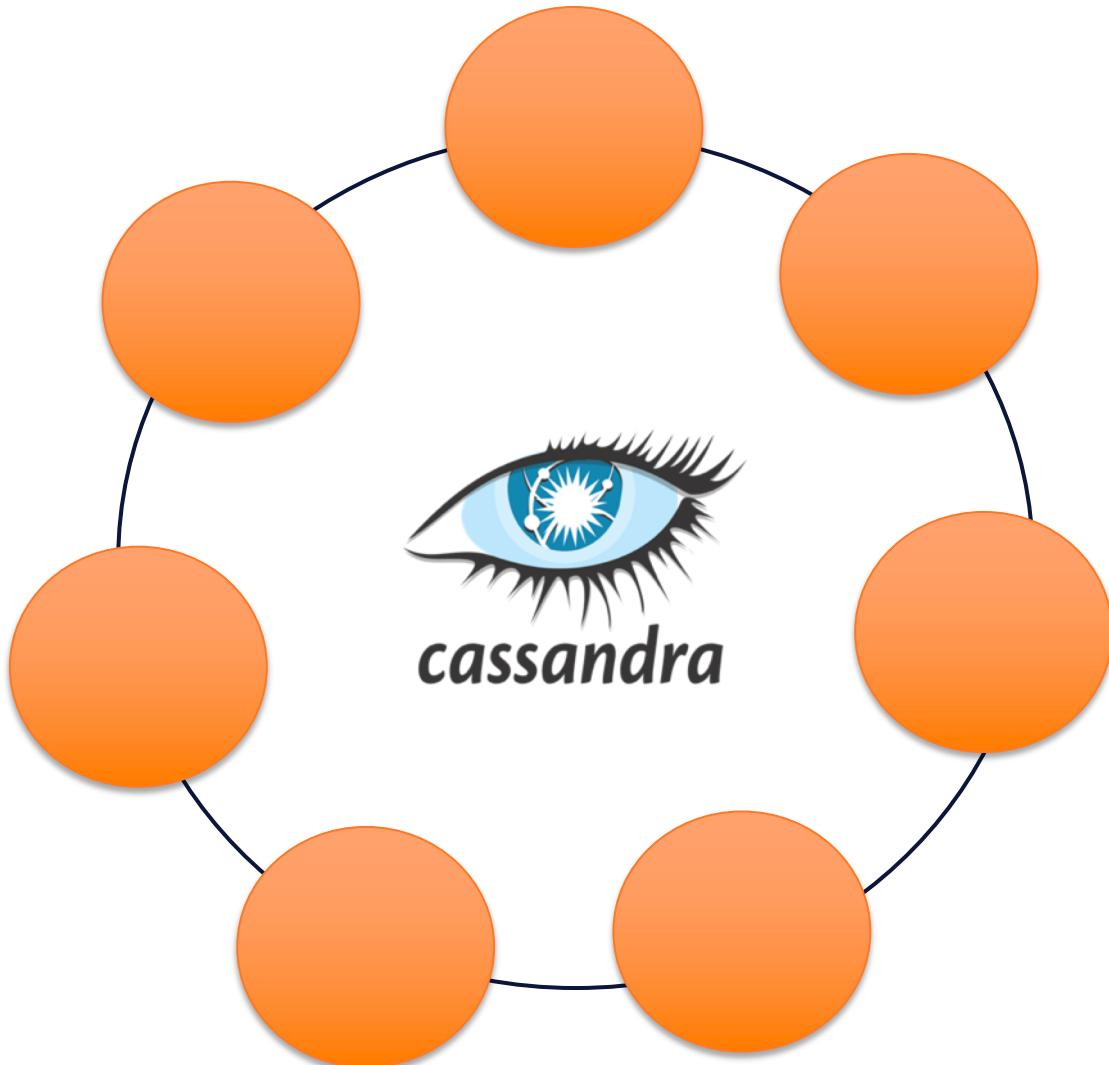
Scales Linearly

- Need more capacity?
- Need more throughput?
- Add nodes!

Balanced Read/Write Mix



Data is Distributed



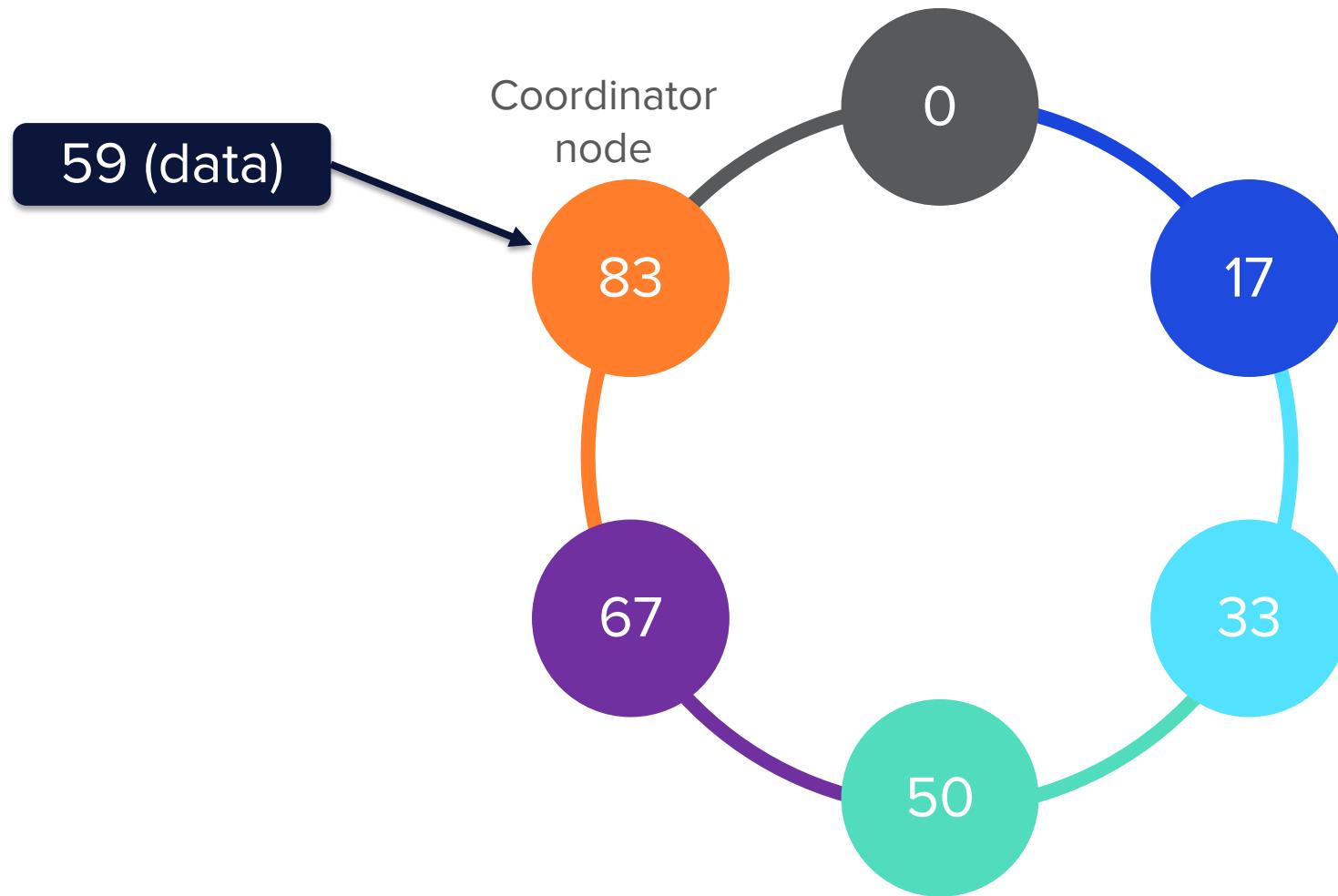
Country	City	Habitant
USA	New York	8.000.000
USA	Los Angeles	4.000.000
FR	Paris	2.230.000
DE	Berlin	3.350.000
UK	London	9.200.000
AU	Sydney	4.900.000
DE	Nuremberg	500.000
CA	Toronto	6.200.000
CA	Montreal	4.200.000
FR	Toulouse	1.100.000
JP	Tokyo	37.430.000
IN	Mumbai	20.200.000

Partition Key



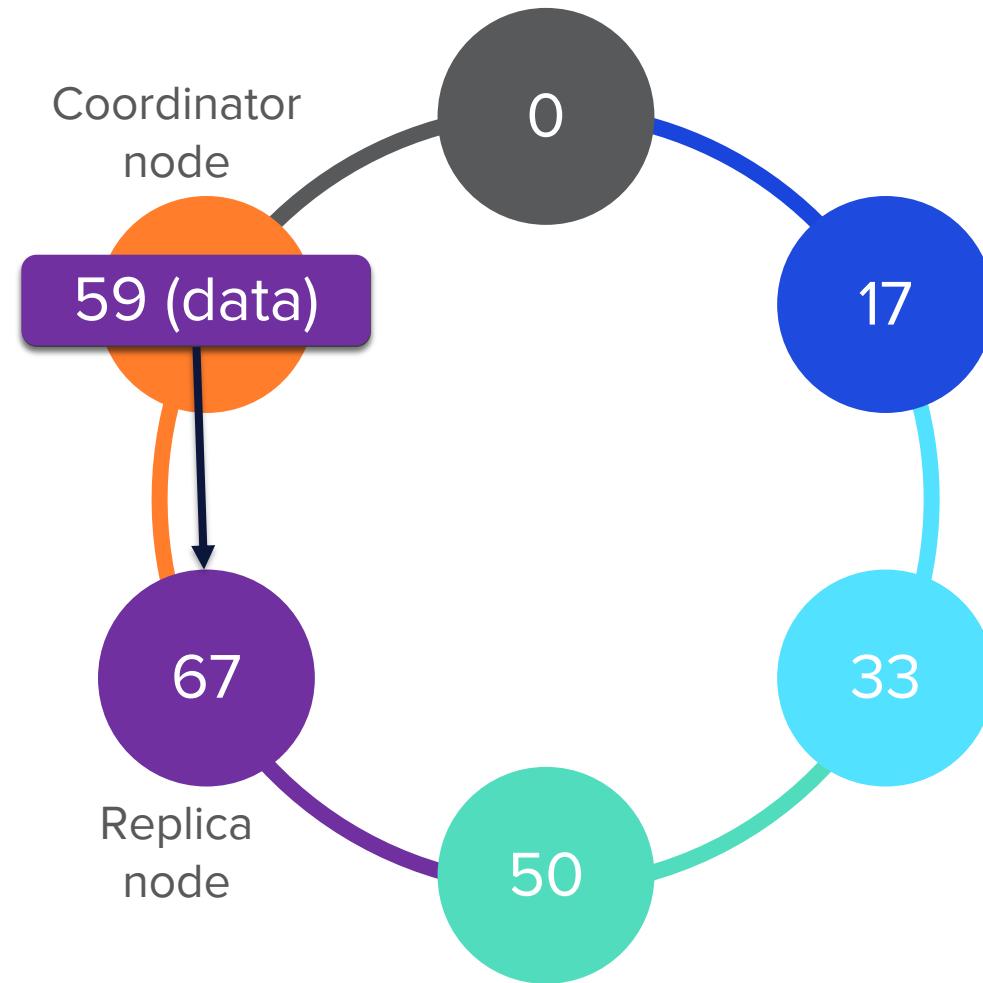


How the Ring Works



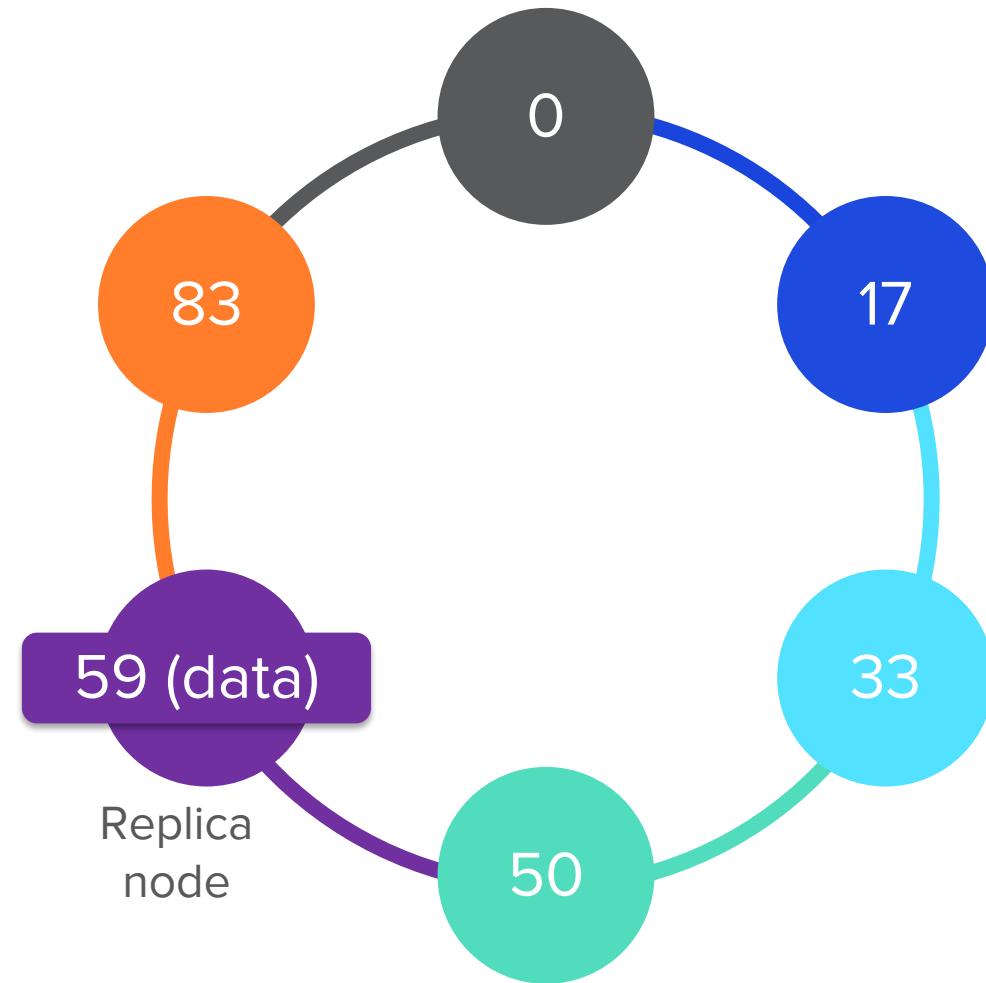


How the Ring Works





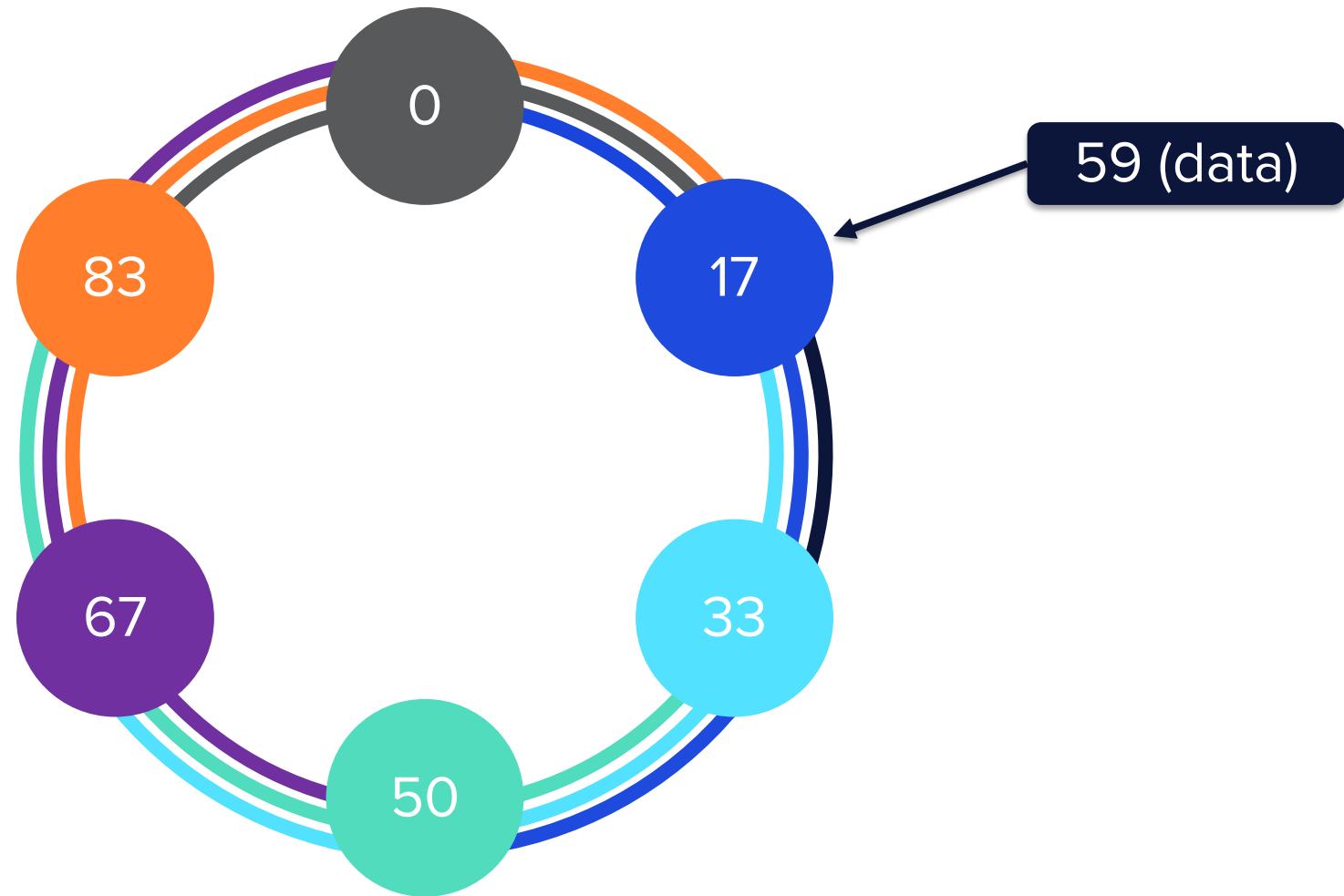
How the Ring Works





Replication within the Ring

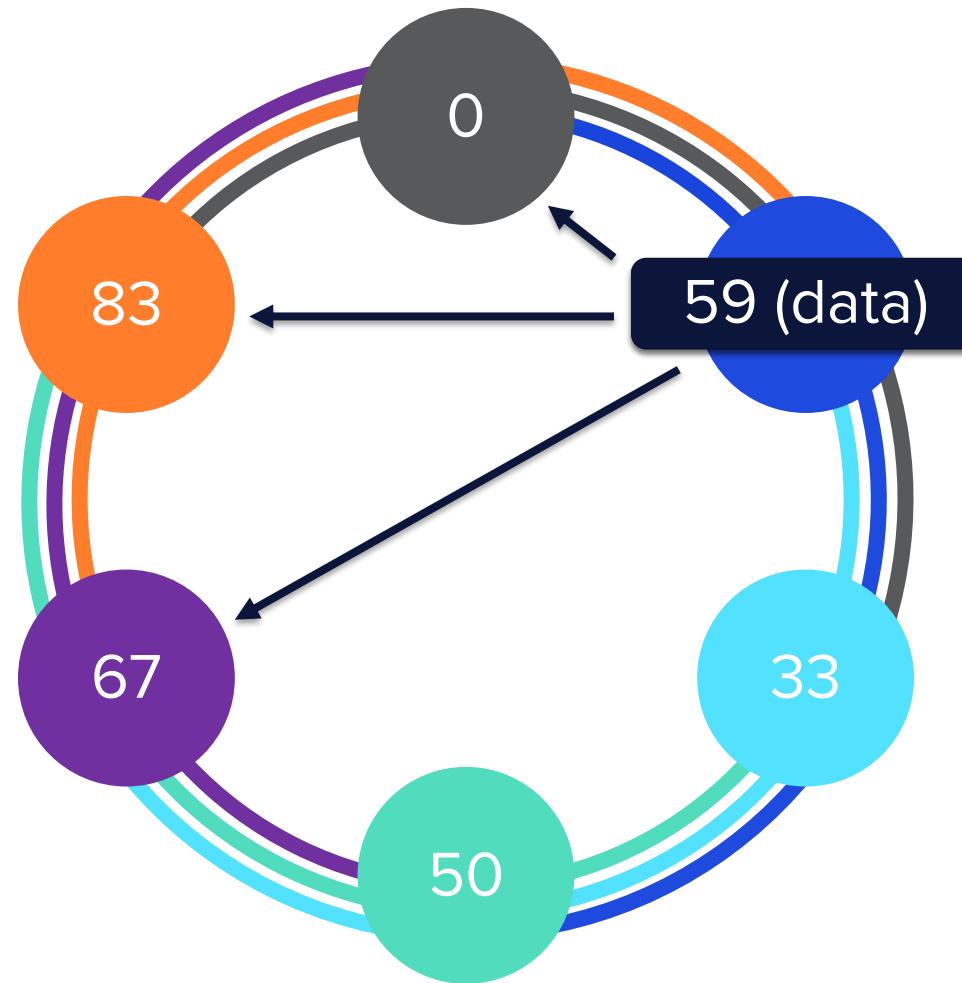
RF = 3





Replication within the Ring

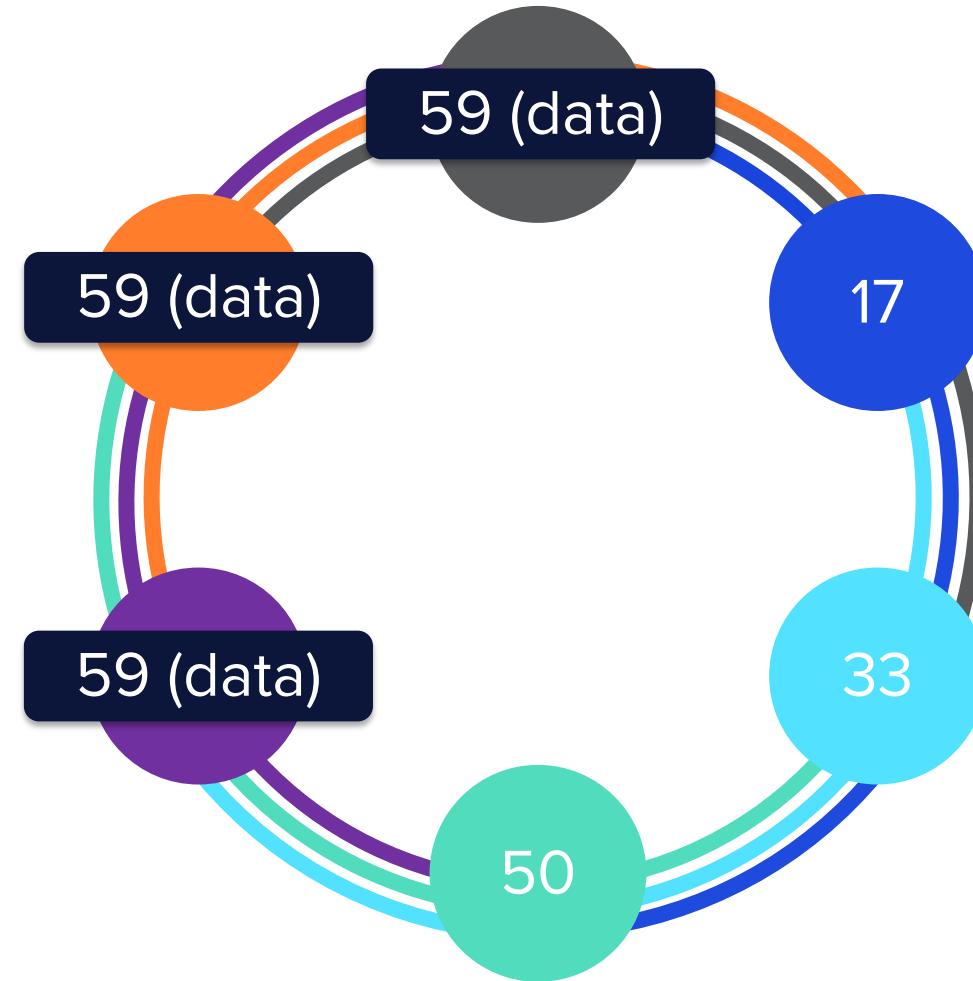
RF = 3





Replication within the Ring

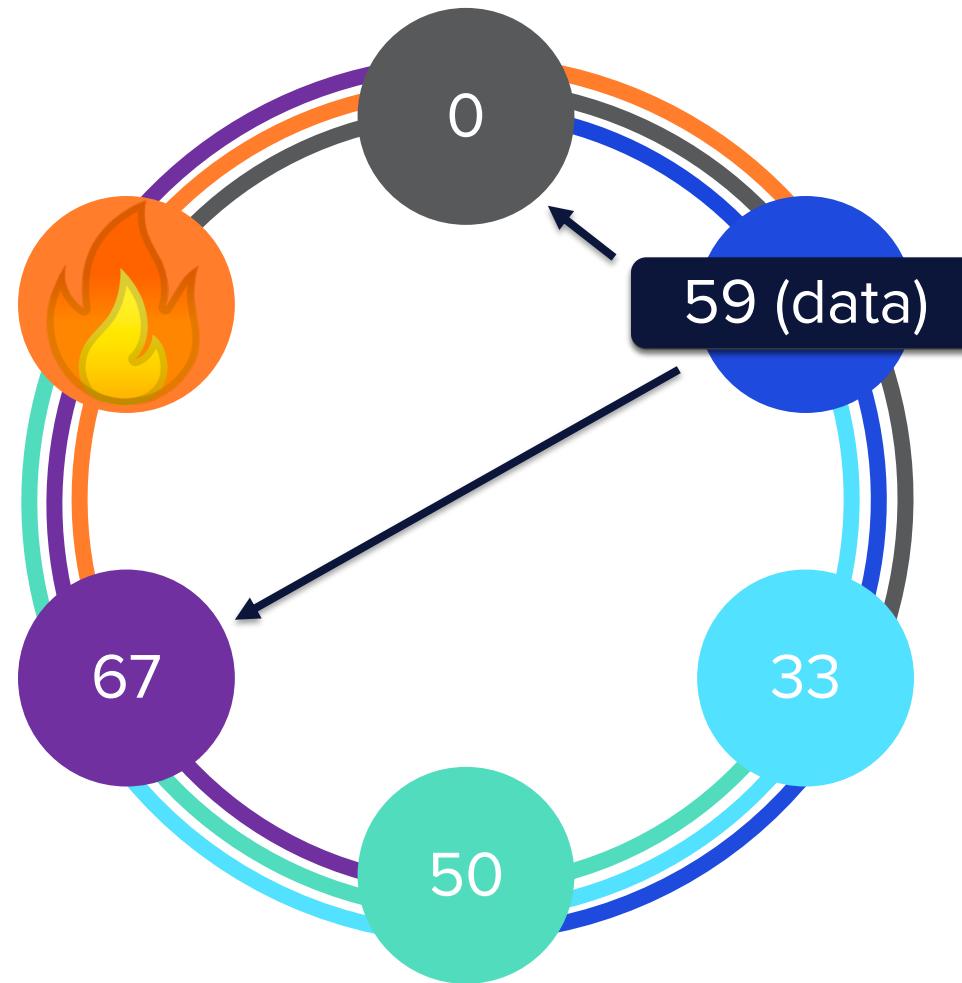
RF = 3





Node Failure

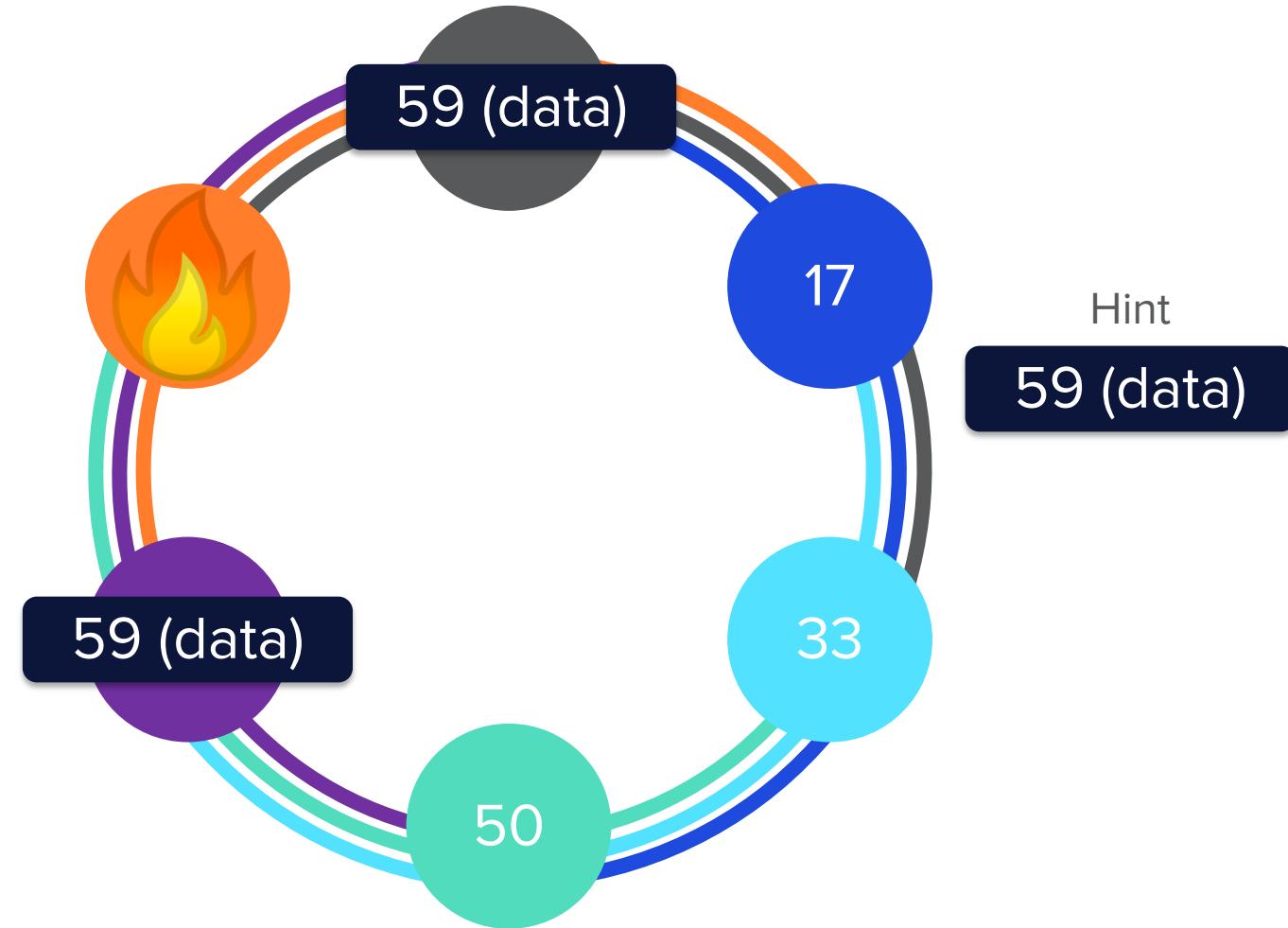
RF = 3





Node Failure

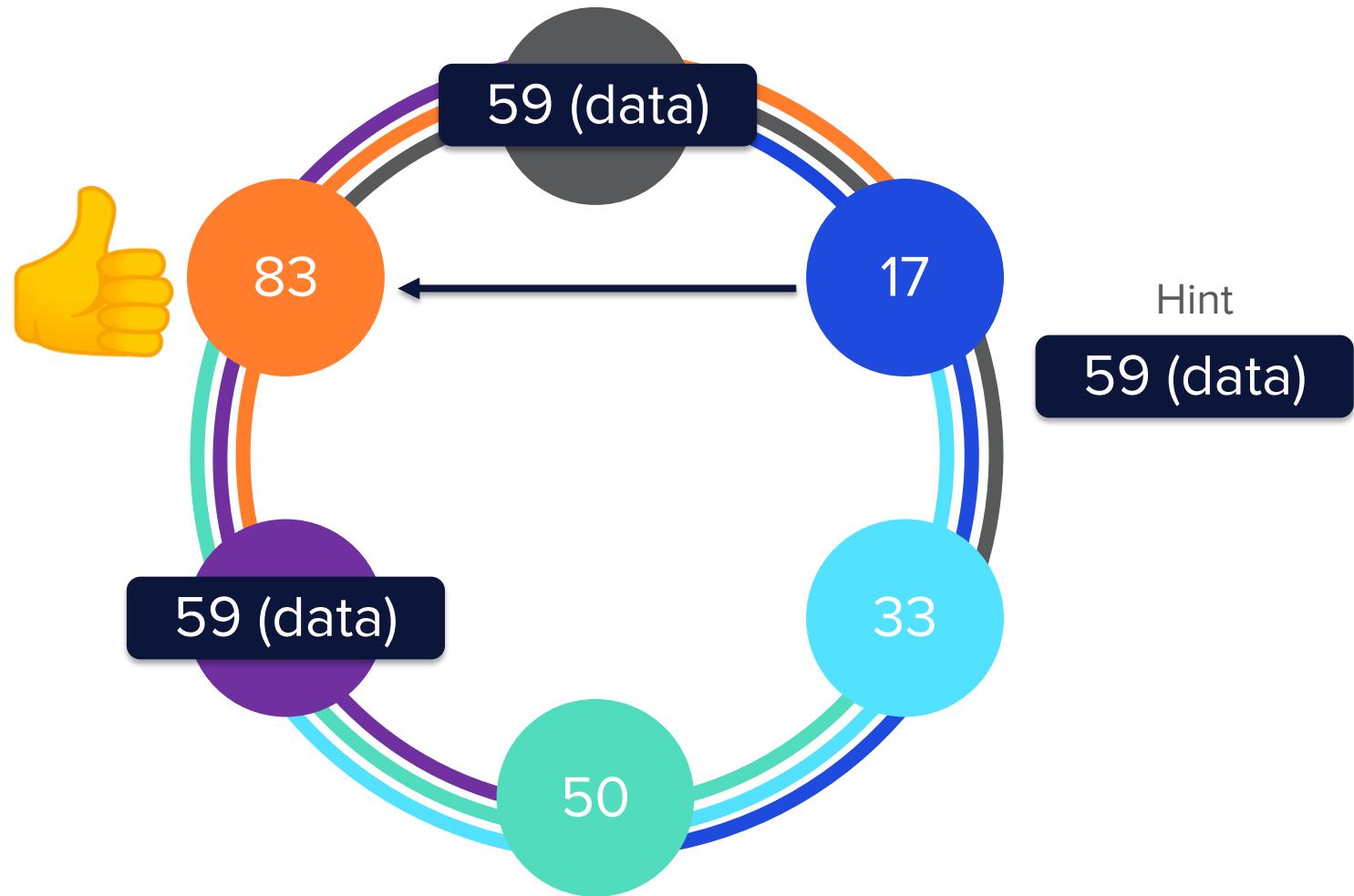
RF = 3



Node Failure



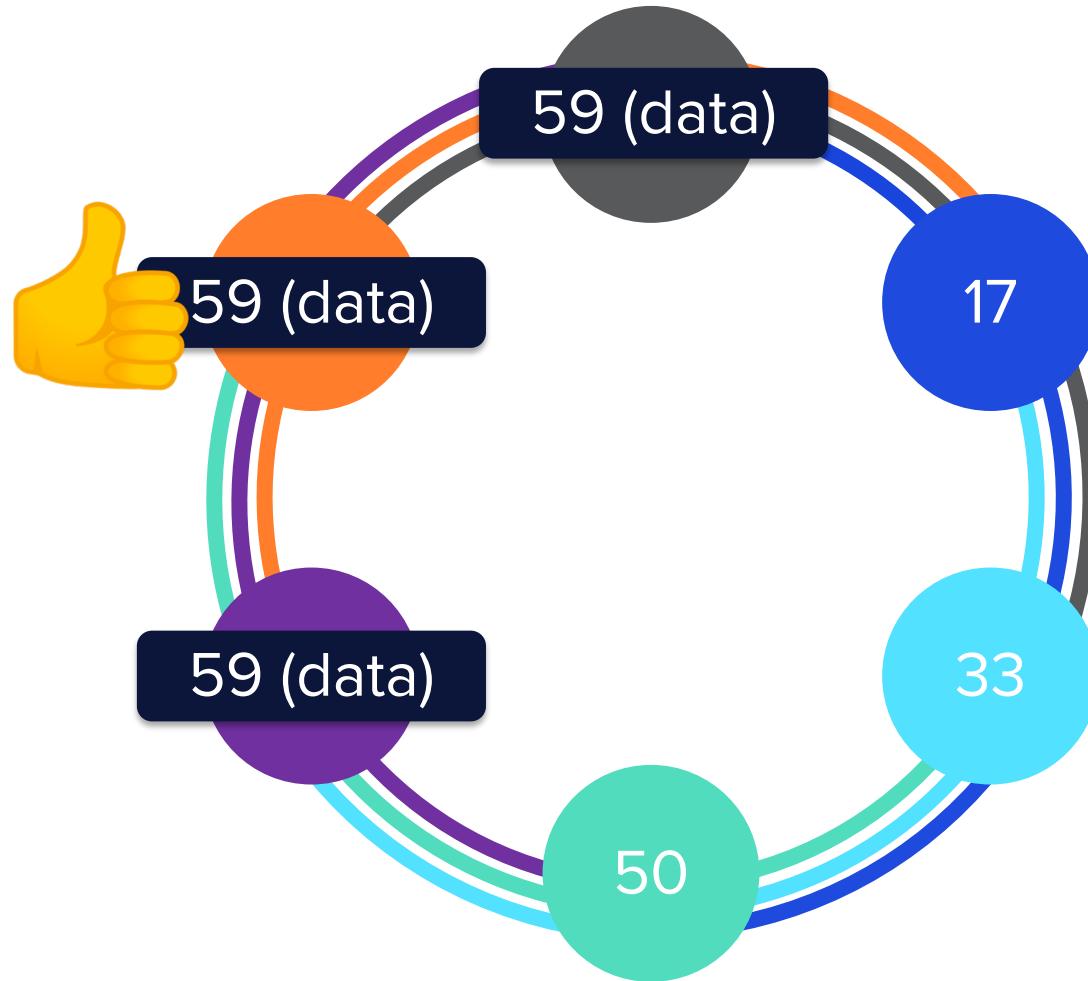
RF = 3





Self Healing

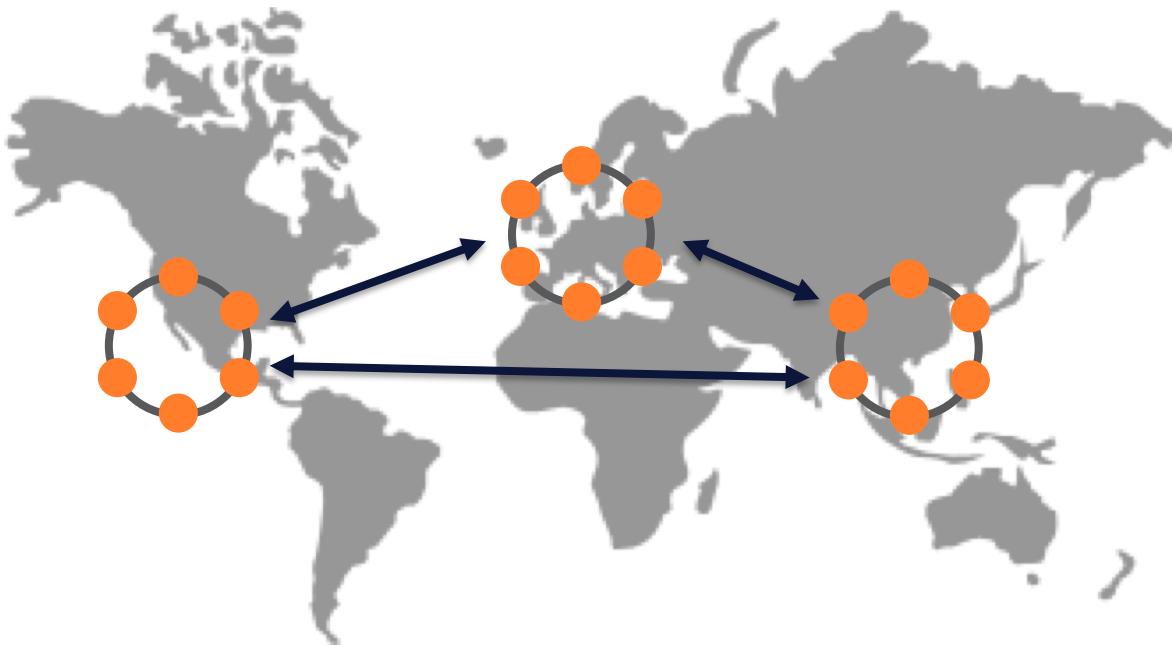
RF = 3



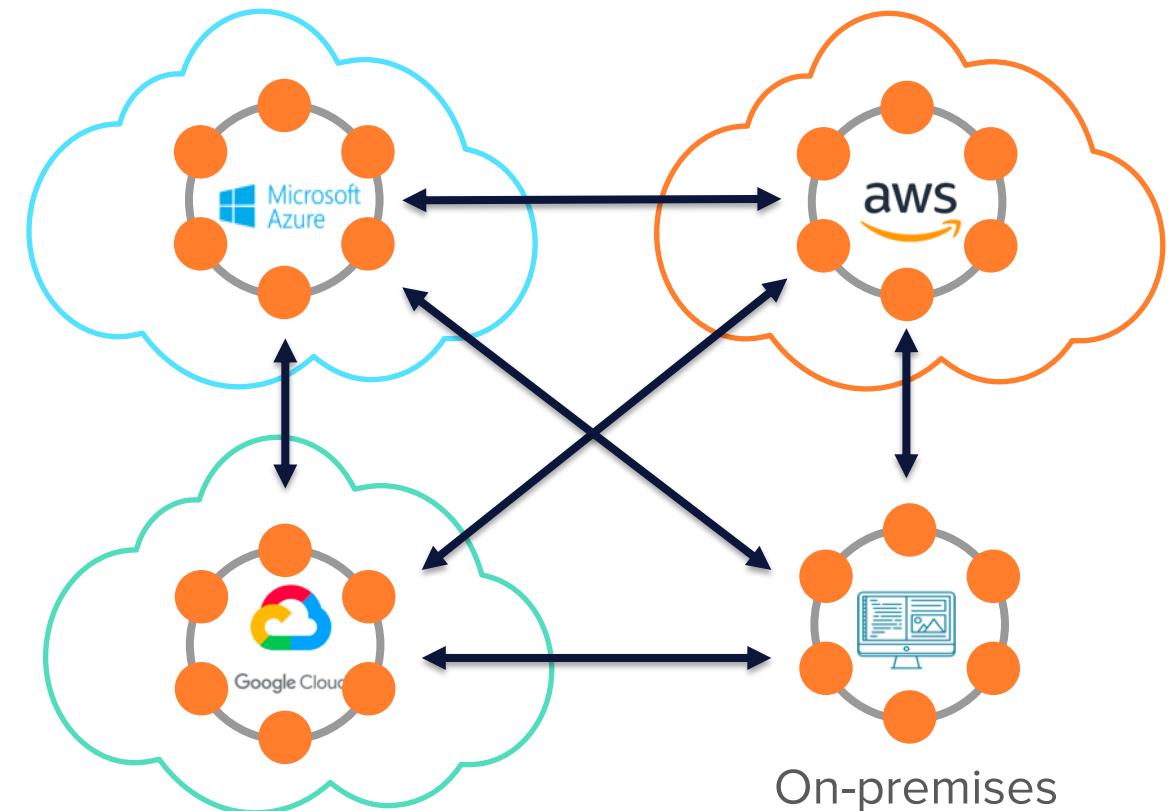


Data Distributed Everywhere

- Geographic Distribution

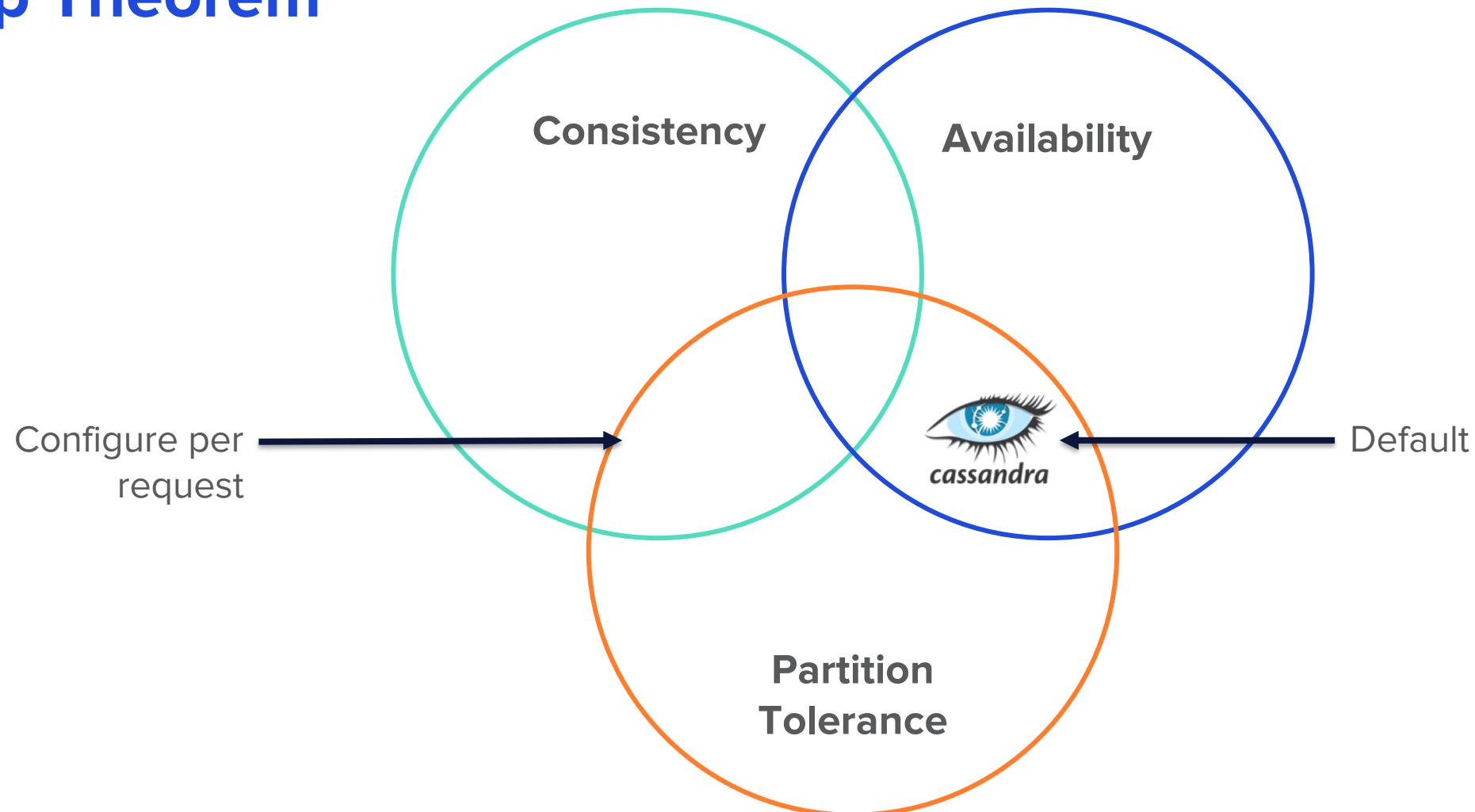


- Hybrid-Cloud and Multi-Cloud

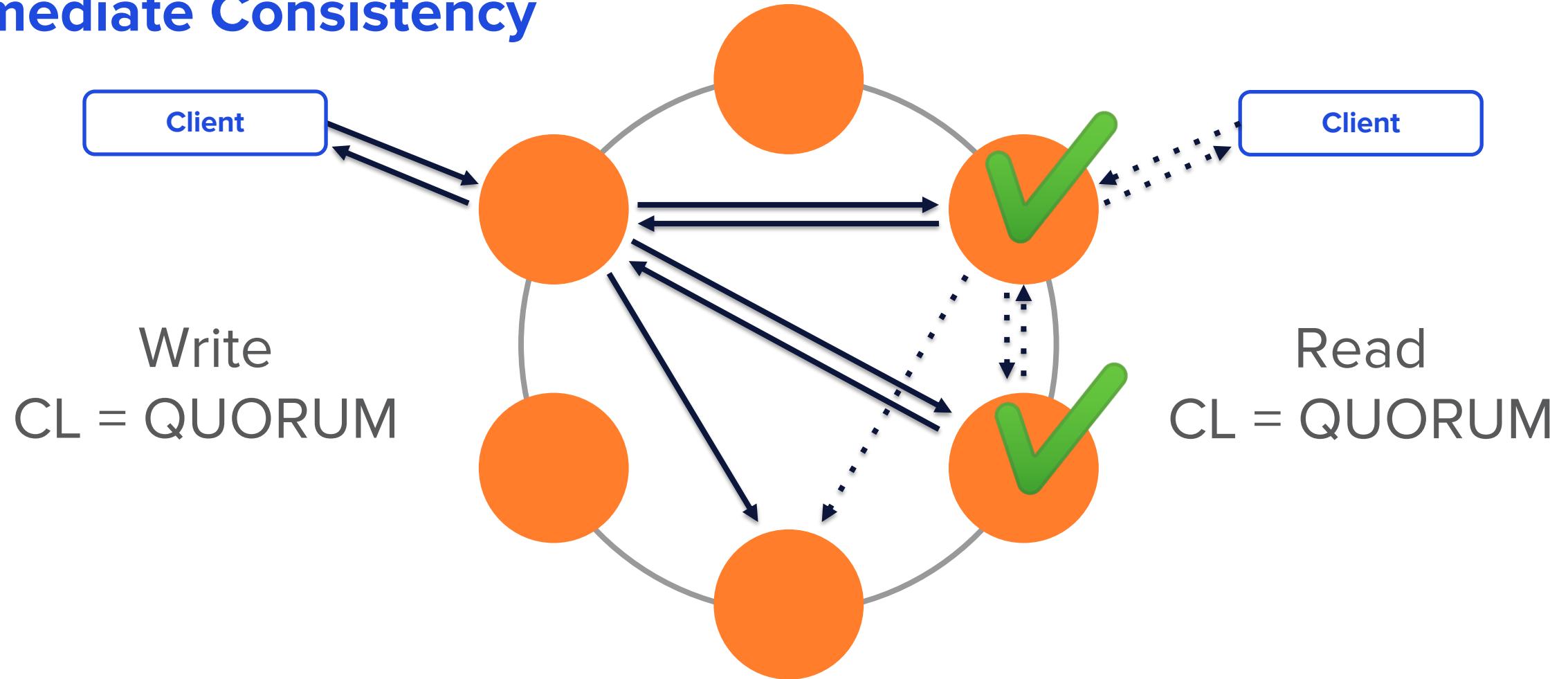




Cap Theorem



Immediate Consistency



Strongly Consistent : $CL_{read} + CL_{write} > RF$





Understanding Use Cases

Scalability

High Throughput
High Volume



Heavy Writes
Heavy Reads



Event Streaming	
Internet of Things	
Log Analytics	Other Time Series
Other Time Series	Log Analytics

Availability

Mission-Critical



No Data Loss
Always-on



Caching	
Market Data	
Pricing	Inventory
Inventory	Pricing

Distributed

Global Presence
Workload Mobility



Compliance /
GDPR



Banking	
Tracking /	
Logistics	Retail
Retail	Tracking /
Customer Experience	Logistics

Cloud-native

Modern Cloud Applications



API Layer	
Enterprise Data Layer	
Hybrid-cloud	Multi-cloud
Multi-cloud	Hybrid-cloud



Apache Cassandra™ with Kubernetes

1

2

3

4

Reminders on Apache Cassandra™

Cassandra and Containers

Kubernetes and Stateful Applications

Cass Operator





cassandra ☆

[Docker Official Images](#)

Apache Cassandra is an open-source distributed storage system.

100M+

Container

Linux

PowerPC 64 LE

ARM

ARM 64

386

x86-64

Databases

Official Image

Linux - x86 (latest)

Copy and paste to pull this image

`docker pull cassandra`



[View Available Tags](#)

Running Cassandra in Docker

- Define a proper [network](#)
- [Env variables](#) can be defined to override keys in `cassandra.yaml`.
- Export ports **7000, 9042, ...**
- Define volumes to stores data
 - **/var/lib/cassandra**

```
$ docker run  
  
--name some-cassandra -d \  
  
-e CASSANDRA_BROADCAST_ADDRESS=10.42.42.42 \  
  
-p 7000:7000,9042:9042  
  
-v /my/own/datadir:/var/lib/cassandra \  
  
cassandra:tag
```

docker-compose

```
docker-compose -f docker-compose.yml up -d --scale cassandra-node=2
```

Define and run multi-container Docker applications through the use of a **YAML** file to configure your applications

```
version: '2'
services:

cassandra-seed:
  container_name: cassandra-seed-node
  image: cassandra: 3.11.6
  ports:
    - "9042:9042"      # Native transport
    - "7199:7199"      # JMX
    - "9160:9160"      # Thrift clients

cassandra-node:
  image: cassandra: 3.11.6
  command: /bin/bash -c "echo 'Waiting for seed node' && sleep 30 && /docker-entrypoint.sh cassandra -f"
  environment:
    - "CASSANDRA_SEEDS=cassandra-seed-node"
  depends_on:
    - "cassandra-seed"
```

Apache Cassandra™ with Kubernetes

1

2

3

4

Reminders on Apache Cassandra™

Cassandra and Containers

Kubernetes and Stateful Applications

Cass Operator

Kubernetes Infrastructure



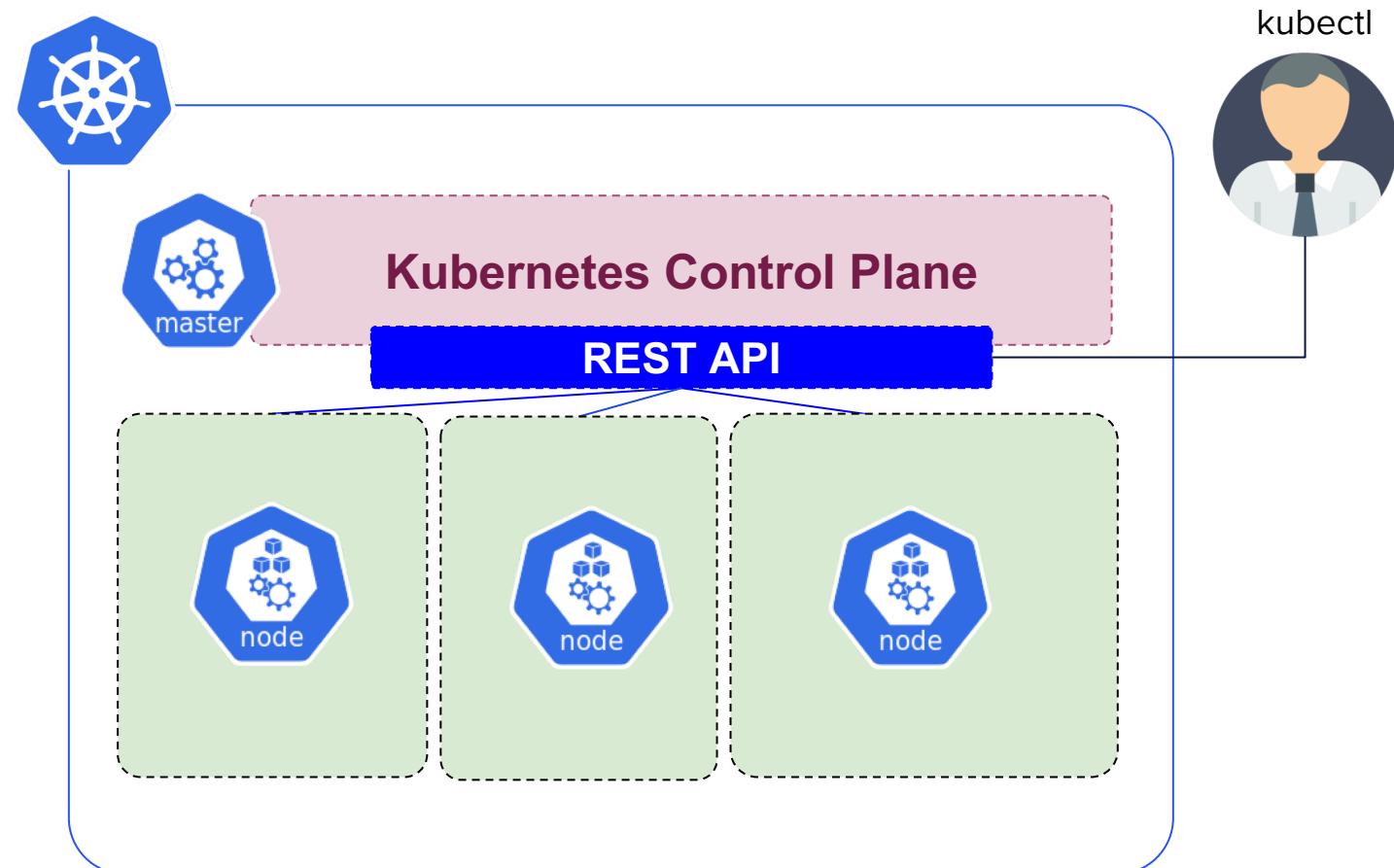
Cluster:
Kubernetes cluster.



Master:
Kubernetes Control Plane.



Node:
Worker machine in Kubernetes cluster.



Kubernetes Control Plane (master)



K8s API Server
Kubernetes API.



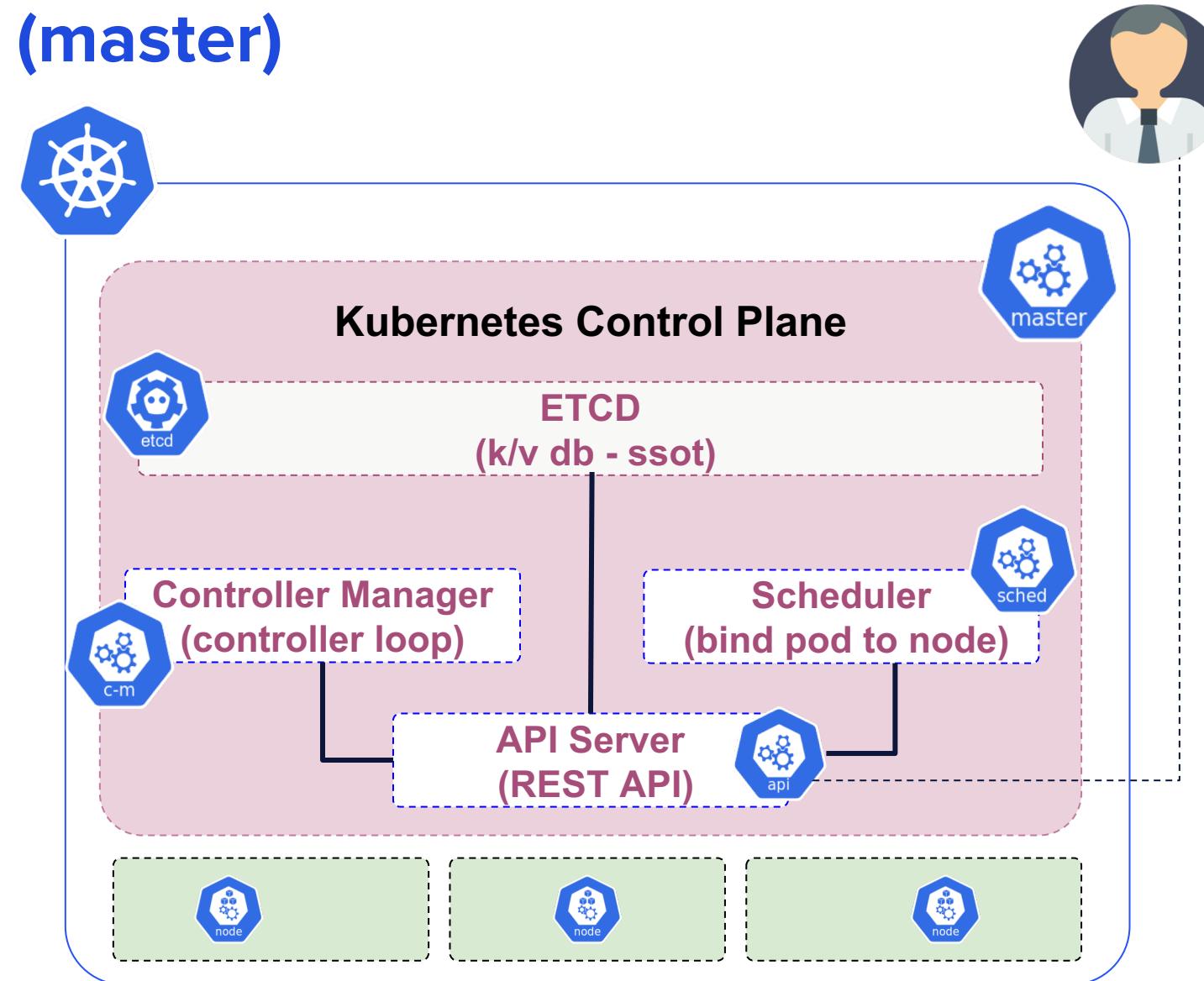
Controller Manager
Kubernetes controller manager.



Scheduler
In charge of ensuring Pods placement.



ETCD
Kubernetes' backing store.



Kubernetes Worker (Node, Minions)



Kubelet:

The kubelet is the primary “node agent” that runs on each node.



Kube-proxy

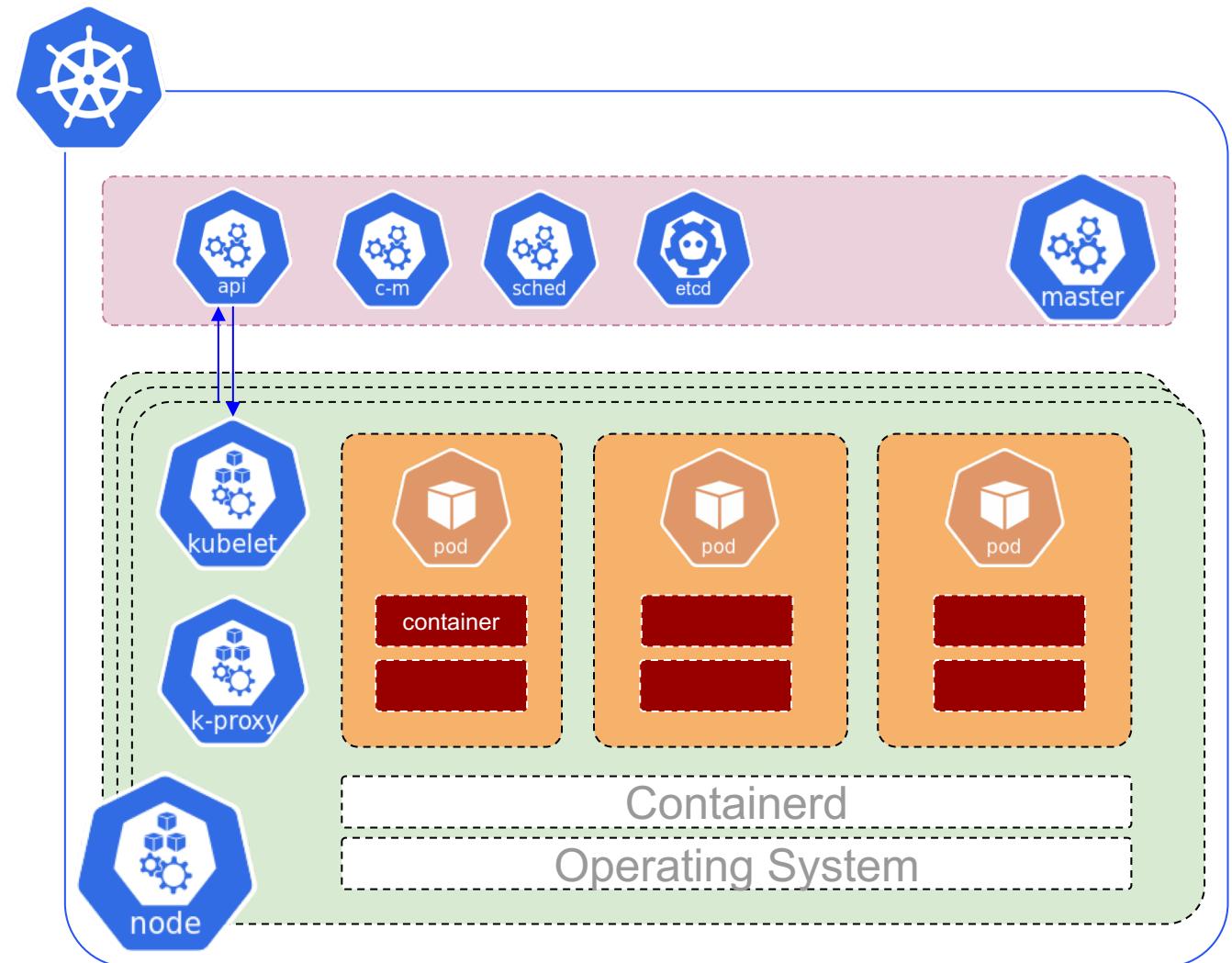
The Kubernetes network proxy runs on each node. This reflects services as defined in the Kubernetes API on each node.



POD

Collection of containers that can run on a host.

This resource is created by clients and scheduled onto hosts.

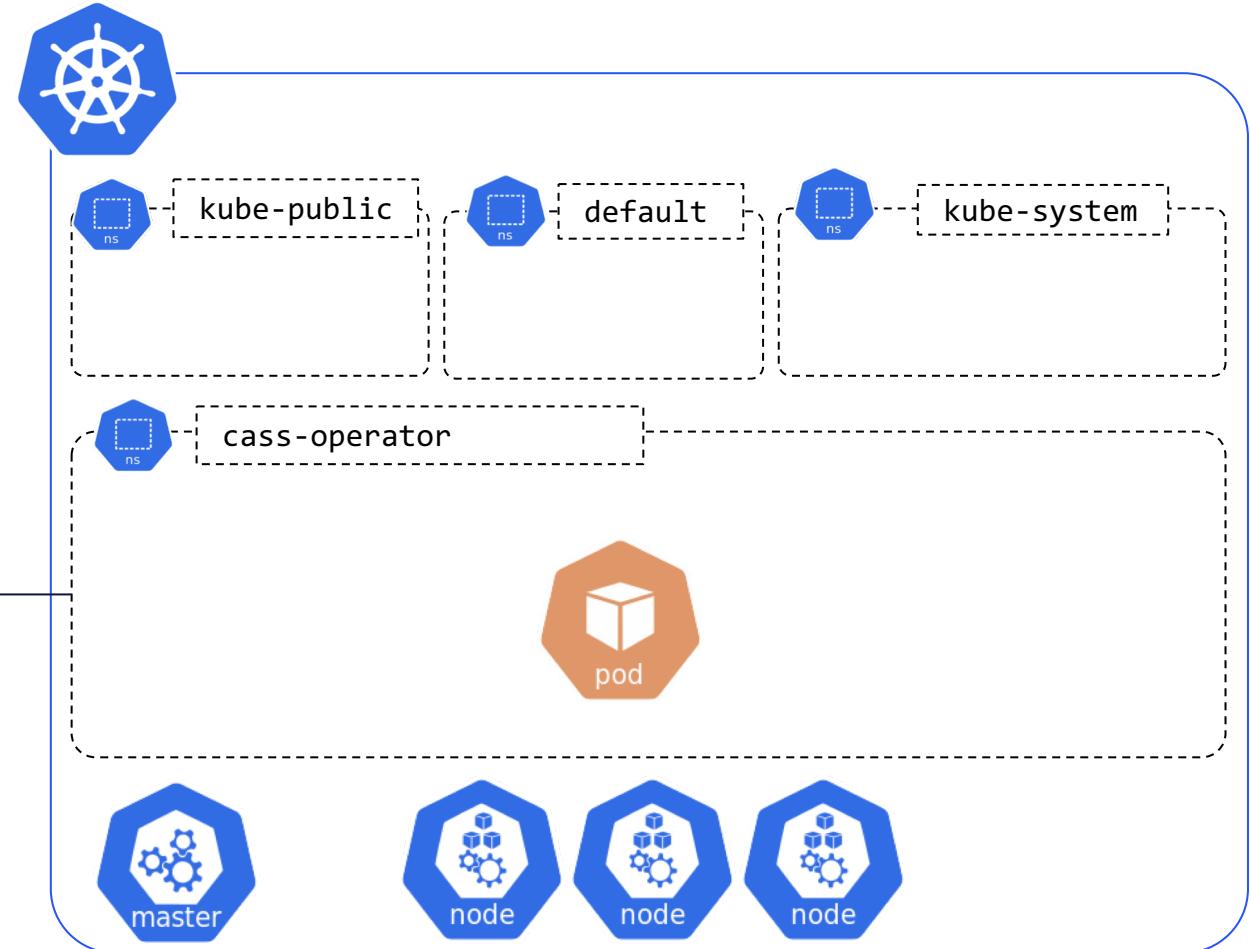


Kubernetes Namespace



Namespace: Namespace provides a scope for Names. Use of multiple namespaces is optional.

We create resources in
namespaces span across node.



K8s Primitives : Storage



PersistentVolume: is a storage resource provisioned by an administrator.

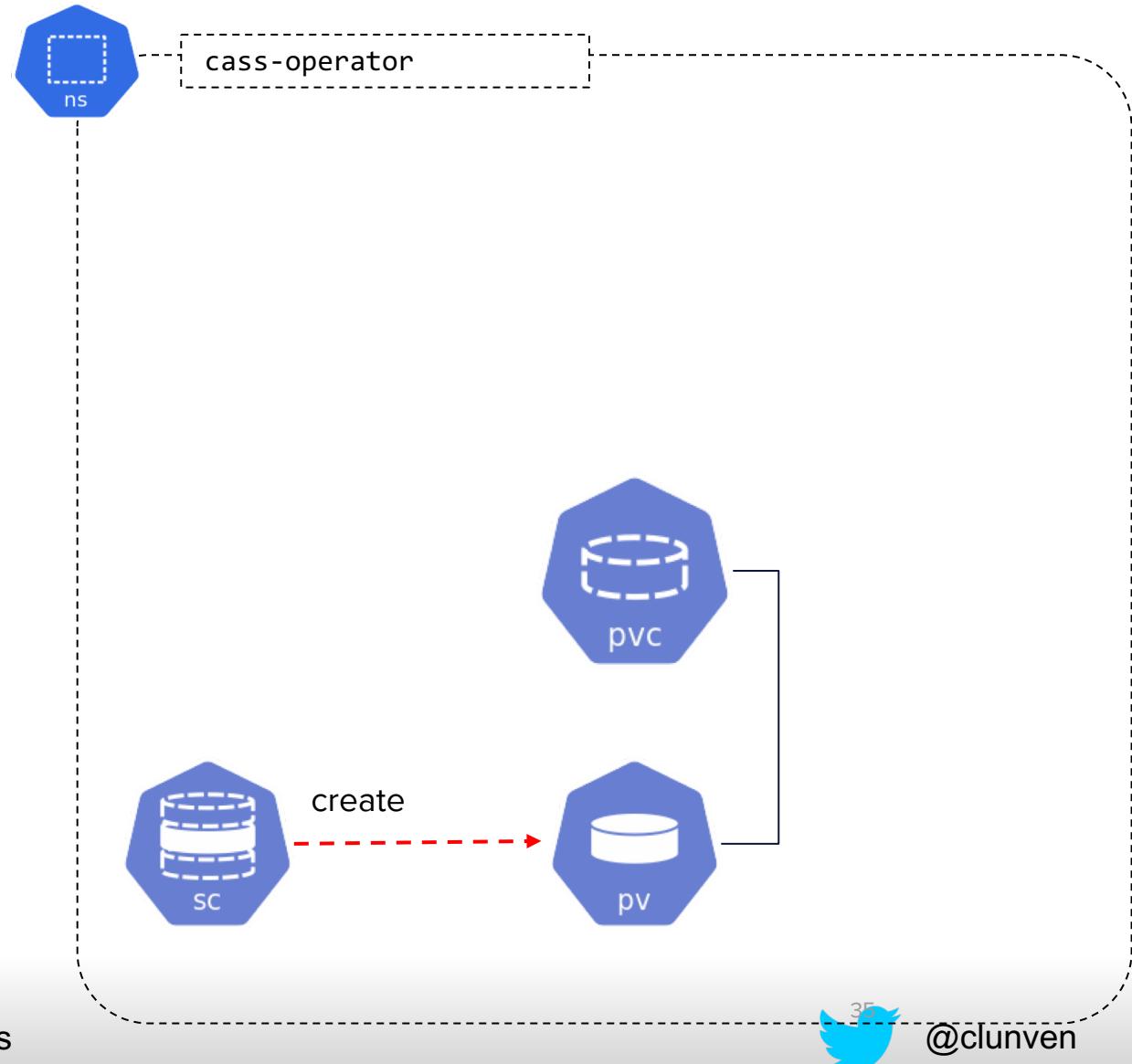


PersistentVolumeClaim:

PersistentVolumeClaim is a user's request for and claim to a persistent volume.



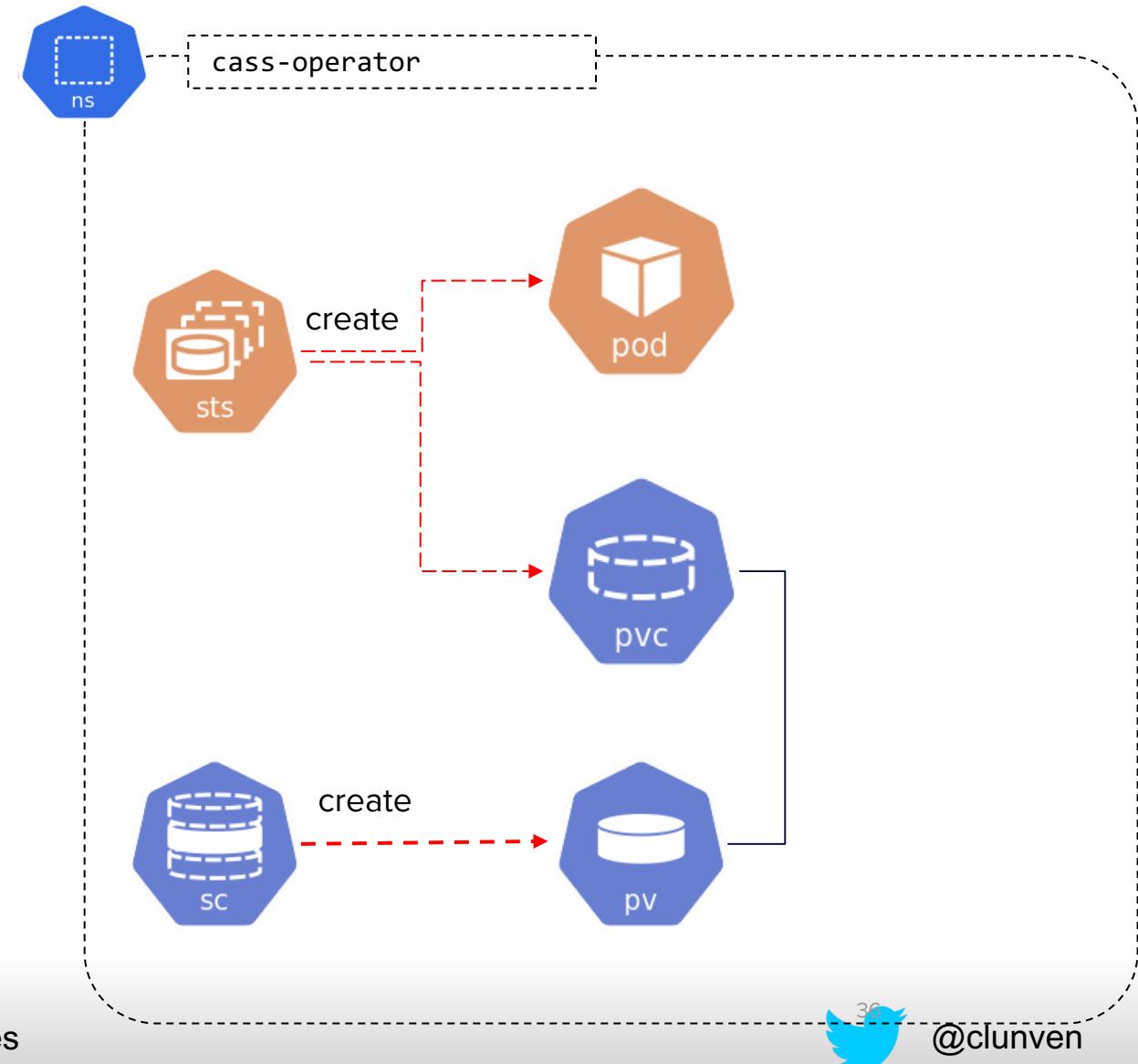
StorageClass: StorageClass describes the parameters for a class of storage for which *PersistentVolumes* can be dynamically provisioned.



K8s Primitives : StatefulSet



StatefulSet: StatefulSet represents a set of pods with consistent identities. Identities are defined as: network, storage.



K8s Primitives : Custom Resources



Custom Resource Definition :
Extensions of the Kubernetes API.
Customization making K8s more modular

- **Spec** declares the desired state of a resource
 - **Configuration settings** provided by the user
 - **Default values** expanded by the system
 - **Other properties** initialized by other internal components after resource creation.
- **Status** : describes the object's current, observed state.
 - Kubernetes API server provides a REST API to clients. A Kubernetes object or resource is a REST resource.
 - The status of a Kubernetes resource is typically implemented as a **REST subresource** that can only be modified by internal, system components

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: crontabs.stable.example.com
spec:
  # group name to use for REST API: /apis/<group>/<version>
  group: stable.example.com
  # list of versions supported by this CustomResourceDefinition
  versions:
    - name: v1
      # Each version can be enabled/disabled by Served flag.
      served: true
      # One and only one version must be marked as the storage version.
      storage: true
      schema:
        openAPIV3Schema:
          type: object
          properties:
            spec:
              type: object
              properties:
                cronSpec:
                  type: string
                image:
                  type: string
                replicas:
                  type: integer
                # either Namespaced or Cluster
                scope: Namespaced
                names:
                  [...]
```

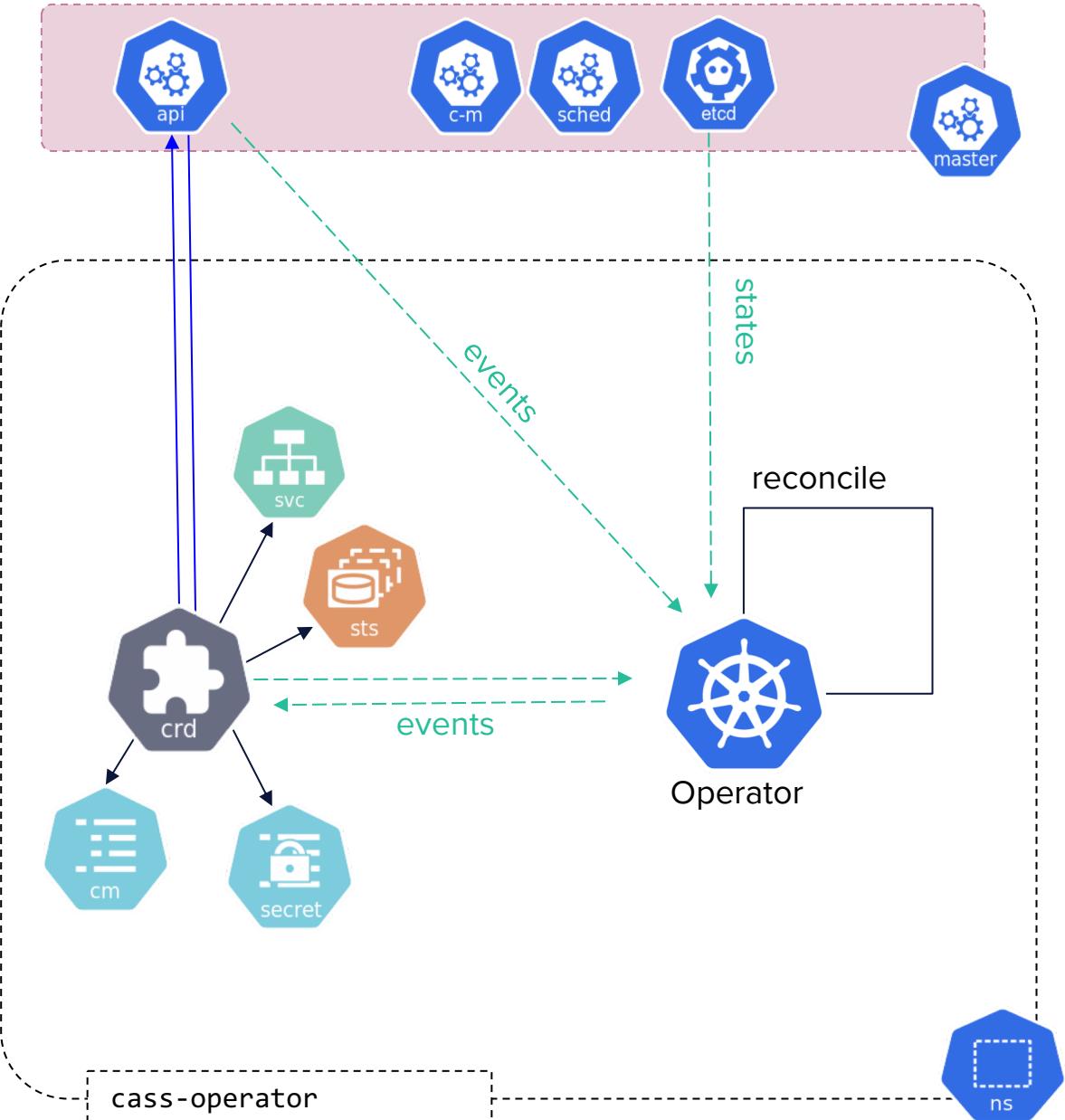


K8s Primitives : Operator

Building an application and driving an application on top of Kubernetes, behind Kubernetes APIs

A Kubernetes Operator helps extend the types of applications that can run on Kubernetes by allowing developers to provide additional knowledge to applications that need to maintain state.” —[Jonathan S. Katz](#)

- **Reconcile** CRD instances which states defined within the “**spec**” attribute.
- **Listen events** and **status evolution** to react accordingly.



Apache Cassandra™ with Kubernetes

1

2

3

4

Reminders on Apache Cassandra™

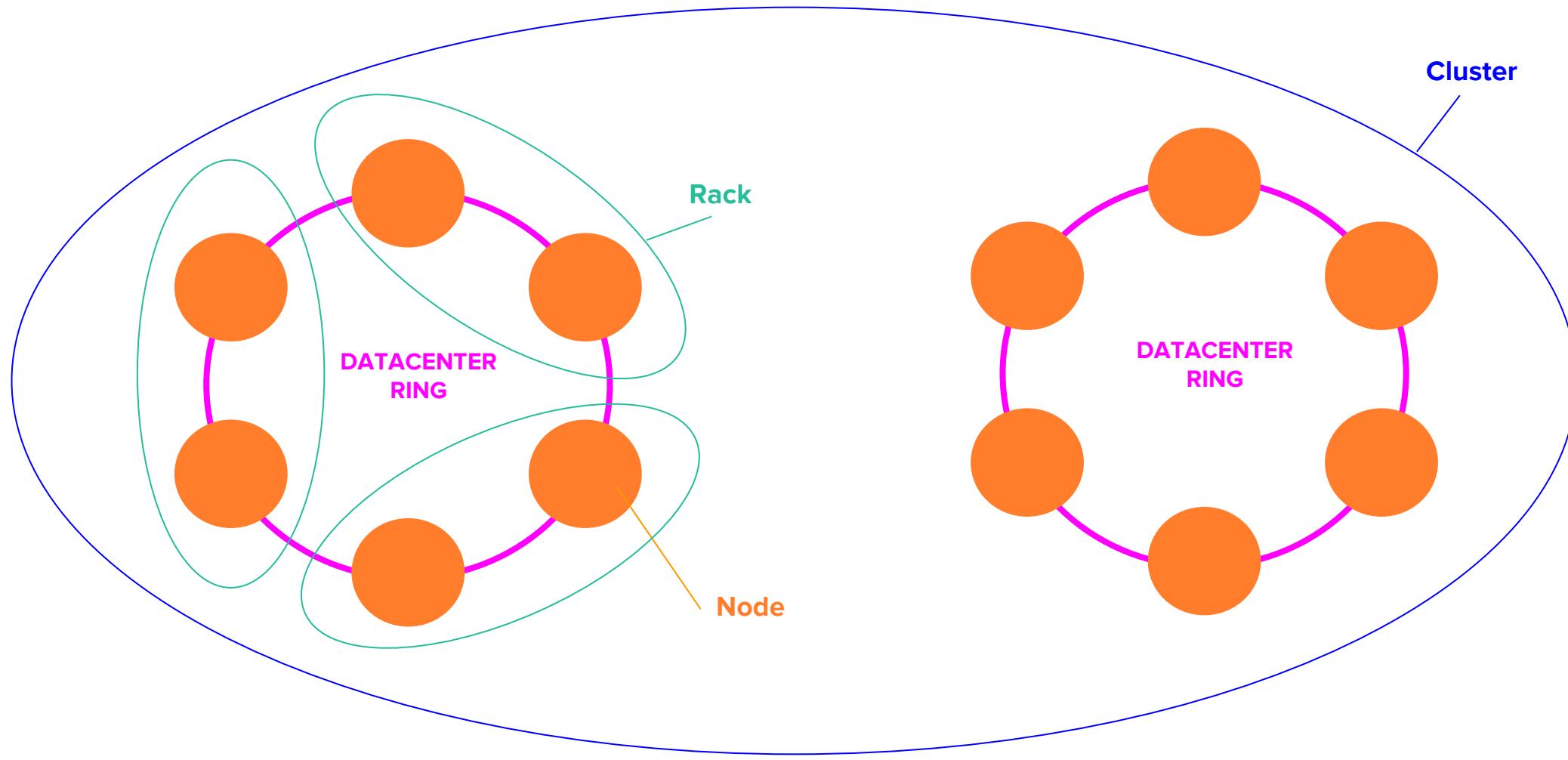
Cassandra and Containers

Kubernetes and Stateful Applications

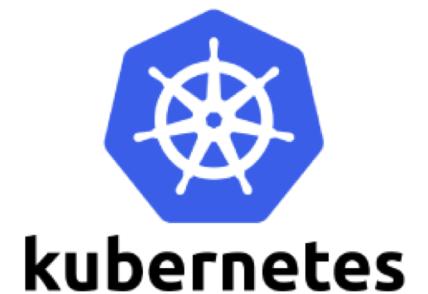
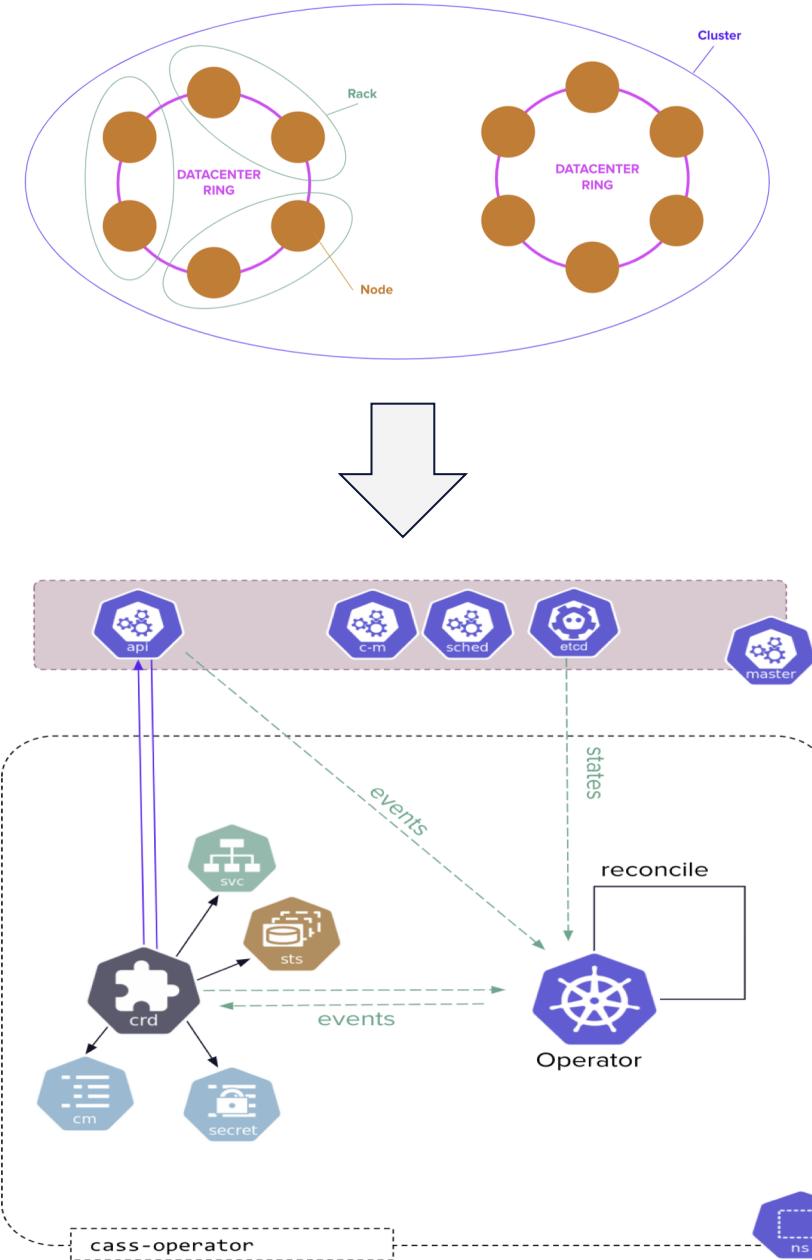
Cass Operator



Apache Cassandra™ Vocabulary



YOUR MISSION SHOULD YOU CHOOSE TO ACCEPT IT

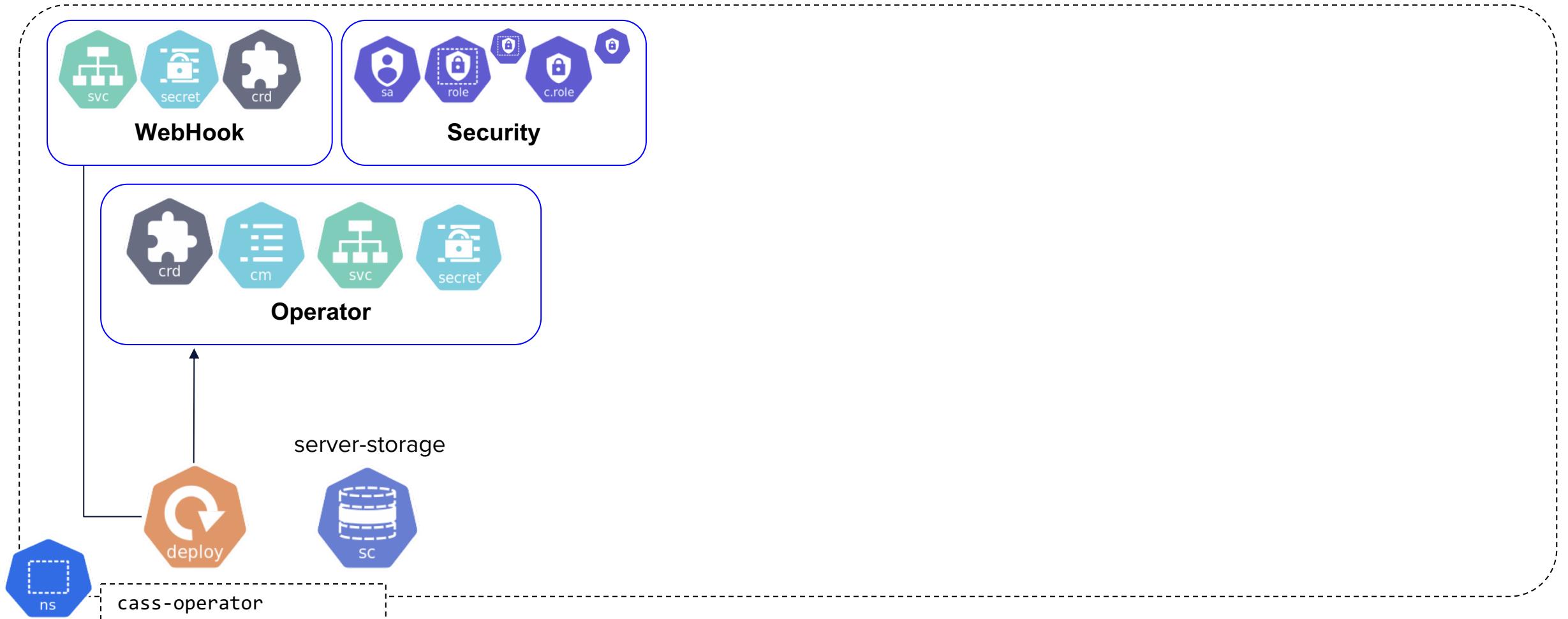


Cass Operator : Features

- Proper token **ring initialization**, with only one node bootstrapping at a time
- **Seed node management** -
 - one per rack, or three per datacenter, whichever is more
- Server configuration integrated into the **CassandraDatacenter CRD**
 - Rolling reboot nodes by changing the CRD
 - Store data in a rack-safe way - one replica per cloud AZ
 - Scale up racks evenly with new nodes
 - Replace dead/unrecoverable nodes
- Multi DC clusters (limited to one Kubernetes namespace)



Installing the Cass Operator Manifest



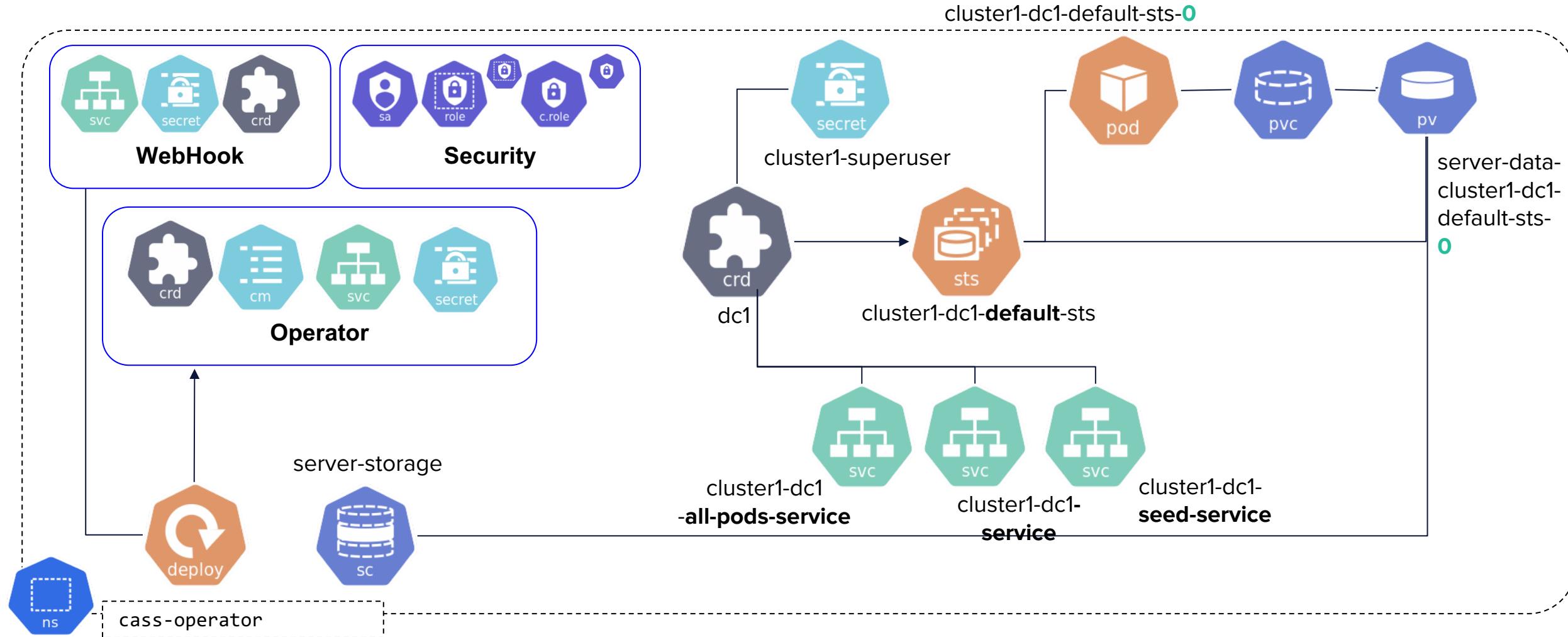
CRD CassandraDataCenter

```
1 # Sized to work on 3 k8s workers nodes with 1 core / 4 GB RAM
2 # See neighboring example-cassdc-full.yaml for docs for each parameter
3 apiVersion: cassandra.datastax.com/v1beta1
4 kind: CassandraDatacenter
5 metadata:
6   name: dc1
7 spec:
8   clusterName: cluster1
9   serverType: cassandra
10  serverVersion: "3.11.6"
11  managementApiAuth:
12    insecure: {}
13  size: 1
14  storageConfig:
15    cassandraDataVolumeClaimSpec:
16      storageClassName: server-storage
17      accessModes:
18        - ReadWriteOnce
19      resources:
20        requests:
21          storage: 5Gi
22 config:
23   cassandra-yaml:
24     authenticator: org.apache.cassandra.auth.PasswordAuthenticator
25     authorizer: org.apache.cassandra.auth.CassandraAuthorizer
26     role_manager: org.apache.cassandra.auth.CassandraRoleManager
27     jvm-options:
28       initial_heap_size: "800M"
29       max_heap_size: "800M"
```

```
apiVersion: cassandra.datastax.com/v1beta1
kind: CassandraDatacenter
metadata:
  name: multi-rack
spec:
  clusterName: multi-rack
  serverType: cassandra
  serverVersion: 3.11.6
  managementApiAuth:
    insecure: {}
  size: 9
  racks:
    - name: us-east1-b
      zone: us-east1-b
    - name: us-east1-c
      zone: us-east1-c
    - name: us-east1-d
      zone: us-east1-d
  storageConfig:
    cassandraDataVolumeClaimSpec:
      storageClassName: standard
      accessModes:
        - ReadWriteOnce
      resources:
        requests:
          storage: 5Gi
```



Creating DataCenter dc1



Our Pods



Cassandra Management API Service

<https://github.com/datastax/management-api-for-apache-cassandra>

Management API for Apache Cassandra 0.1 OAS3
<https://raw.githubusercontent.com/datastax/management-api-for-apache-cassandra/master/management-api-server/doc/openapi.json>

This is a Restful service for operating Apache Cassandra. You can find out more about the Management API on [Github](#)

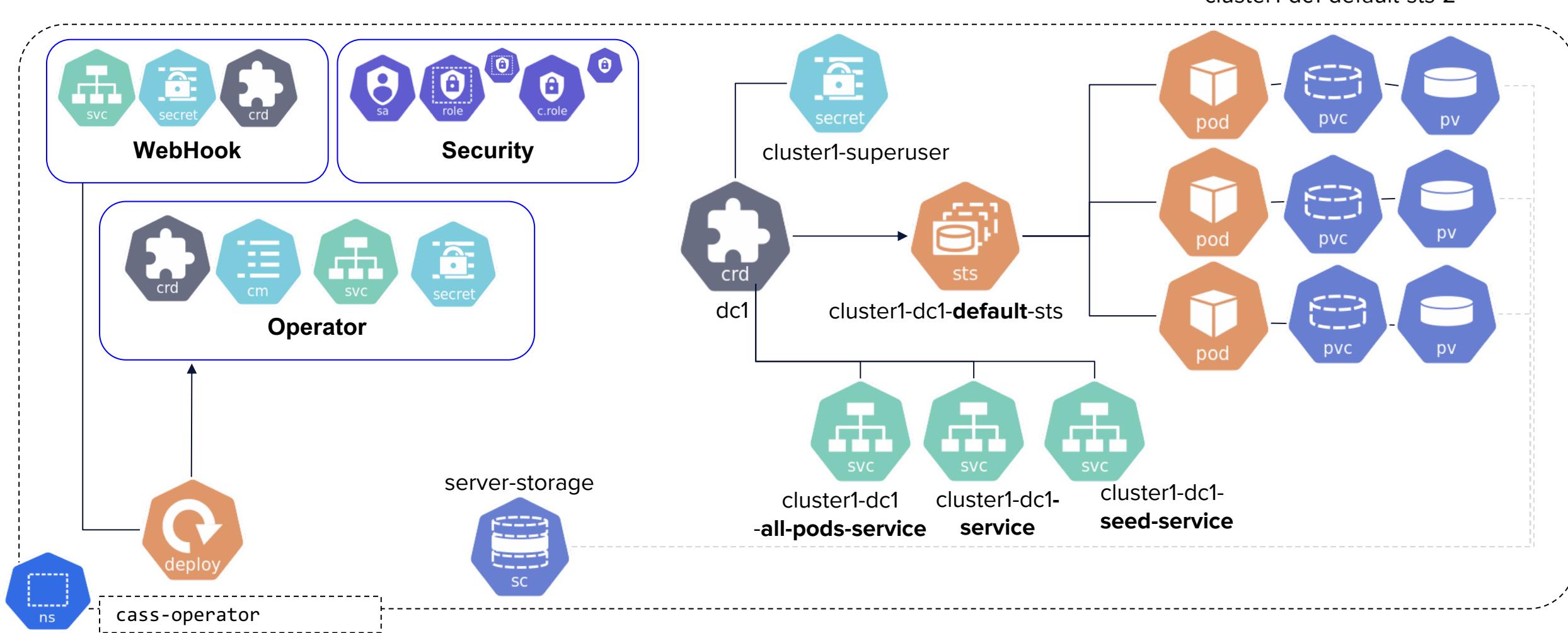
Apache 2.0

default

- POST** /api/v0/ops/auth/role Creates a new user role
- GET** /api/v0/probes/liveness Indicates whether this service is running
- GET** /api/v0/probes/readiness Indicates whether the Cassandra service is ready to service requests
- GET** /api/v0/probes/cluster Indicated whether the Cassandra cluster is able to achieve the specified consistency
- POST** /api/v0/ops/seeds/reload
- POST** /api/v0/ops/keyspace/refresh Load newly placed SSTables to the system without restart
- POST** /api/v0/ops/keyspace/cleanup Triggers the immediate cleanup of keys no longer belonging to a node. By default, clean all keyspaces
- POST** /api/v0/lifecycle/start
- POST** /api/v0/lifecycle/stop
- POST** /api/v0/lifecycle/configure
- GET** /api/v0/lifecycle/pid
- GET** /api/v0/metadata/versions/release Returns the Cassandra release version
- GET** /api/v0/metadata/endpoints Returns this nodes view of the endpoint states of nodes
- POST** /api/v0/ops/node/drain Drain the node (stop accepting writes and flush all tables)



Scale up DataCenter dc1



Cassandra Developers Workshops



DataStax Developers
2.44K subscribers

Example Data – Users organized by city

The screenshot shows a video conference with two participants. On the left, a participant named Aleks Volochnev is visible. On the right, a participant named Cedrick Lunven is visible. They are looking at a presentation slide. The slide has a title "1/3 - How much experience with Apache Cassandra?". Below the title is a bar chart showing the distribution of responses:

Experience Level	Count
None at all	124
Less than 1 year	95
1-3 years	49
3-5 years	8
More than 5 years	5

At the bottom of the slide, there is a call to action: "Join us to ask questions: bit.ly/cassandra-workshop".



EVERY WEDNESDAY
5PM-7PM BST



#CloudNativeLon | June 3rd | Deploy Cassandra on Kubernetes



@clunven

Cassandra on Kubernetes Workshop 6/11

- **REGISTER**

- <https://www.eventbrite.com/e/workshop-build-a-devops-pipeline-with-kubernetes-and-apache-cassandra-tm-tickets-106944768682?aff=erelexpmilt>

- **MATERIALS**

- <https://github.com/DataStax-Academy/kubernetes-workshop-online>



#CloudNativeLon | June 3rd | Deploy Cassandra on Kubernetes



@clunven

Cassandra The Definitive Guide

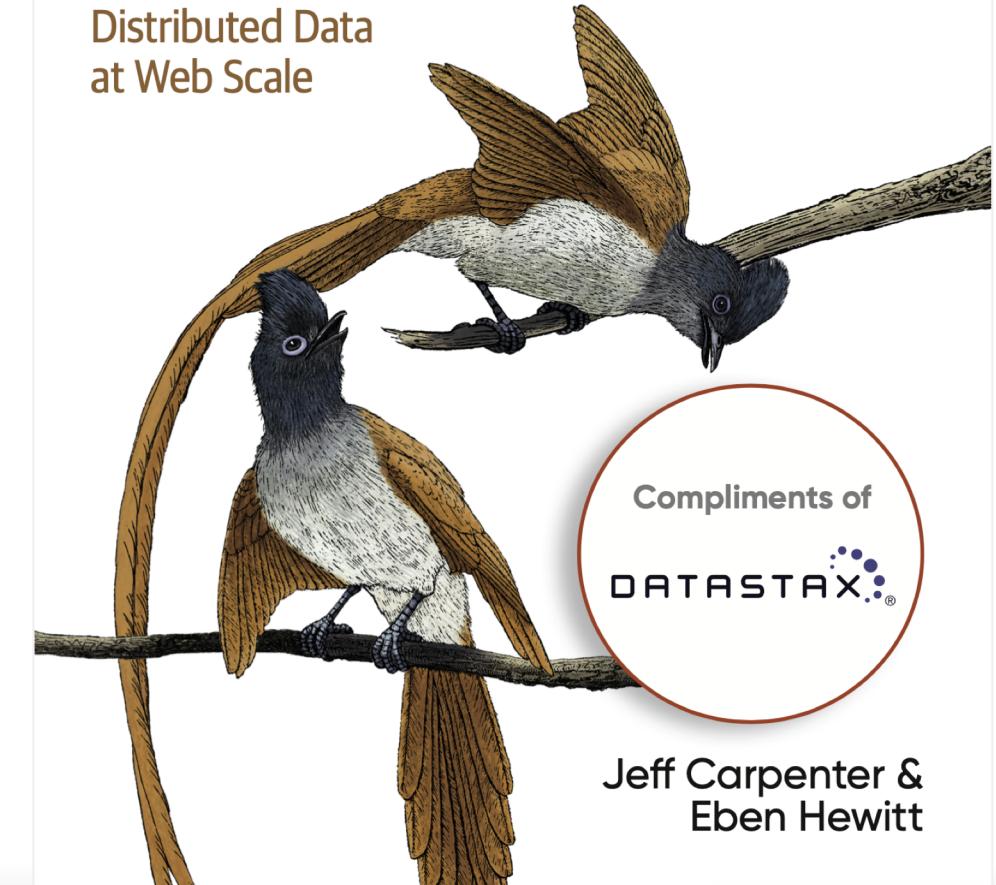
<https://www.datastax.com/resources/e-book/oreilly-cassandra-definitive-guide>



O'REILLY®

Cassandra The Definitive Guide

Distributed Data
at Web Scale



Jeff Carpenter &
Eben Hewitt

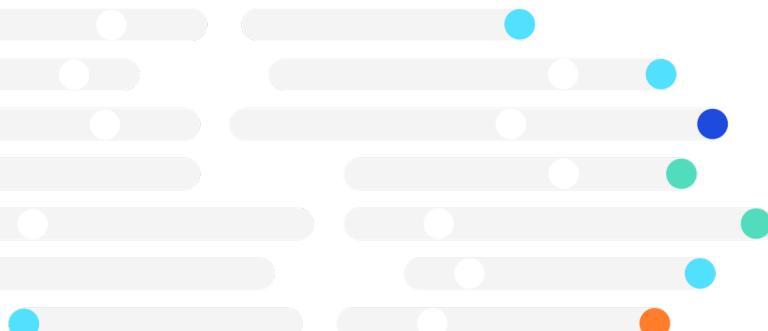


#CloudNativeLon | June 3rd | Deploy Cassandra on Kubernetes



@clunven

Third
Edition



Thank you

SURVEY