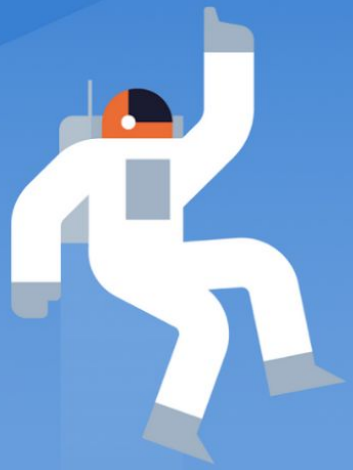
DataStax

# K8ssandra - Apache Cassandra™ meets Kubernetes!



LEVEL UP with the DataStax Developers

# Your presenters

**Aleks Volochnev**
Developer Advocate at DataStax

- Apache Cassandra™ expert
- Experienced developer and educator
- Certified cloud architect

@hadesarchitect

**Jeff Carpenter**
Developer Adoption at DataStax

- Apache Cassandra™ author (O'Reilly)
- Experienced architect and developer

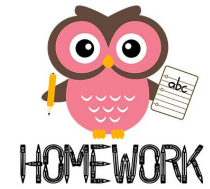
@jeffreyscarpenter

@jscarp

# K8ssandra - Apache Cassandra™ meets Kubernetes!

- Housekeeping and Setup
- Cassandra and Kubernetes Basics
- Introducing K8ssandra
- Monitoring Cassandra
- Working with Data
- Scaling Up and Down
- API Access with Stargate
- Running Cassandra Repairs
- Backup and Restore
- Wrapping up

# Kubernetes Workshop – Homework



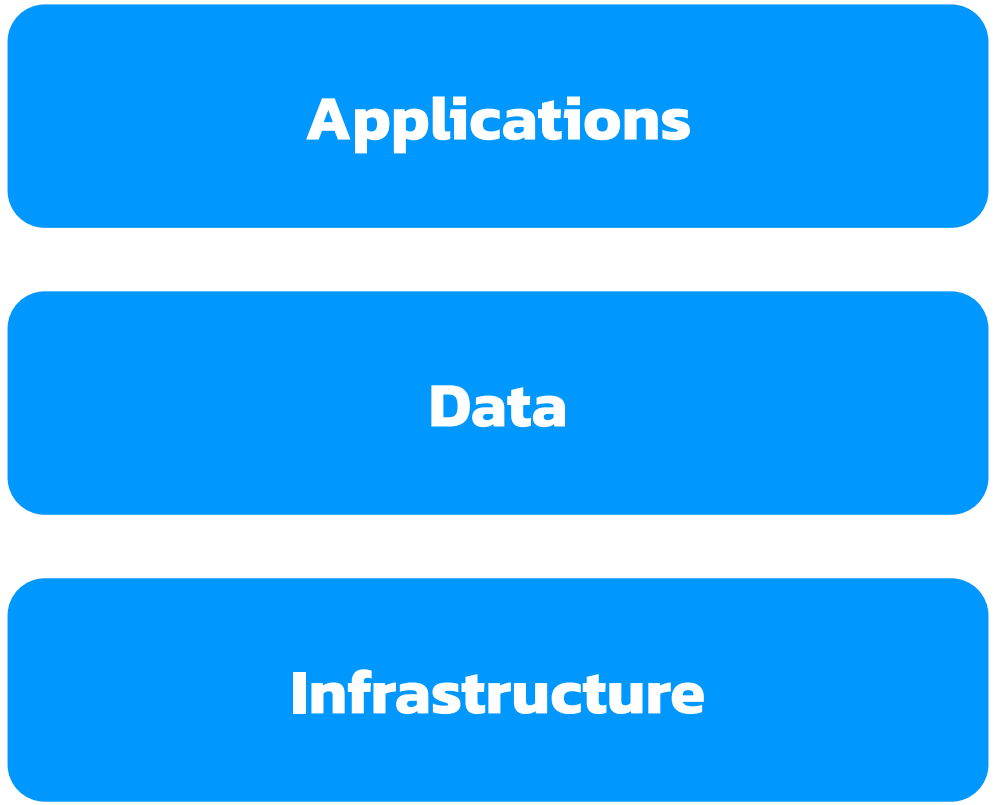

https://github.com/datastaxdevs/k8ssandra-workshop/wiki


menti.com
Mentimeter
What is the most desirable feature of your ideal workplace?
Flexible time
Social factor
Big paycheck
Available on the iPhone
App Store
GET IT ON
Google play

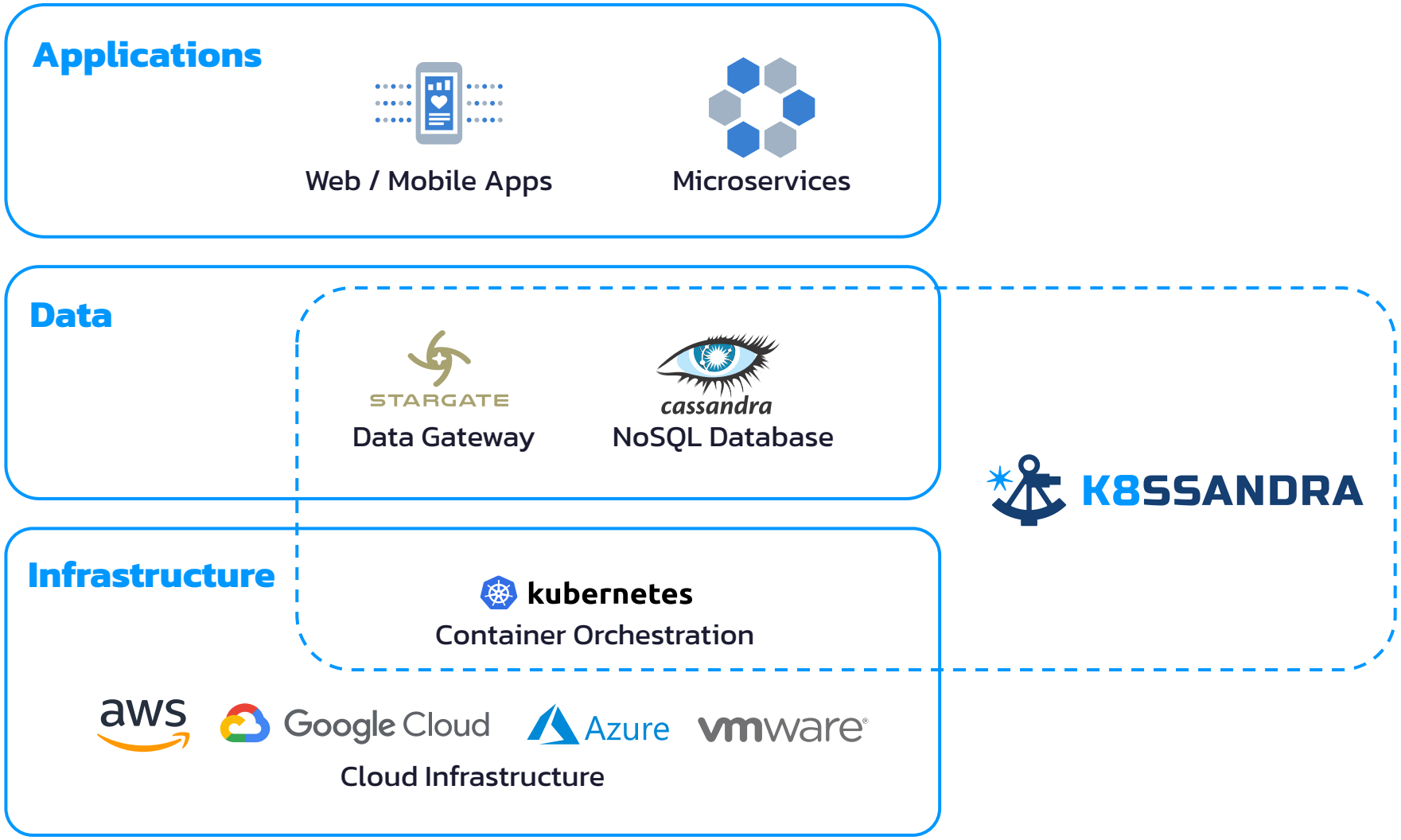
# K8ssandra - Apache Cassandra™ meets Kubernetes!

- Housekeeping and Setup
- Cassandra and Kubernetes Basics
- Introducing K8ssandra
- Monitoring Cassandra
- Working with Data
- Scaling Up and Down
- API Access with Stargate
- Running Cassandra Repairs
- Wrapping up

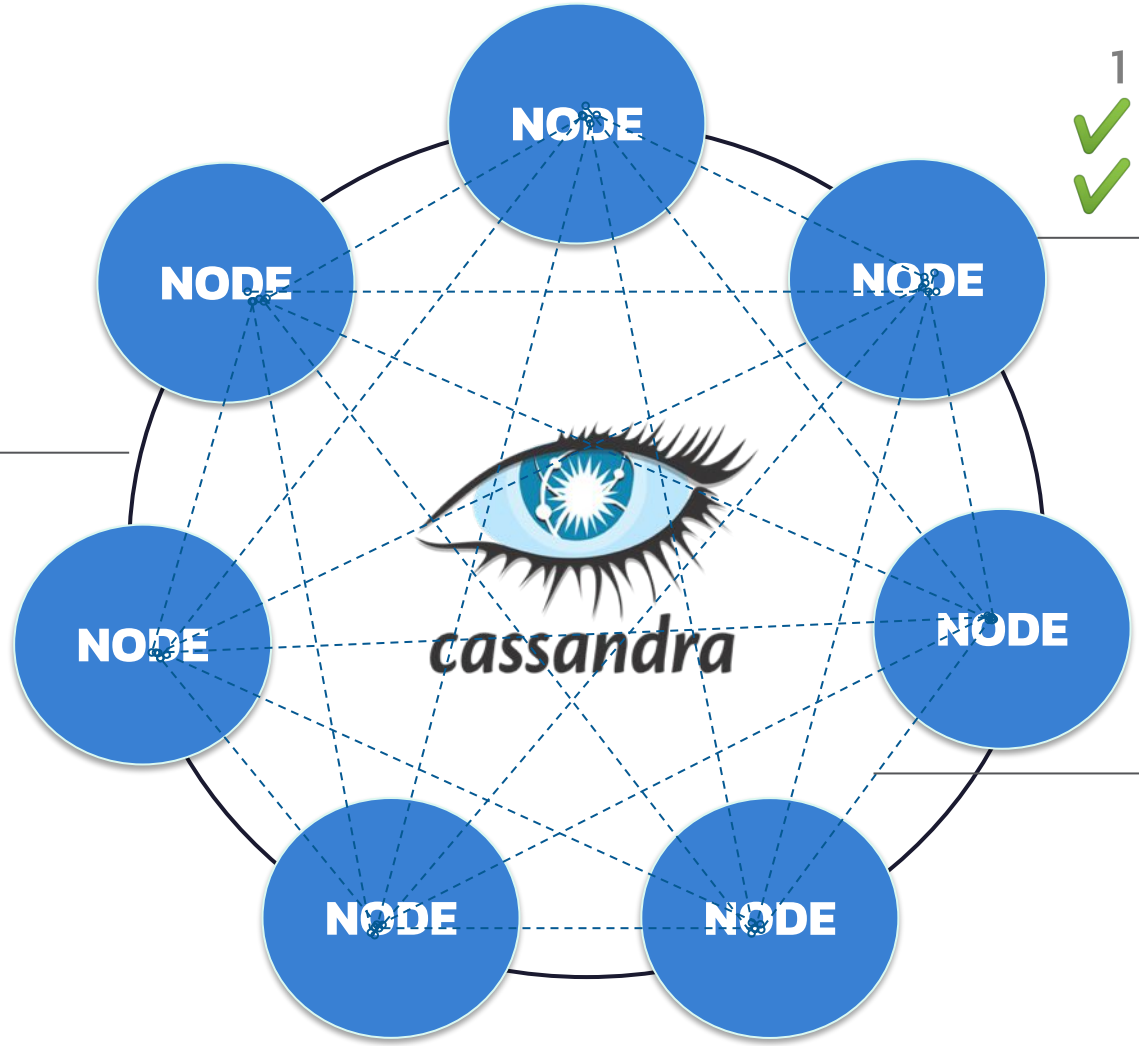
# Cloud Application Architecture – 50,000 ft



# Highly Scalable Cloud Application Architecture



# Apache Cassandra™ = NoSQL Distributed Database




1 Installation = 1 NODE

✔ Capacity = ~ 2–4TB

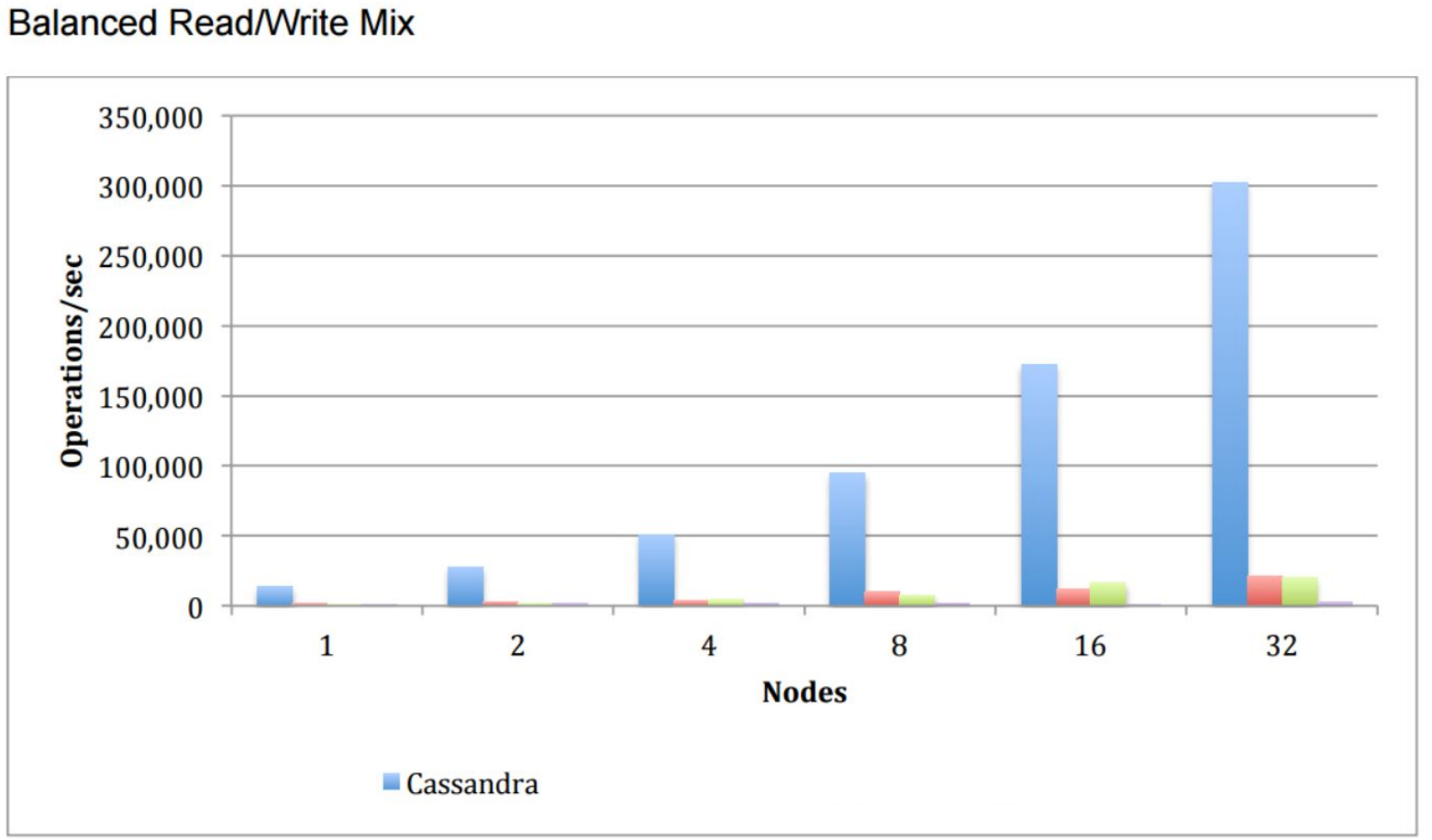
✔ Throughput = LOTS Tx/sec/core

DataCenter | Ring

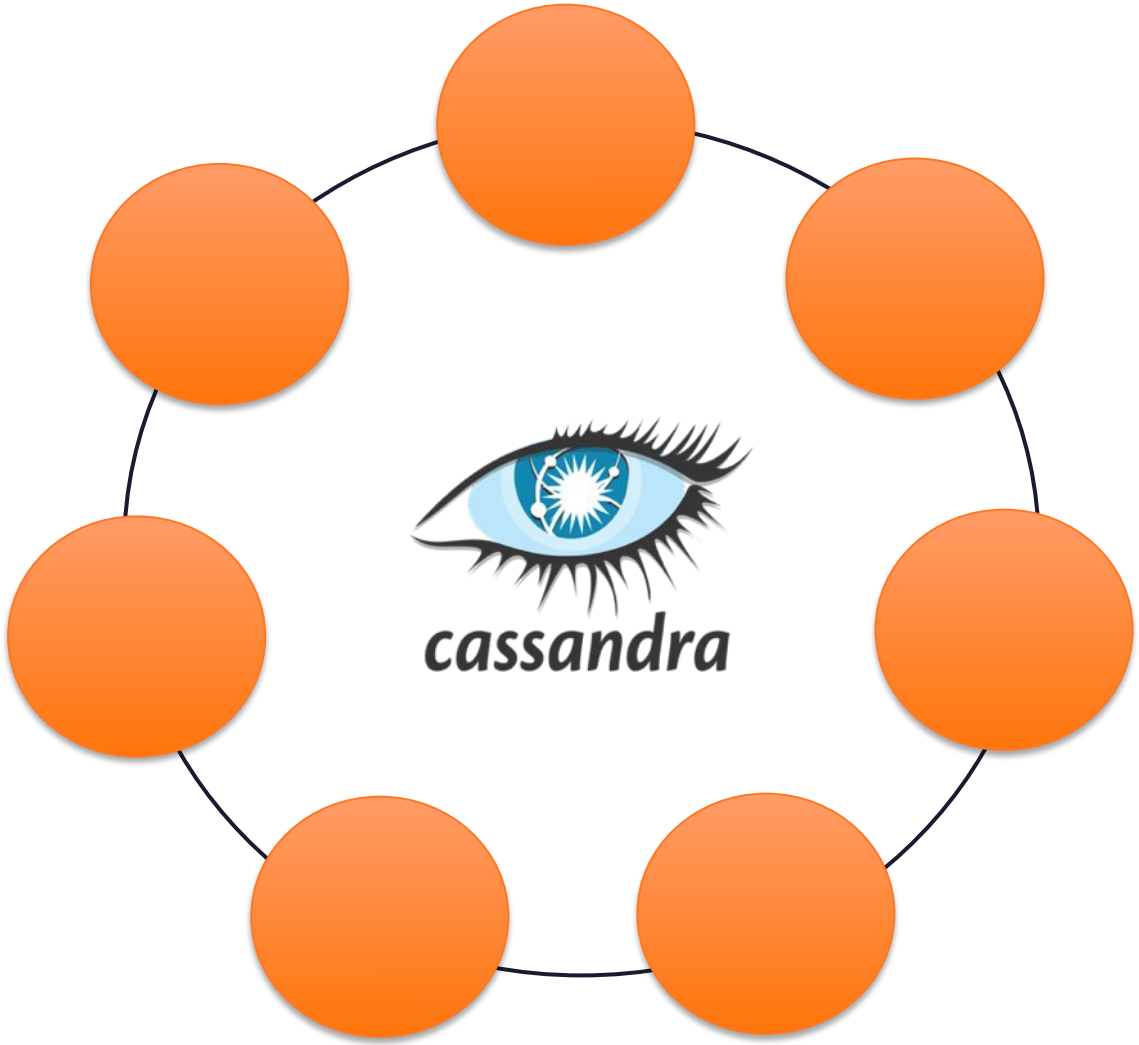
Communication:

✔ Gossiping

# Apache Cassandra™ Scales linearly



# Data is distributed

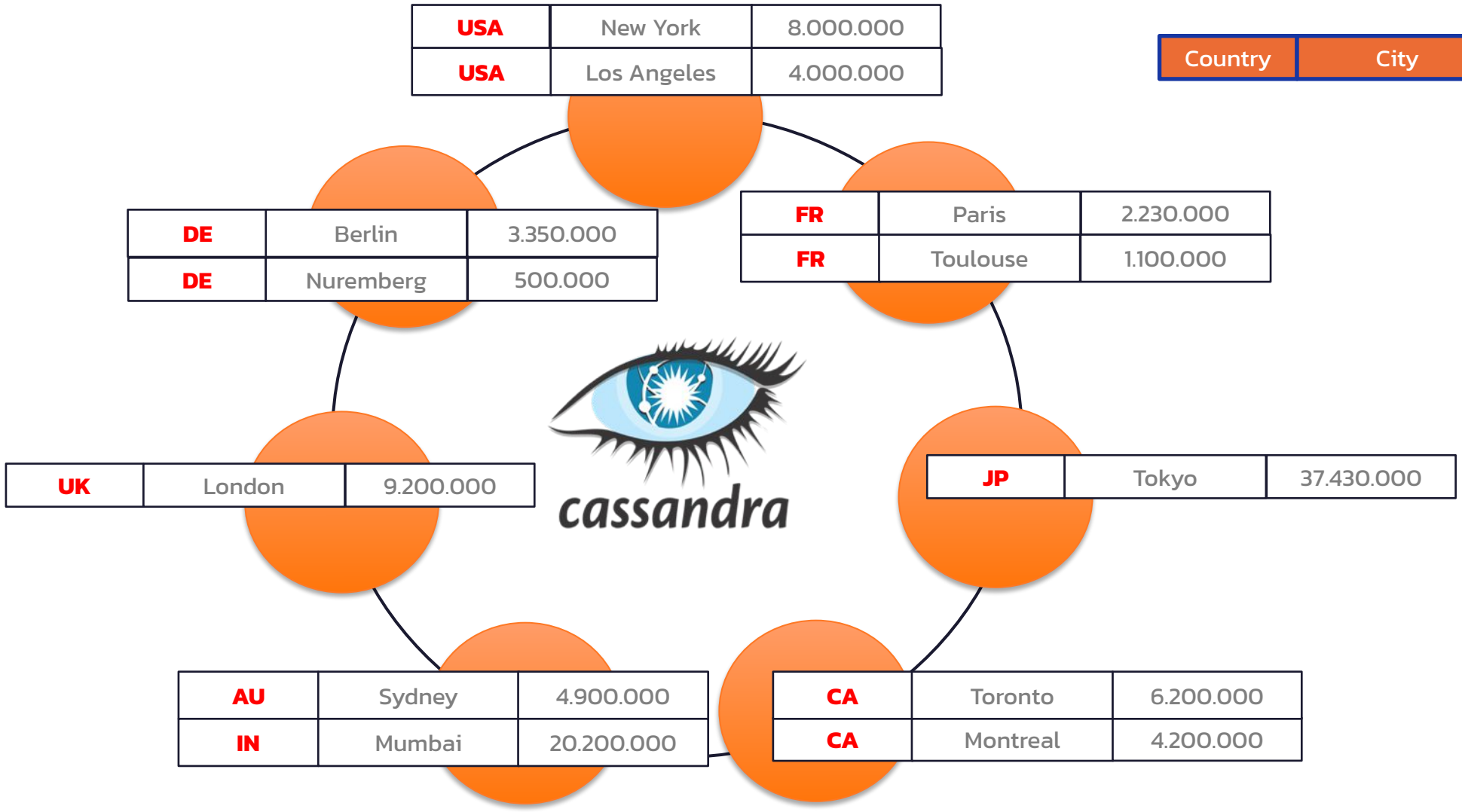



| Country | City | Population |
|---|---|---|
| USA | New York | 8.000.000 |
| USA | Los Angeles | 4.000.000 |
| FR | Paris | 2.230.000 |
| DE | Berlin | 3.350.000 |
| UK | London | 9.200.000 |
| AU | Sydney | 4.900.000 |
| DE | Nuremberg | 500.000 |
| CA | Toronto | 6.200.000 |
| CA | Montreal | 4.200.000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

*Partition Key*

# Data is Distributed

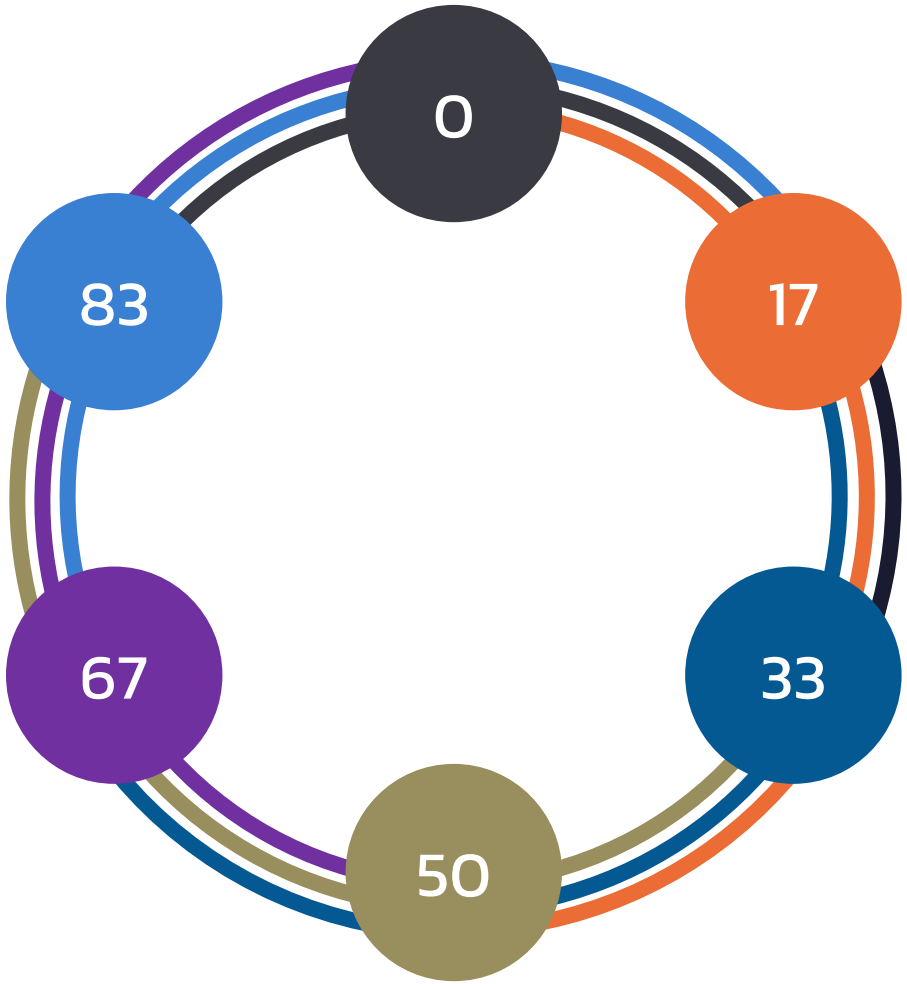
| Country | City | Population |
|---|---|---|



# Data is Replicated

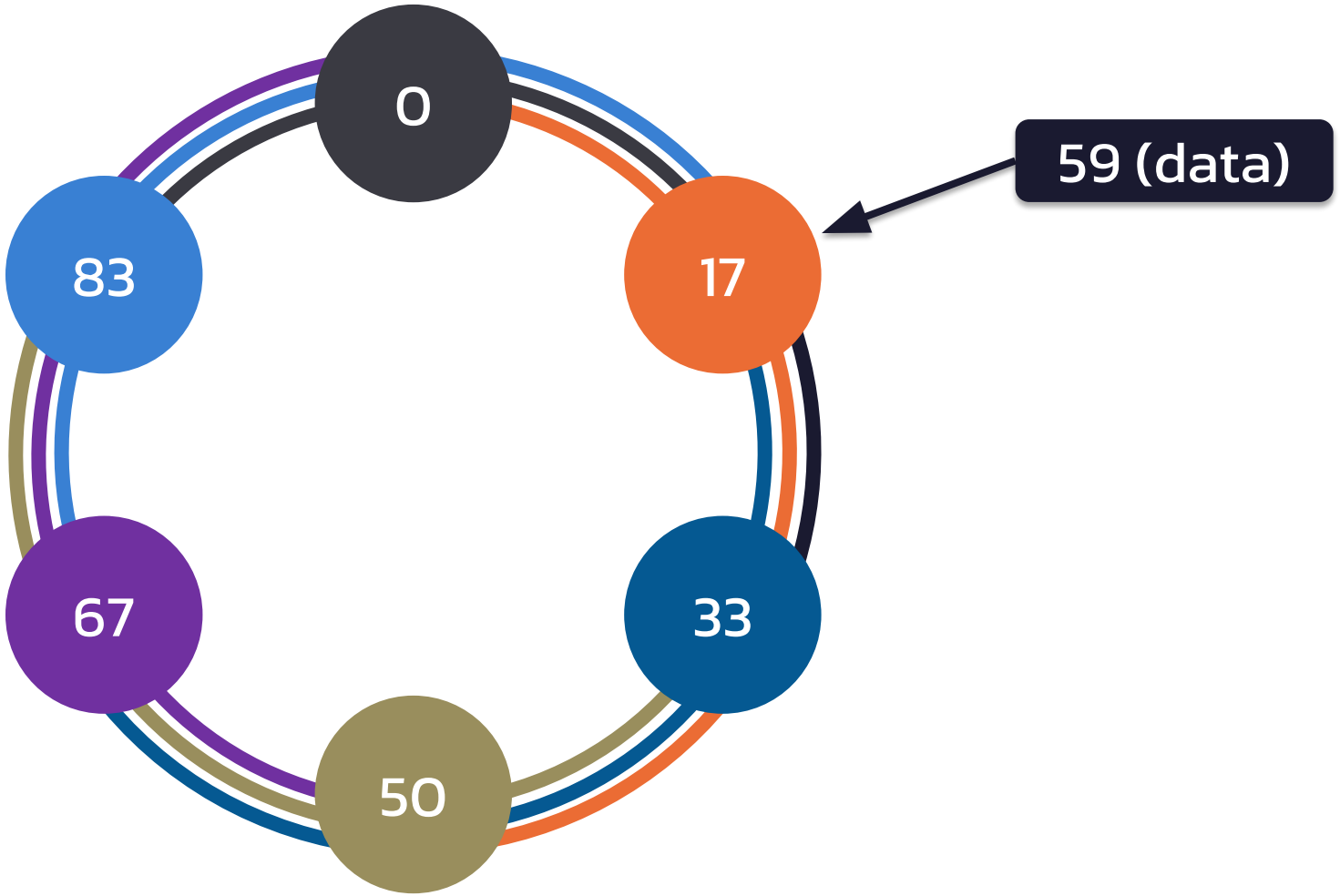
RF = 3

Replication Factor 3 means that every row is stored on 3 different nodes



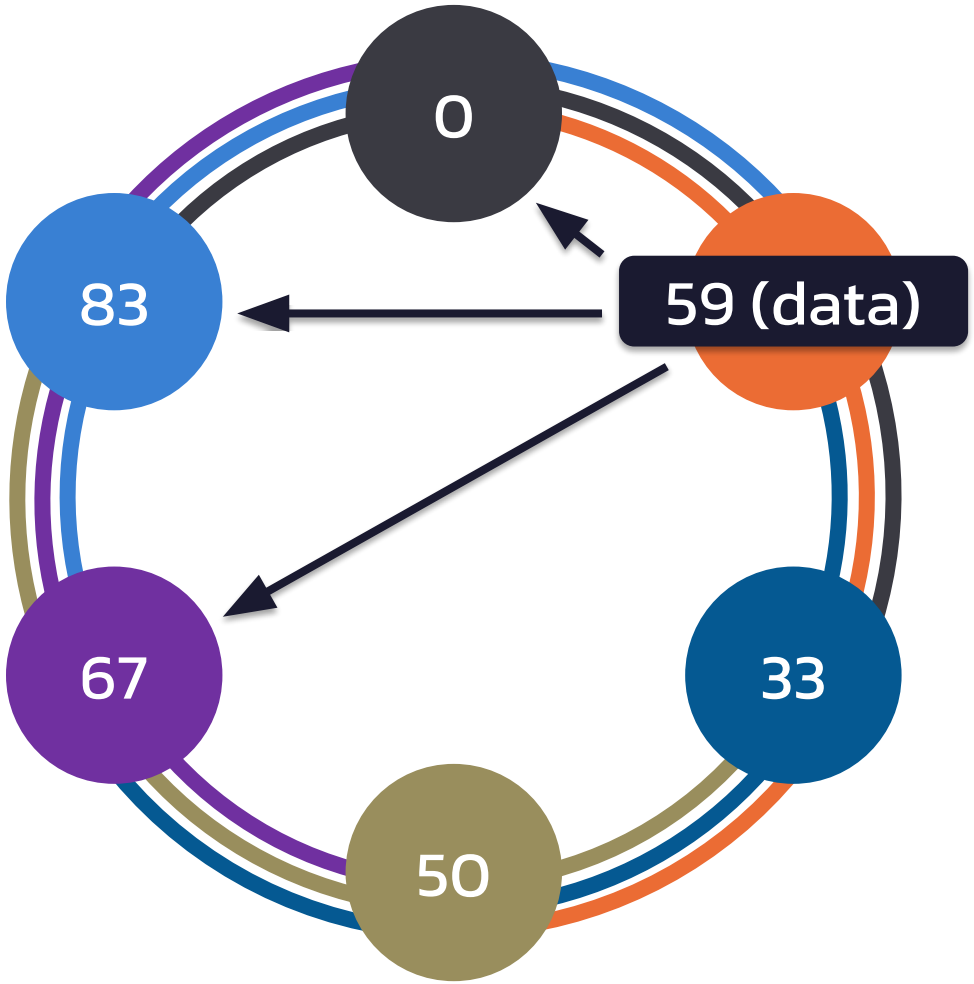
# Replication within the Ring

RF = 3



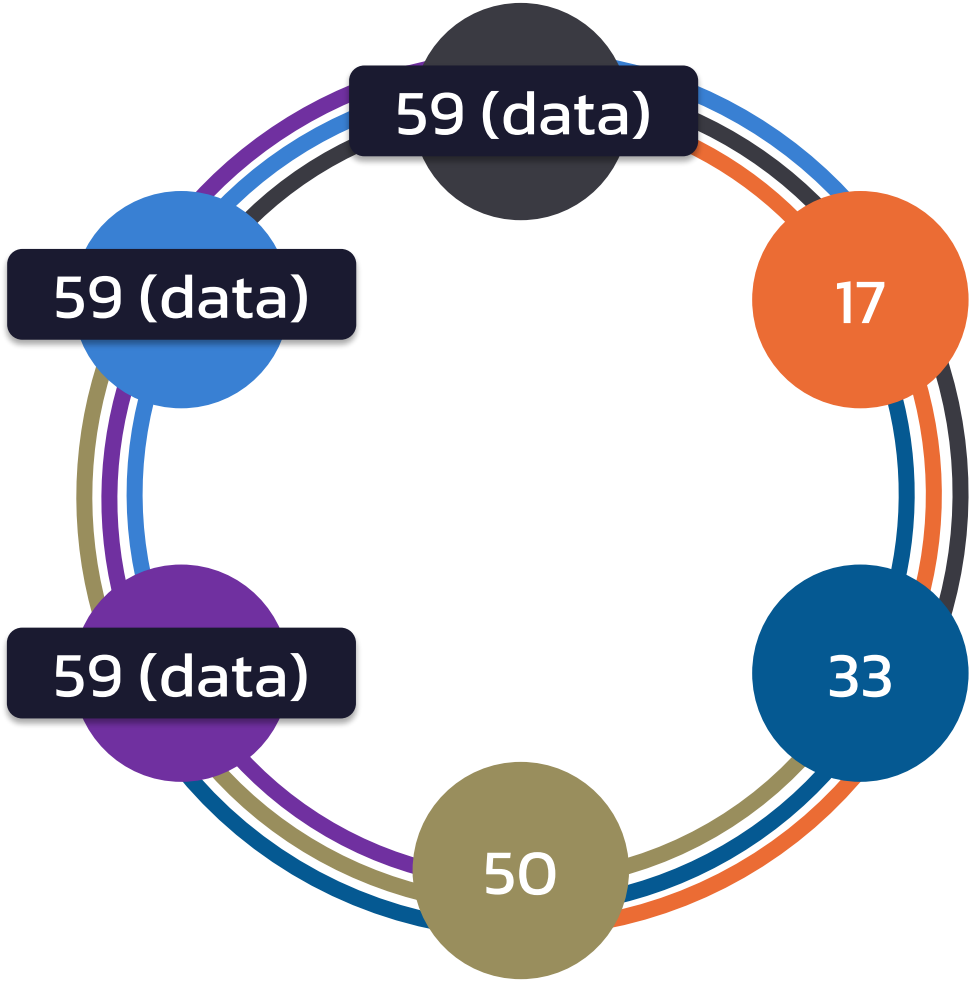
# Replication within the Ring

RF = 3



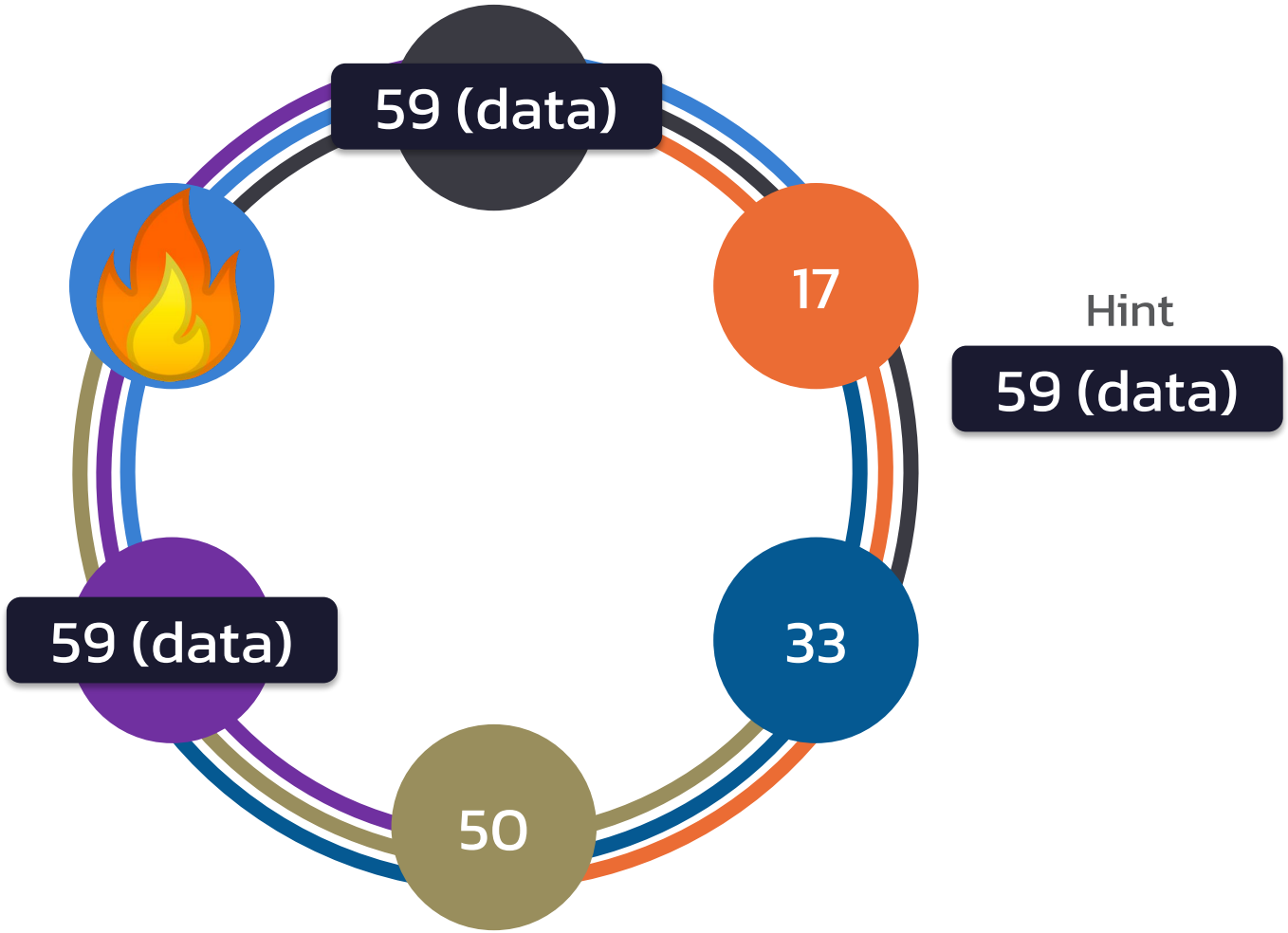
# Replication within the Ring

RF = 3



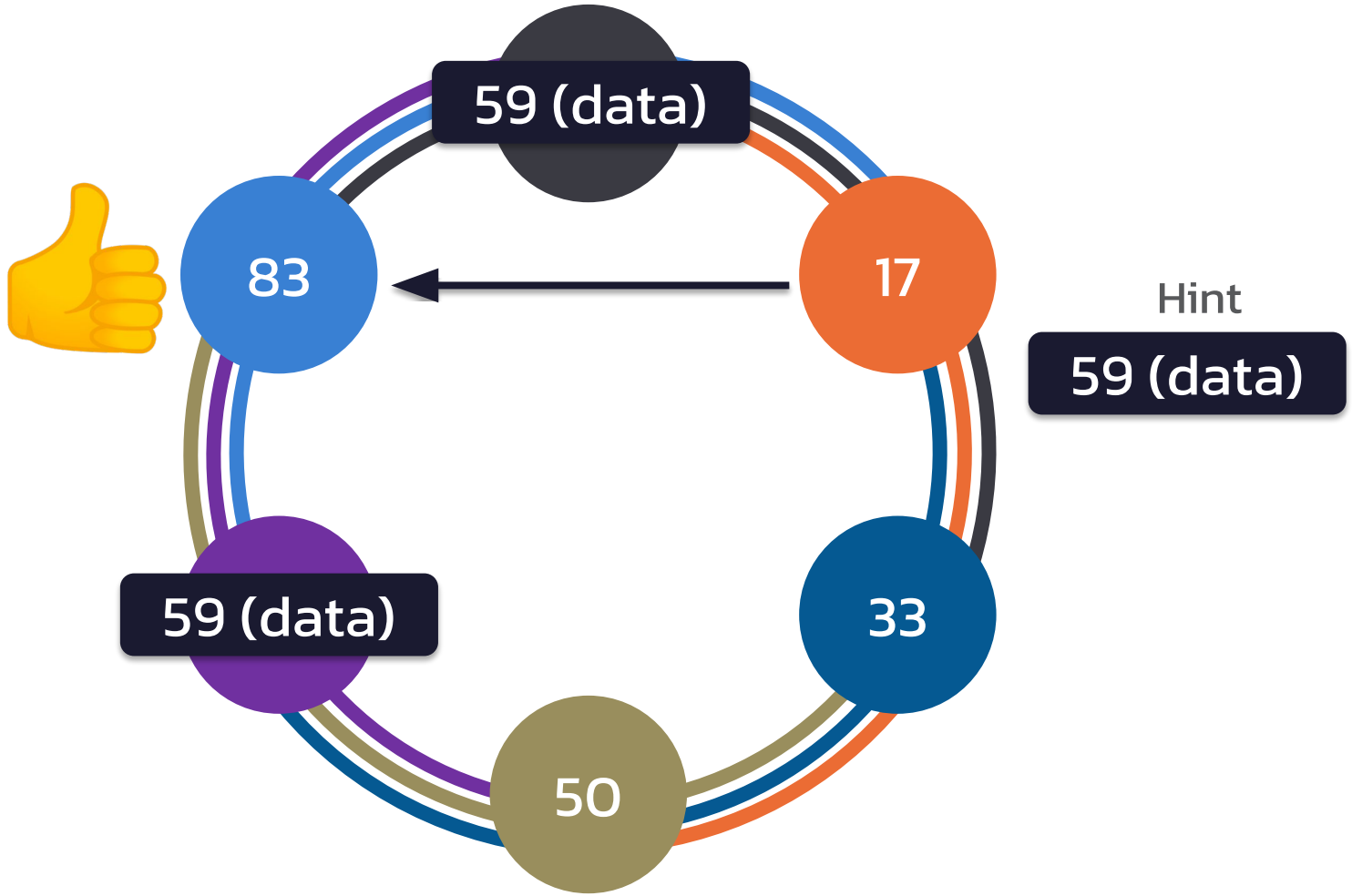
# Node Failure

RF = 3

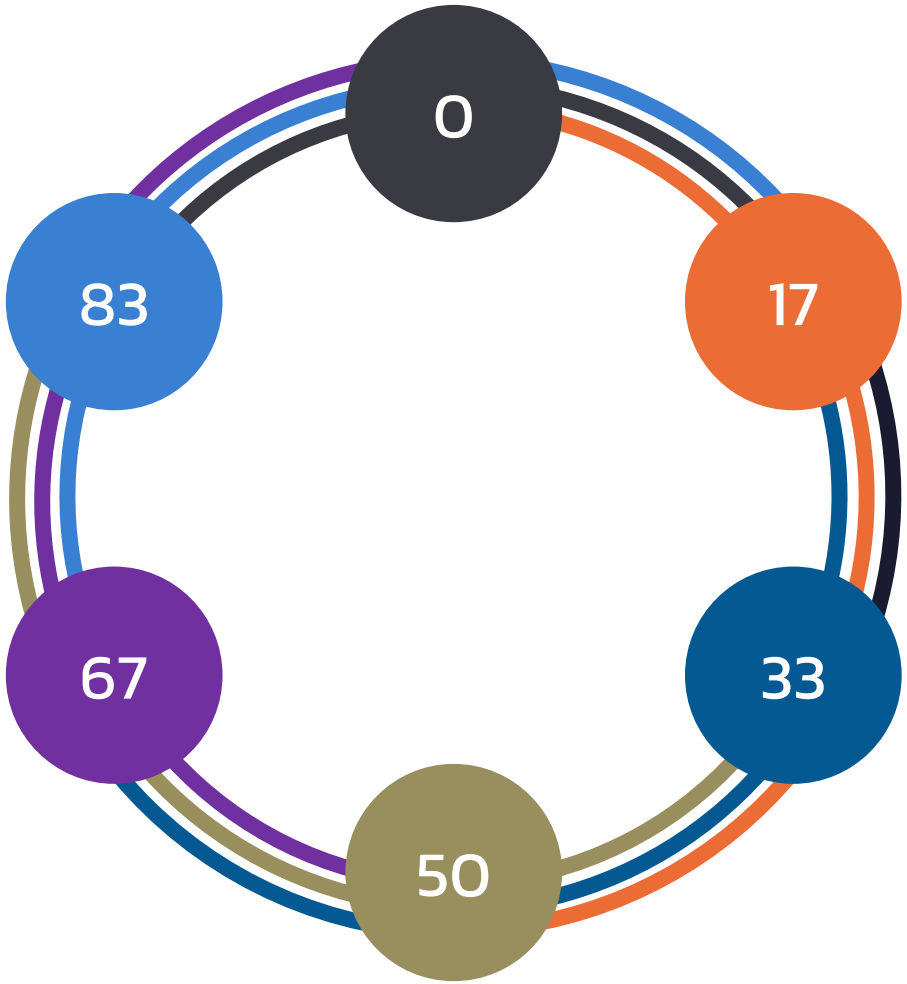


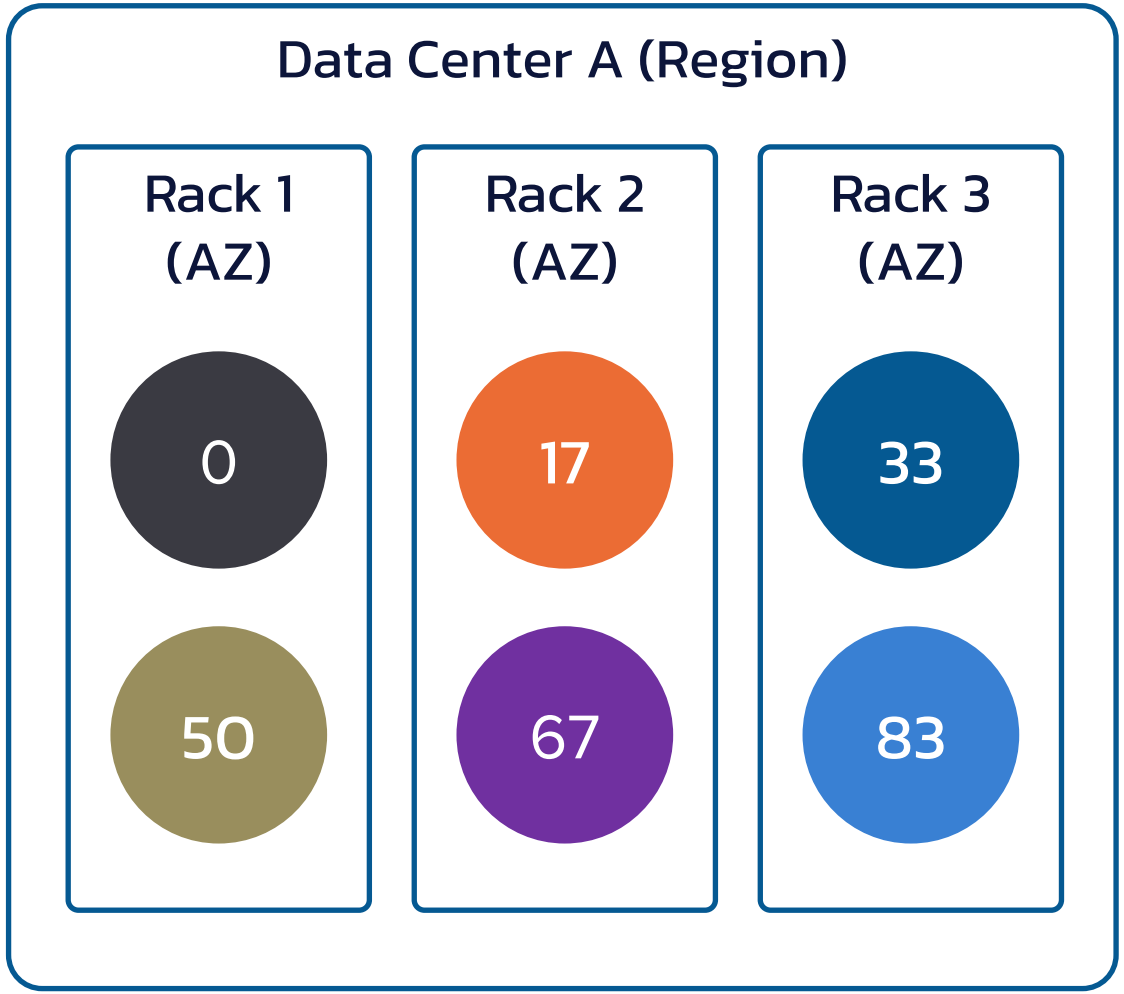
# Node Failure Recovered

RF = 3



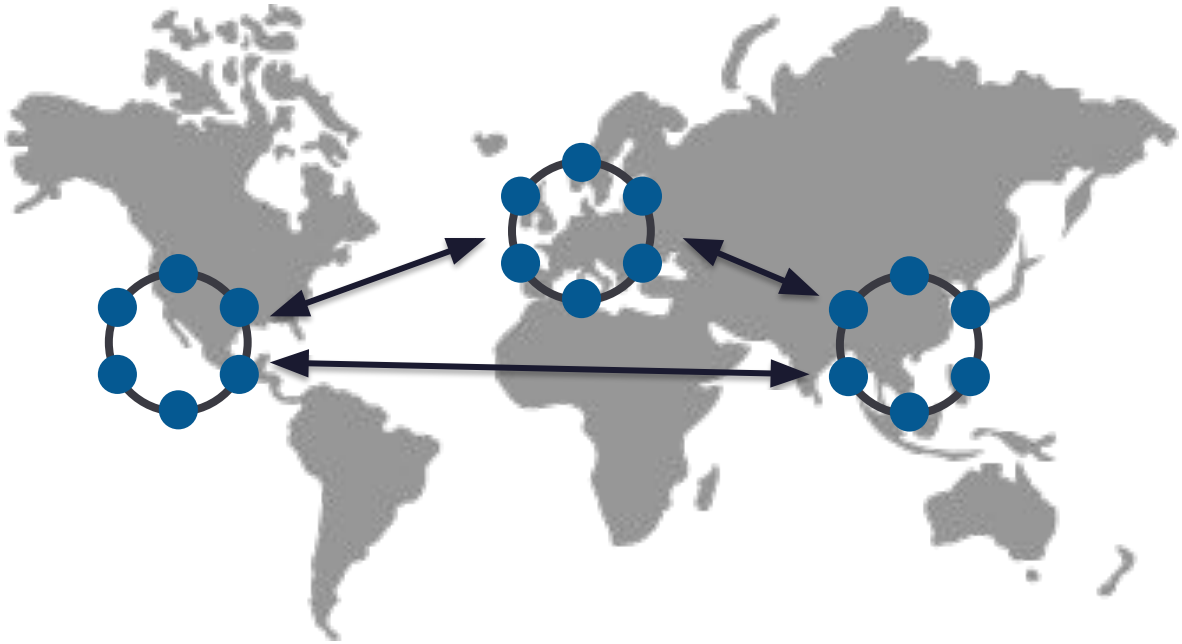
# Topology View – Data Centers and Racks



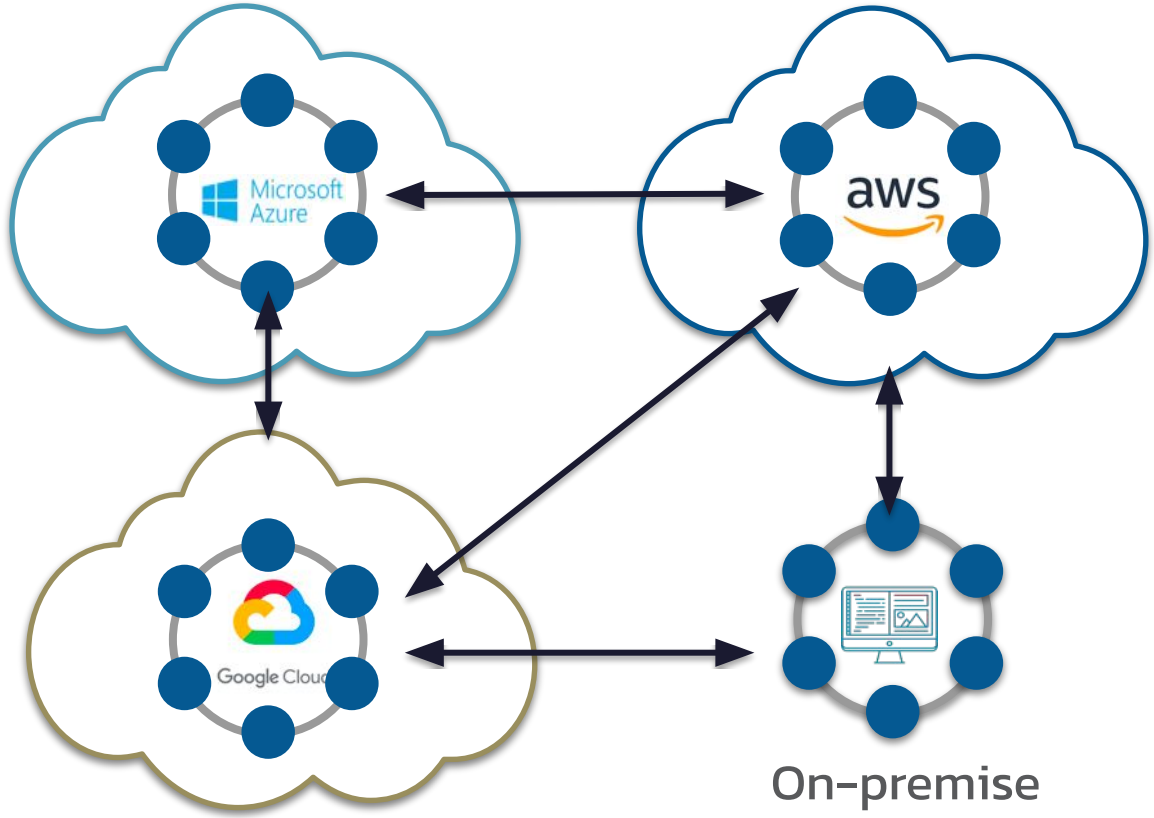



# Data Distributed Everywhere

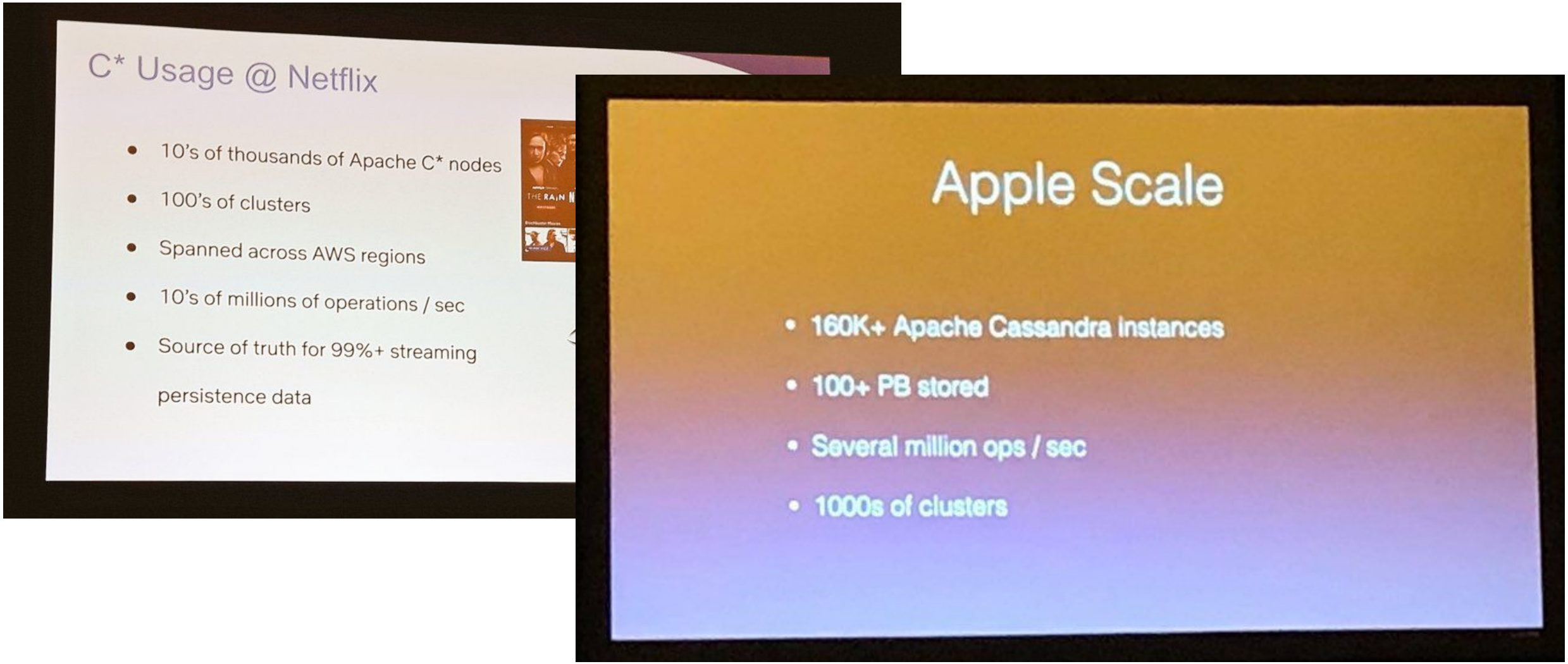
- Geographic Distribution



- Hybrid-Cloud and Multi-Cloud



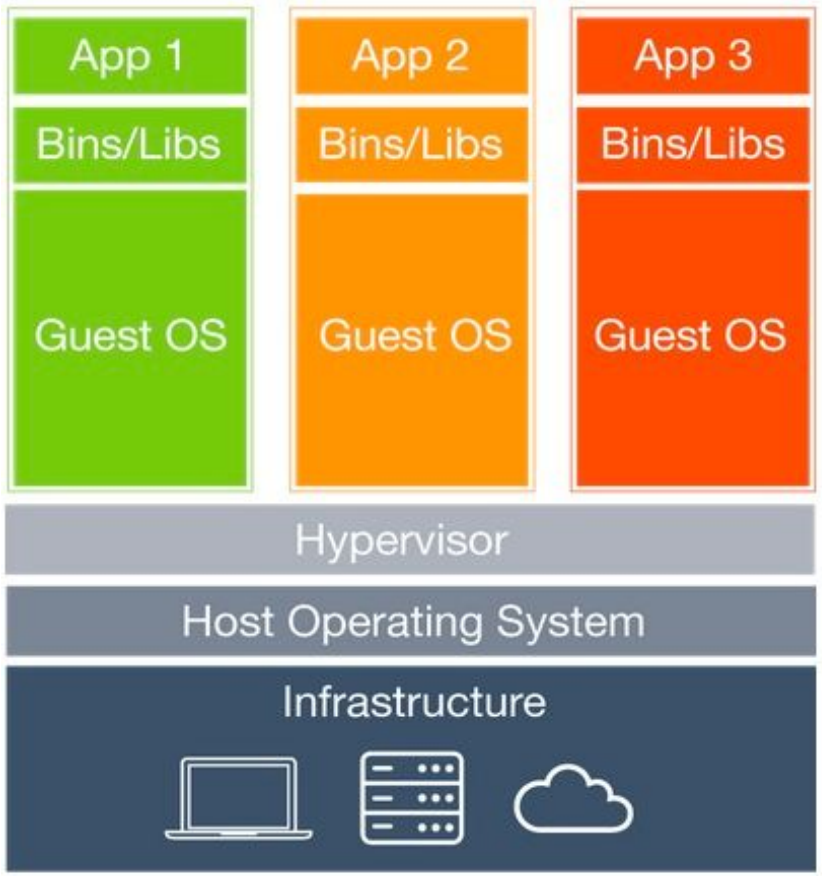
# References (2019)

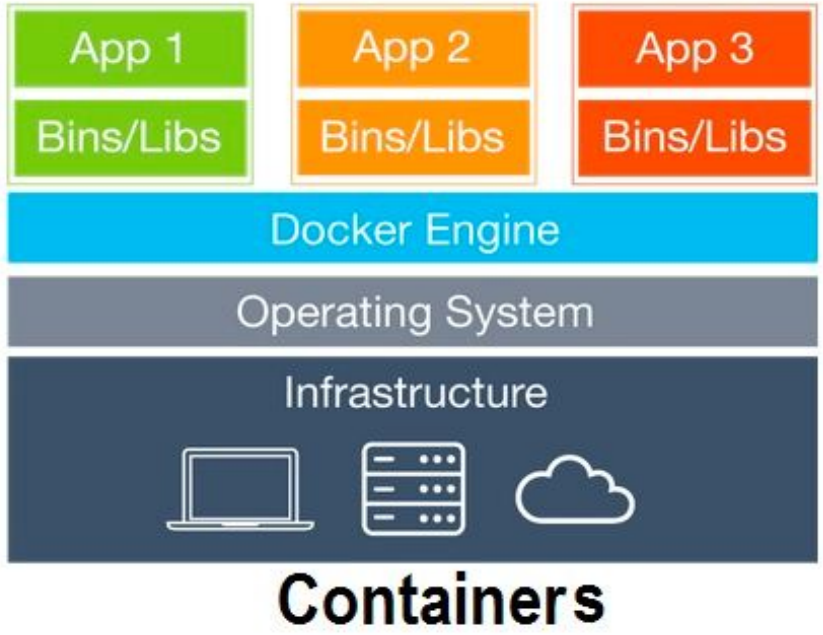


# kubernetes

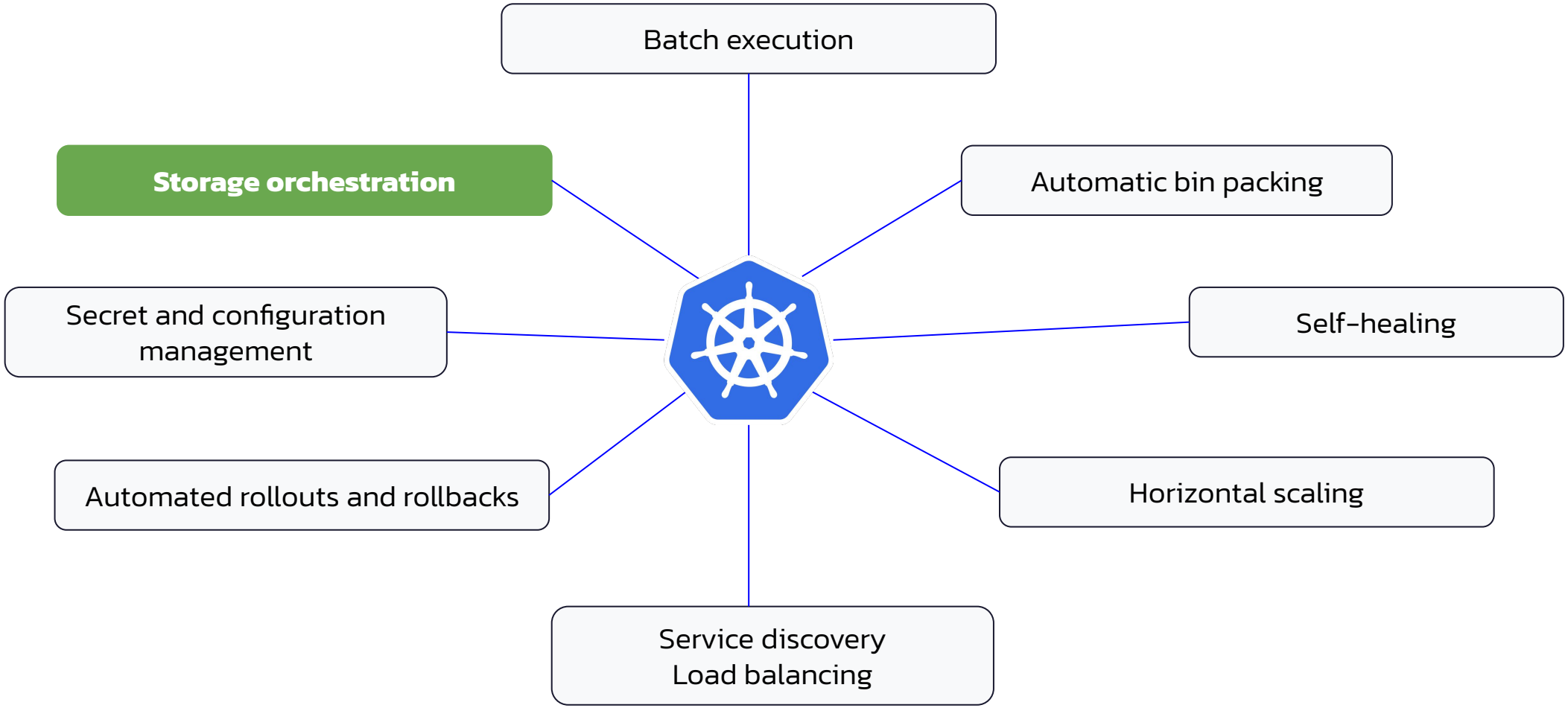
"Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications."

# Why containers?



# Why Kubernetes ?

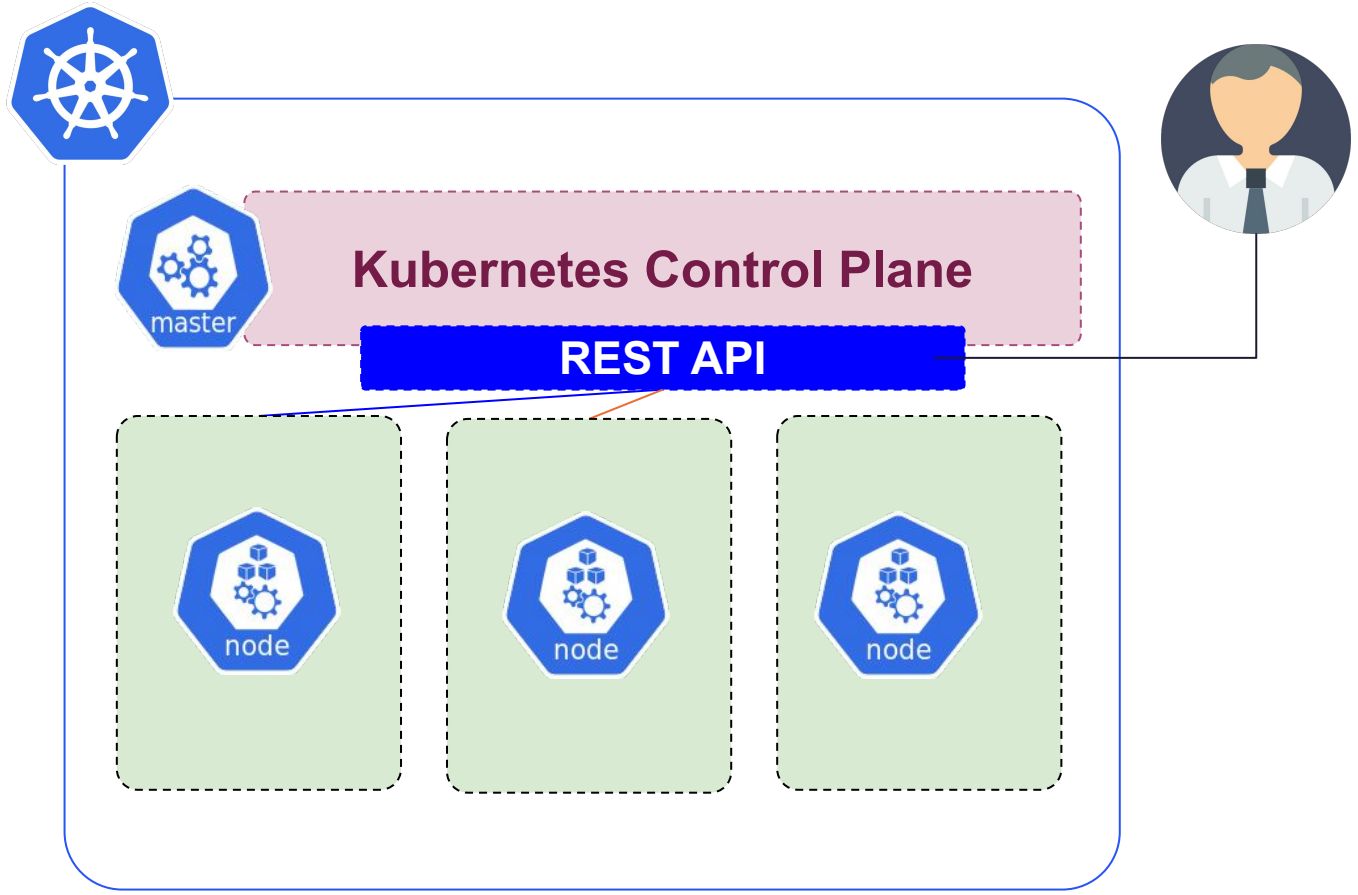


# K8s Infrastructure

**Cluster:**
Kubernetes cluster.

**Master:**
Kubernetes Control Plane.

**Node:**
Worker machine



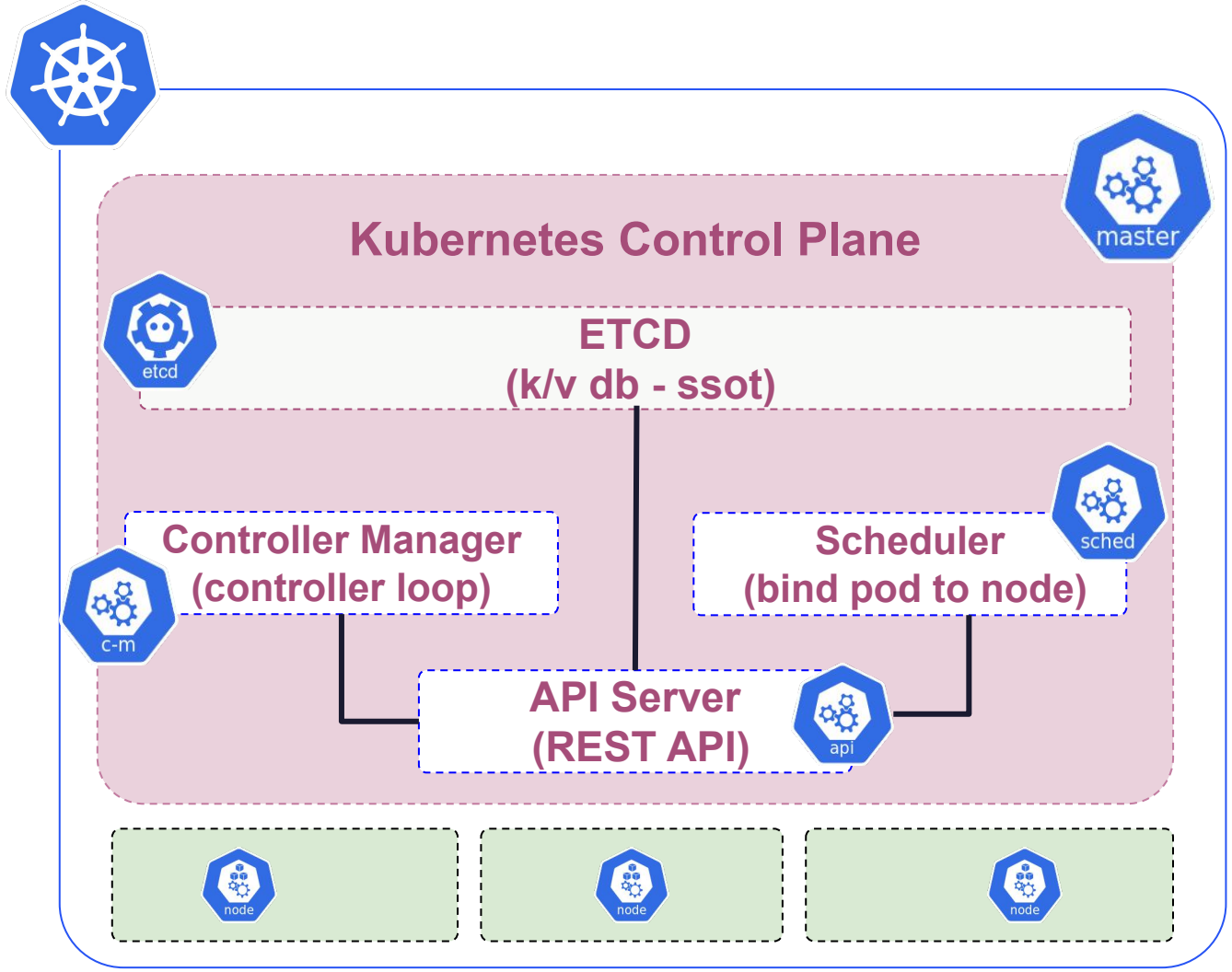
# K8s Infrastructure – Control Plane


**K8s API Server**
Kubernetes API


**Controller Manager**
Kubernetes controller manager


**Scheduler**
In charge of placing pods


**ETCD**
Kubernetes' backing store.



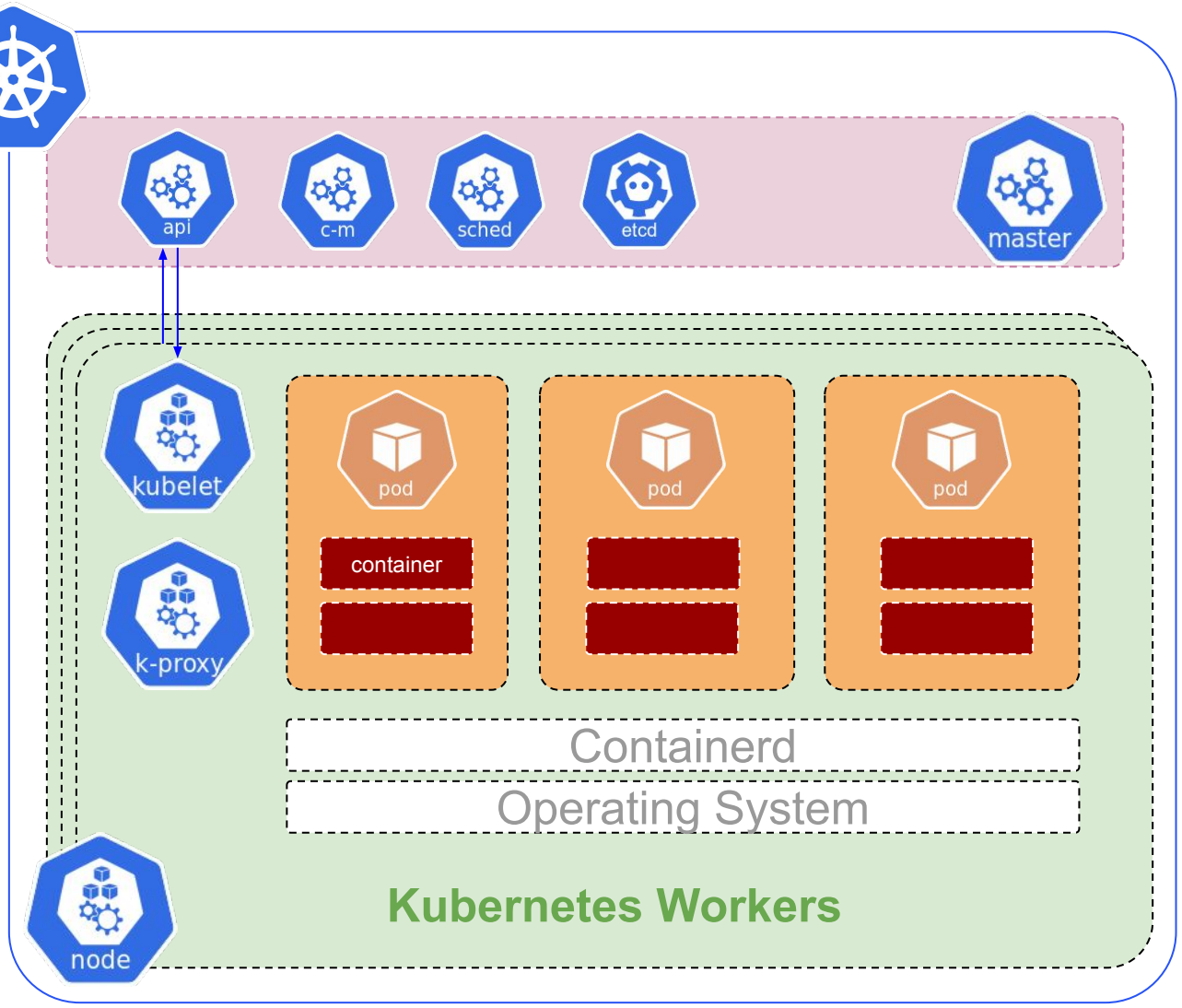
# K8s Infrastructure – Worker Node


**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.

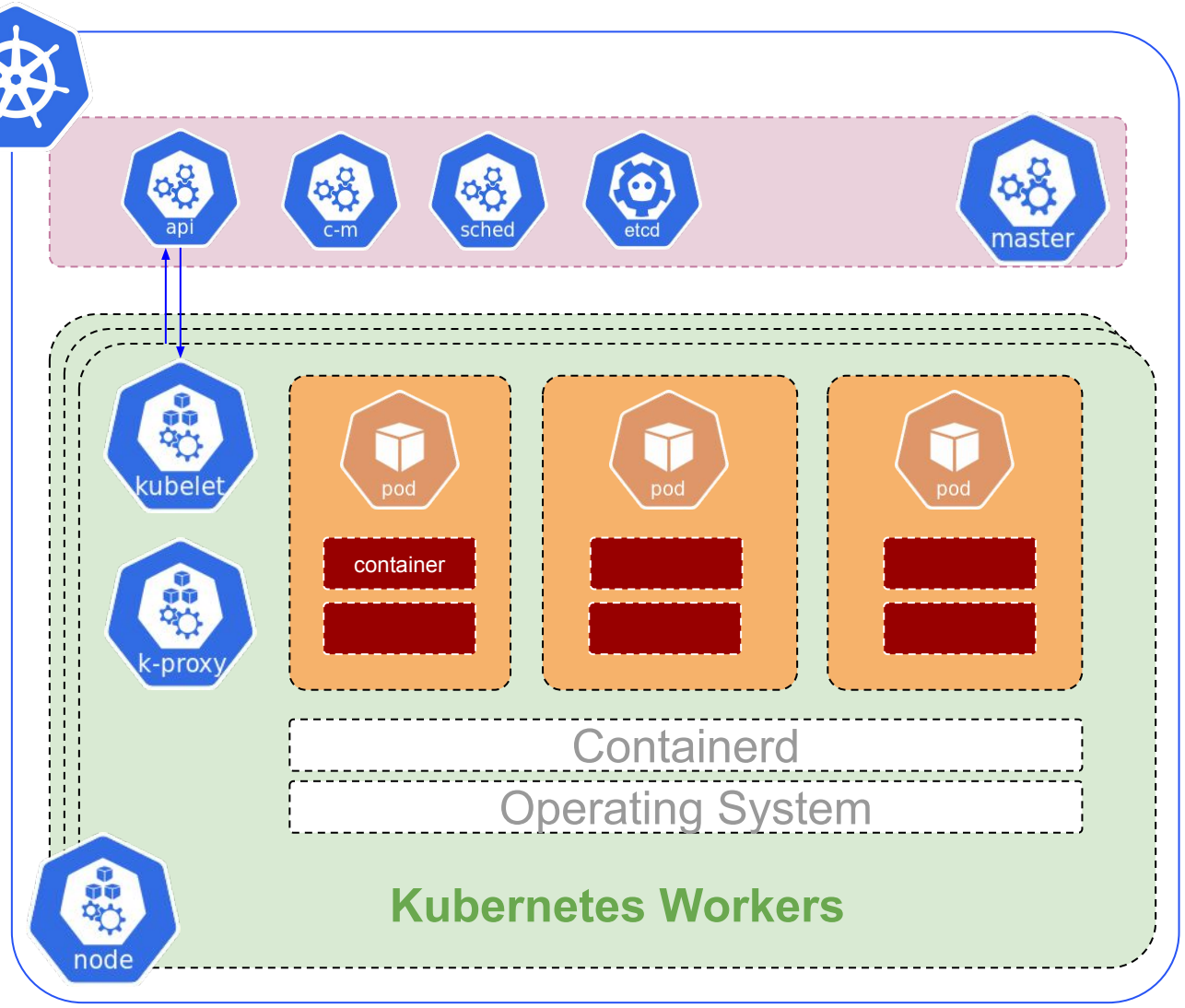

**POD**
Collection of containers that can run on a host.

This resource is created by clients and scheduled onto hosts.



# Your Task: Deploy C* to K8s. You have 90 minutes. Go!

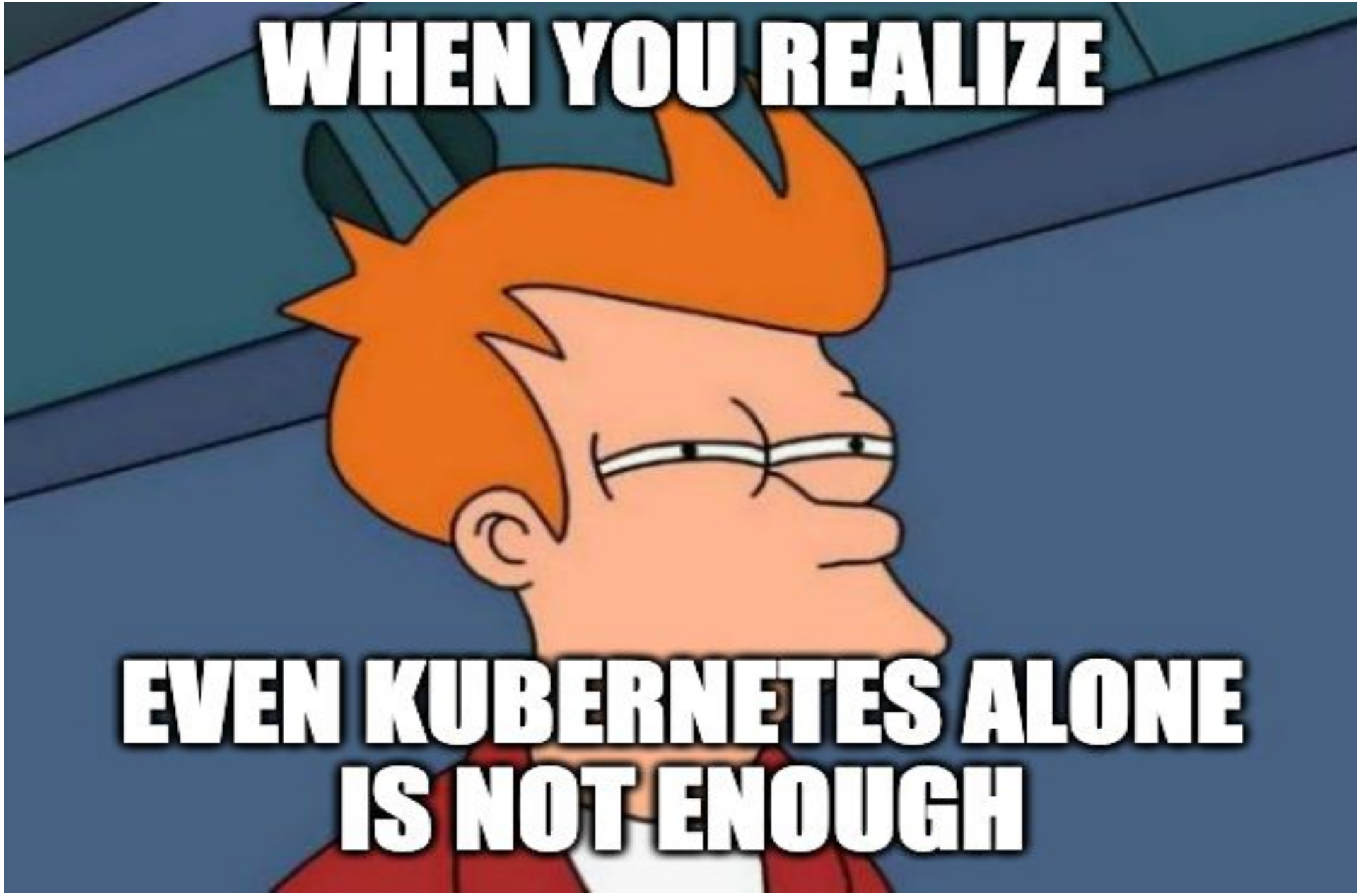

YOUR MISSION
SHOULD YOU
CHOOSE TO
ACCEPT IT



# So much to do, so little time...

- Setup networking
- Setup storage
- Setup firewalls
- Install Cassandra on node1
- Install Cassandra on node2
- Install Cassandra on node3
- Install Prometheus and Grafana
- Install agents for management
- Configure backups
- Configure repairs
- Connect our app
- Install all user agents
- and SO MUCH MORE!!!!!

# Just Kidding!!!


WHEN YOU REALIZE
EVEN KUBERNETES ALONE
IS NOT ENOUGH

# K8ssandra – Apache Cassandra™ meets Kubernetes!

- Housekeeping and Setup
- Cassandra and Kubernetes Basics
- Introducing K8ssandra
- Monitoring Cassandra
- Working with Data
- Scaling Up and Down
- API Access with Stargate
- Running Cassandra Repairs
- Wrapping up

# K8SSANDRA

Full working scalable database with administration tools and easy data access

“Cloud native”

# K8ssandra Components


Kubernetes ingress for external access


Data Gateway providing REST, GraphQL, Document APIs


Scalable cloud-native database managed via cass-operator

Reaper
Medusa

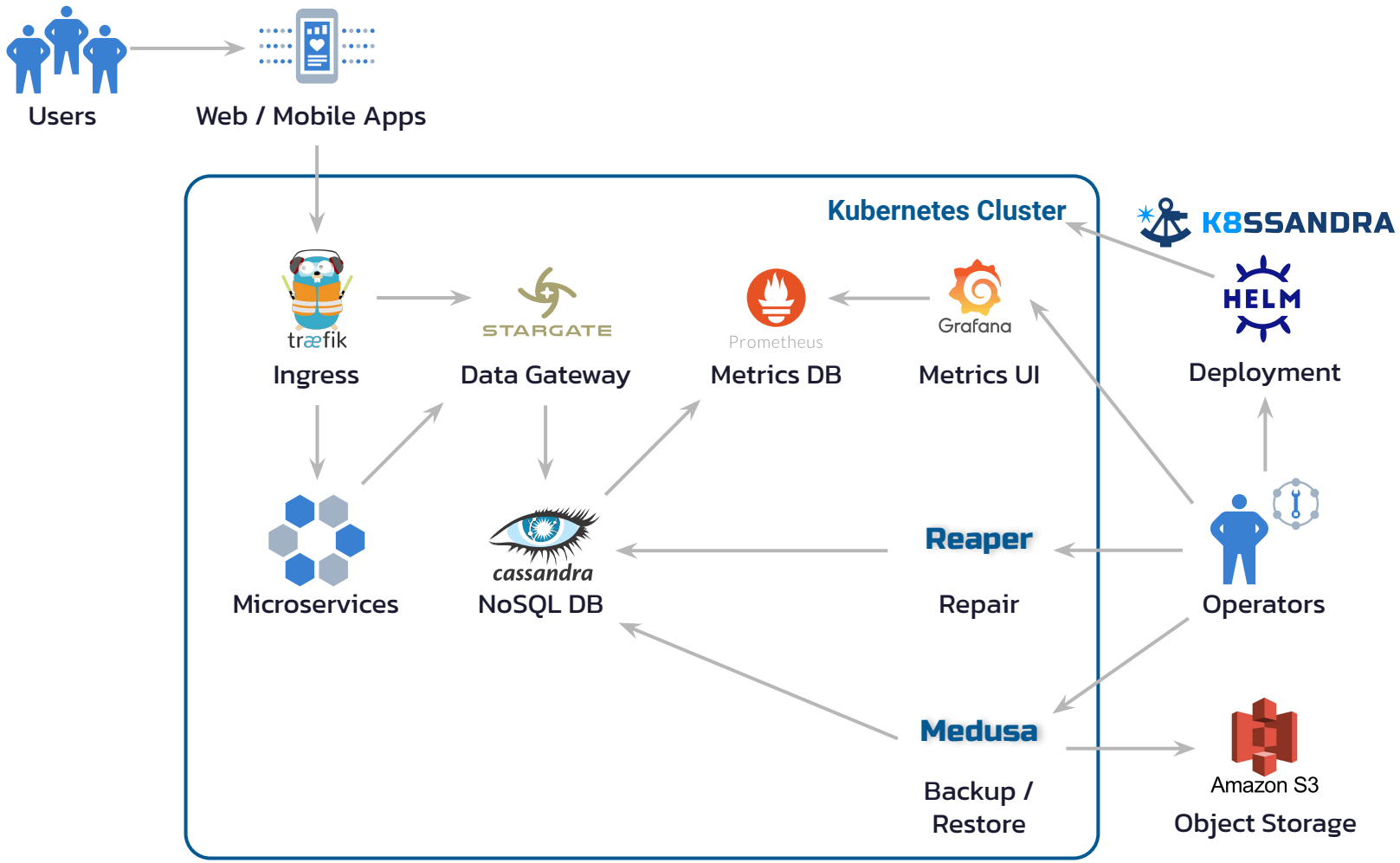
Cassandra utilities for repair and backup/restore



Metrics aggregation and visualization


Packaged and delivered via Helm charts

# K8ssandra Components in Context



# Setting up K8ssandra

```
helm repo add k8ssandra https://helm.k8ssandra.io/stable
helm repo add traefik https://helm.traefik.io/traefik
helm repo update
helm install k8ssandra k8ssandra/k8ssandra -f k8ssandra.yaml
```


https://k8ssandra.io/docs/getting-started/
https://k8ssandra.io/docs/getting-started/developer/
https://k8ssandra.io/docs/getting-started/site-engineer/

```yaml
cassandra:
  version: "3.11.10"
  cassandraLibDirVolume:
    storageClass: local-path
    size: 5Gi
  allowMultipleNodesPerWorker: true
  heap:
   size: 1G
   newGenSize: 1G
  resources:
    requests:
      cpu: 1000m
      memory: 2Gi
    limits:
      cpu: 1000m
      memory: 2Gi
  datacenters:
  - name: dc1
    size: 1
    racks:
    - name: default
kube-prometheus-stack:
  grafana:
  adminUser: admin
  adminPassword: admin123
stargate:
  enabled: true
  replicas: 1
  heapMB: 256
  cpuReqMillicores: 200
  cpuLimMillicores: 1000
```

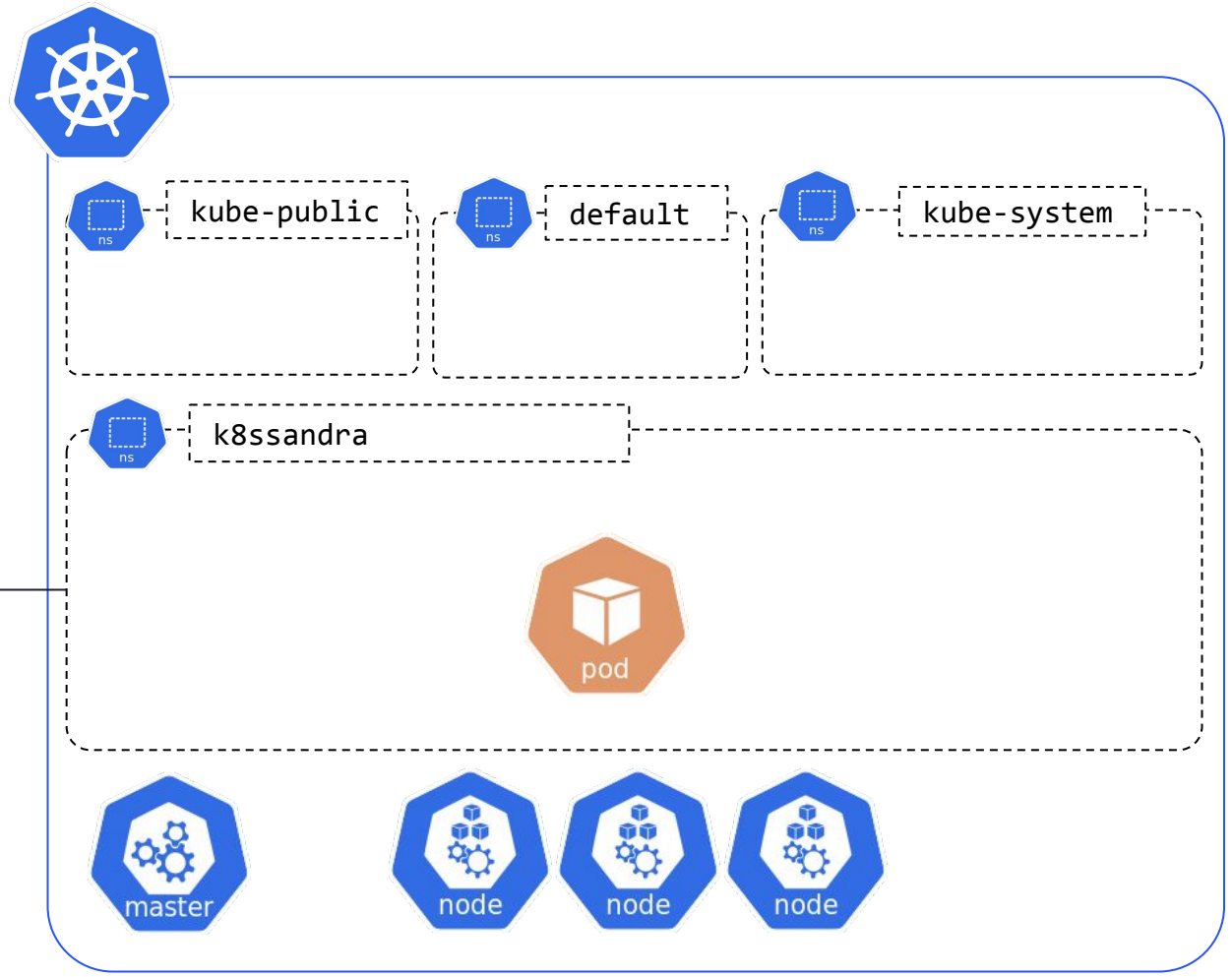
# Demo #1

- Setting up K8ssandra

# Kubernetes Namespaces


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

**We create resources in namespaces that span across nodes.**

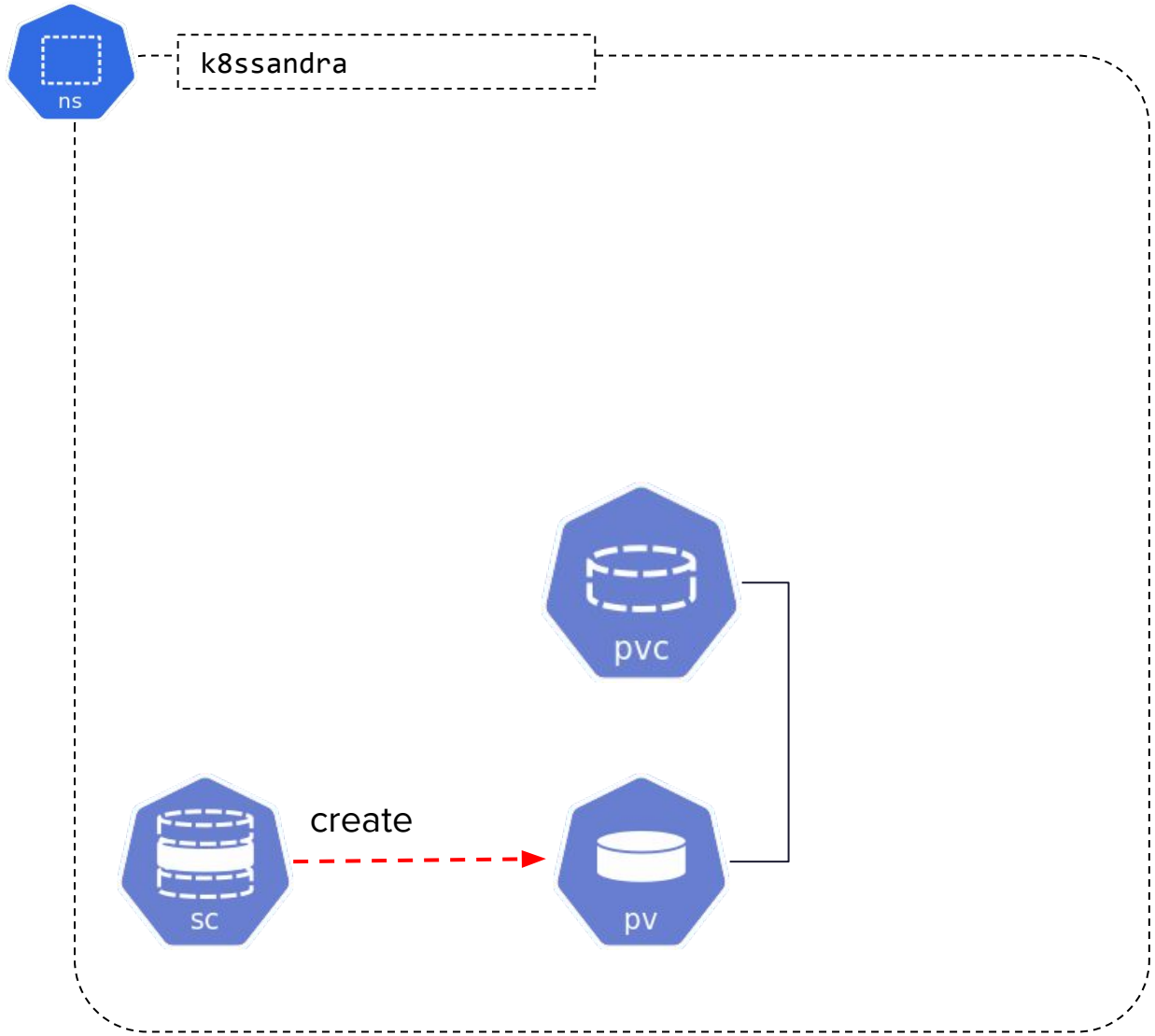


# K8s Primitives : Storage


**PersistentVolume:** is a storage resource provisioned by an administrator or dynamic provisioner.

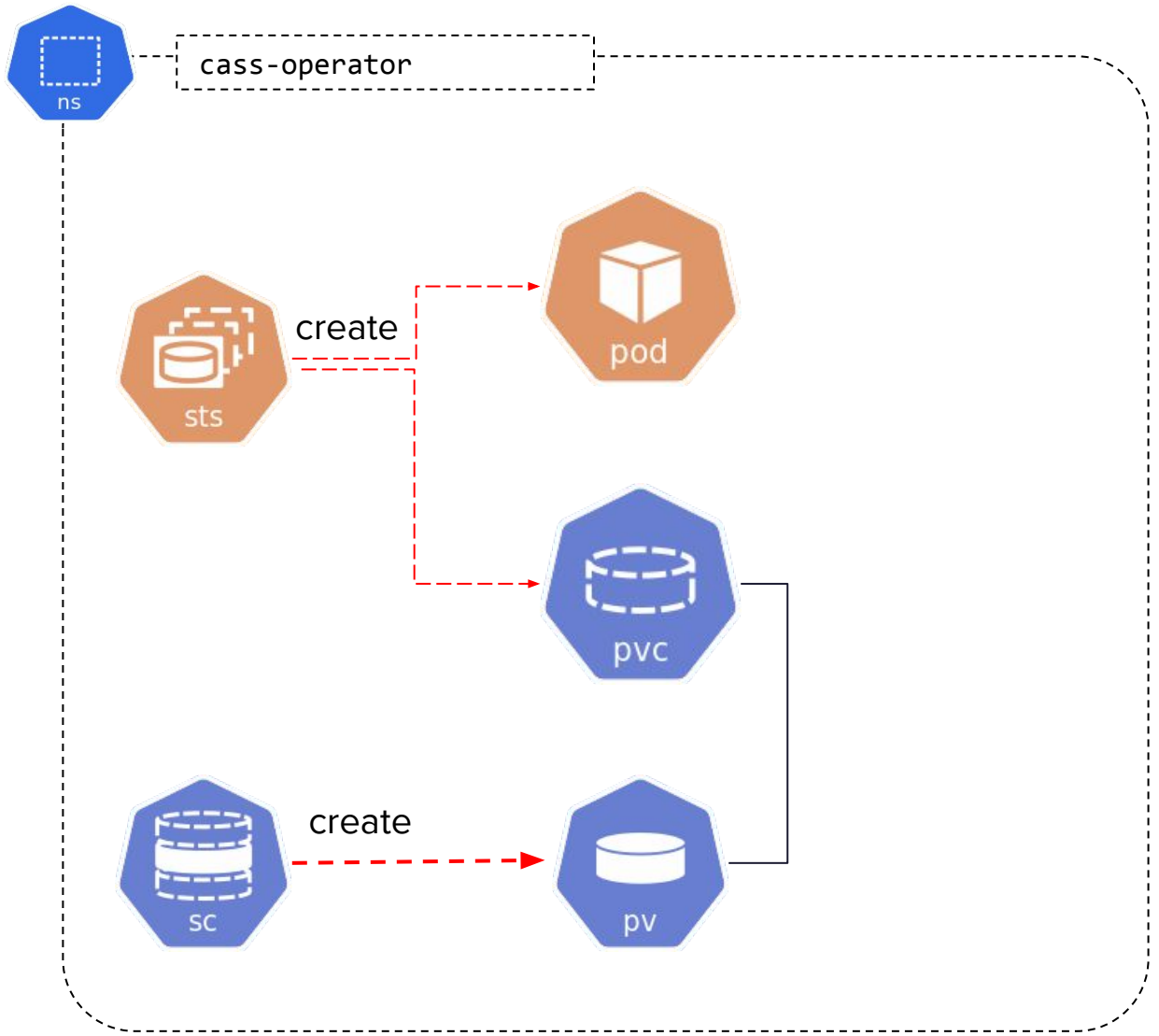

**PersistentVolumeClaim:** PersistentVolumeClaim is a user's request for and claim to a persistent volume.


**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/v1beta1
kind: CustomResourceDefinition
metadata:
  name: cassandradatacenters.cassandra.datastax.com
spec:
  group: cassandra.datastax.com
  names:
    kind: CassandraDatacenter
    listKind: CassandraDatacenterList
    plural: CassandraDatacenters
    shortNames:
      - cassdc
      - cassdcs
    singular: CassandraDatacenter
  scope: Namespaced
subresources:
  status: {}
...
```

- **Examples used in K8ssandra:**
  - `CassandraDataCenter` **(cass-operator)**
  - `CassandraBackup`, `CassandraRestore` **(cassandra-medusa)**
  - `Reaper` **(reaper)**
  - `GrafanaDataSource` **(grafana)**

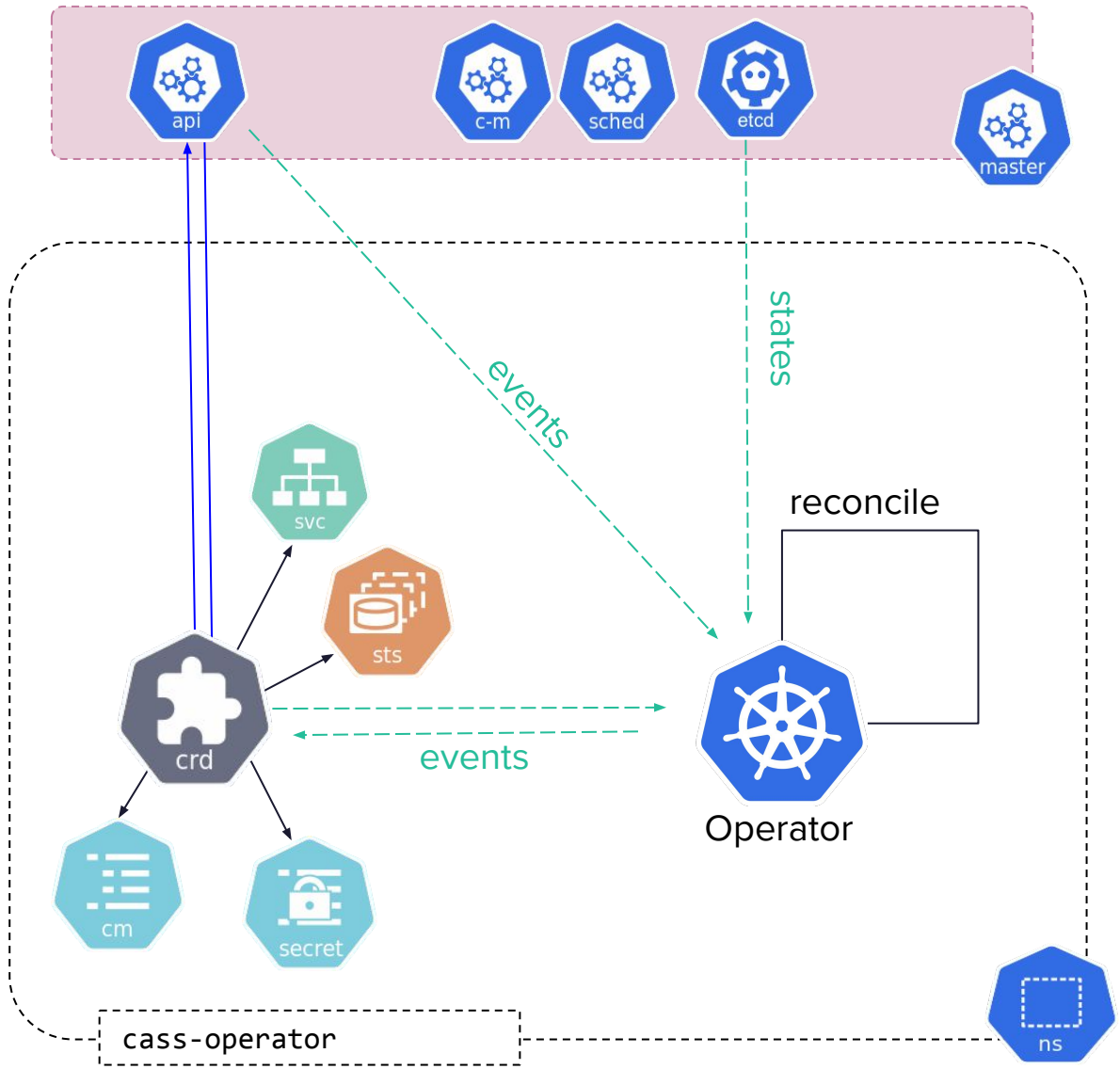
# K8s Primitives : Operator

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

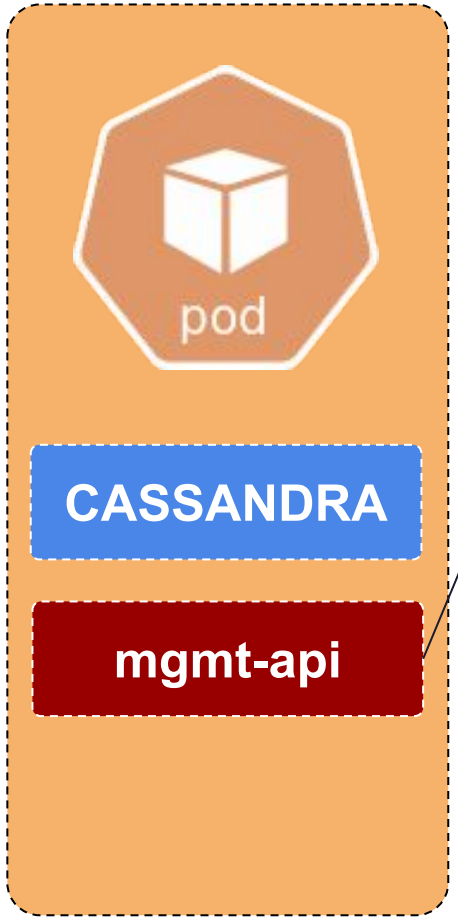
*A Kubernetes Operator helps extend the types of applications that can run on Kubernetes by allowing developers to provide additional knowledge to applications that need to maintain state." –Jonathan S. Katz*

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

- **Examples used in K8ssandra:**
  - `cass-operator` **(cass-operator)**
  - `reaper-operator` **(cassandra-reaper)**
  - `PrometheusOperator` **(Prometheus)**



# Kubernetes Probes (readiness, liveness)




## 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

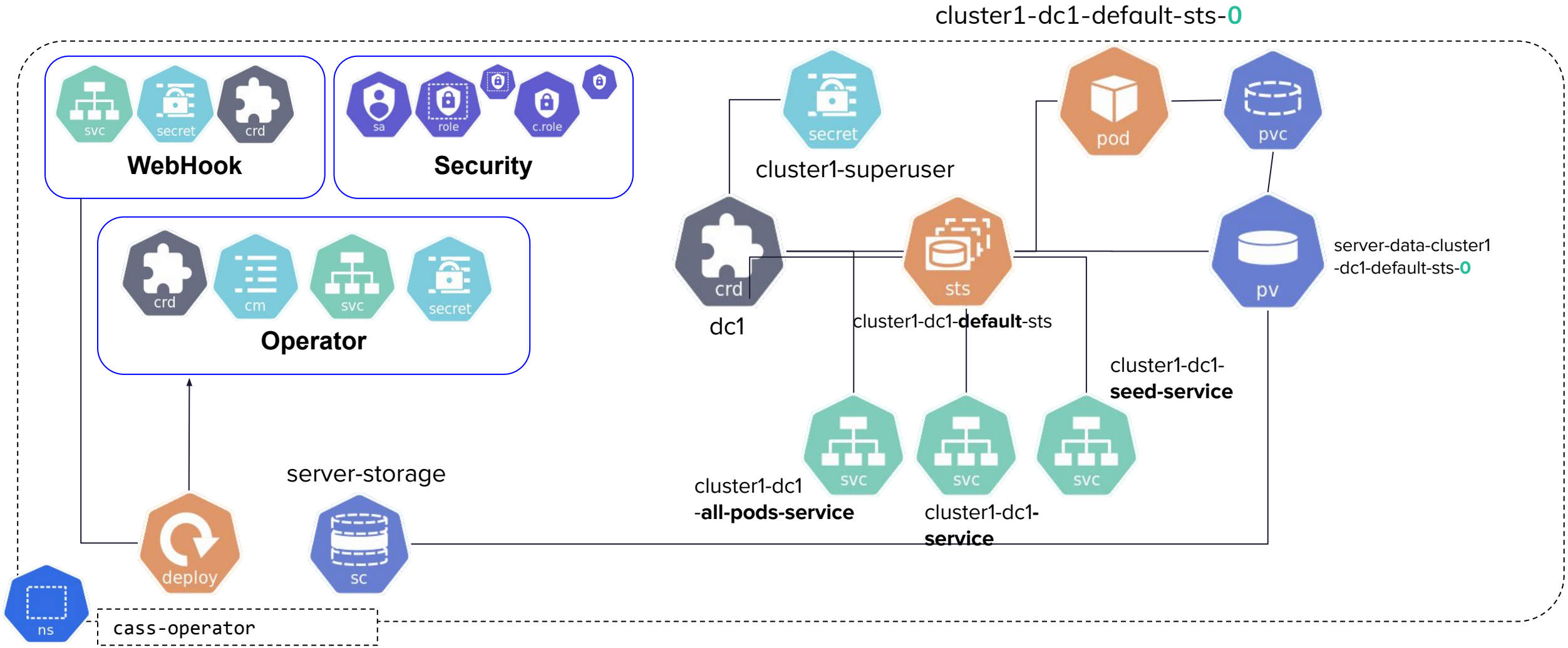
This is a Restful service for operating Apache Cassandra. You can find out more about the Management API on Github

Apache 2.0

**default**

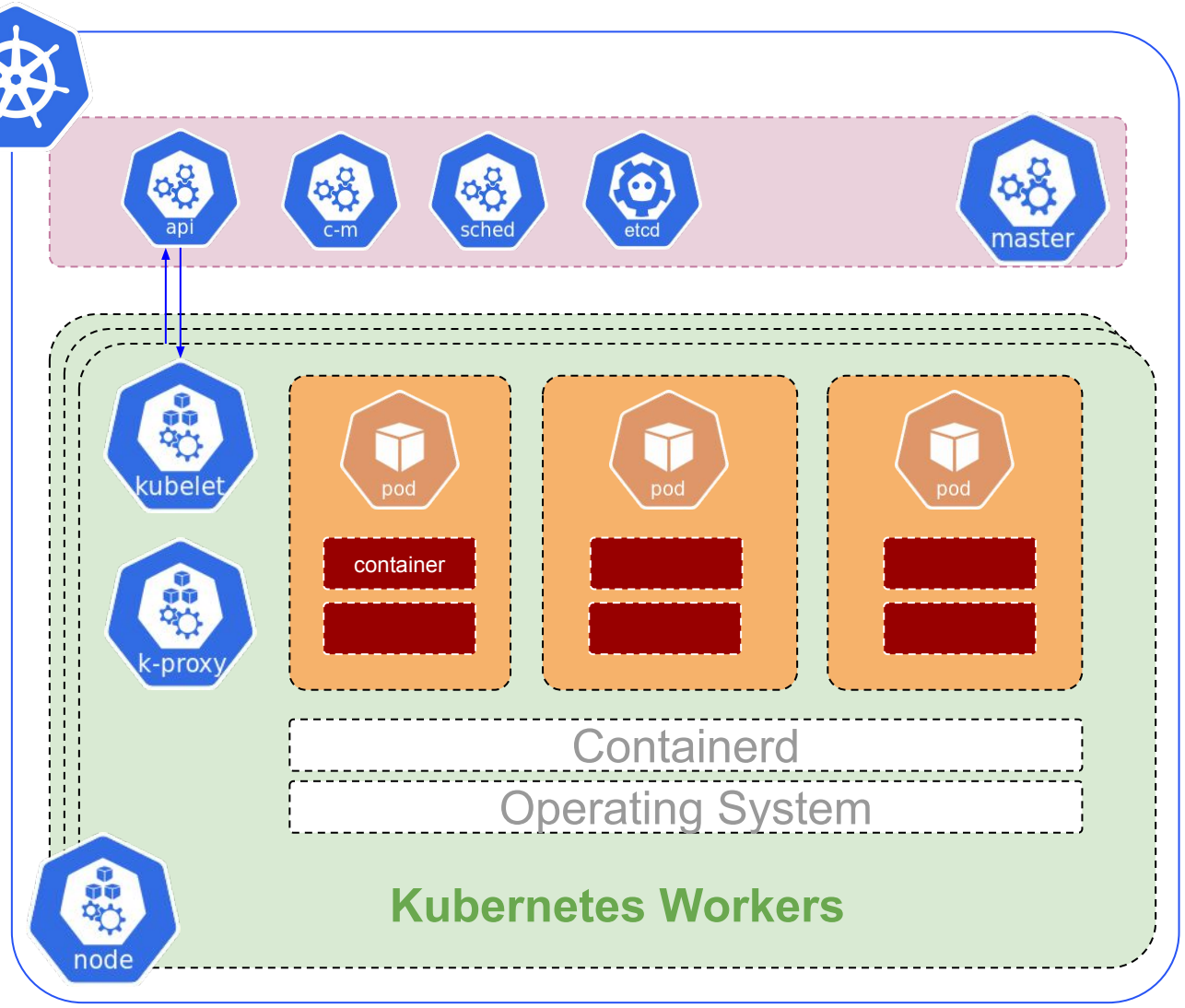
| | | |
|---|---|---|
| POST | /api/v0/ops/auth/role | Creates a new user role |
| GET | /api/v0/probes/liveness | Indicates whether this service is running |
| GET | /api/v0/probes/readiness | Indicates whether the Cassandra service is ready to service requests |
| GET | /api/v0/probes/cluster | Indicated 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) |

# Sample Deployment with Cass-operator



# Remember this?


YOUR MISSION
SHOULD YOU
CHOOSE TO
ACCEPT IT



# Deploying a Cassandra Cluster in Kubernetes, Simplified

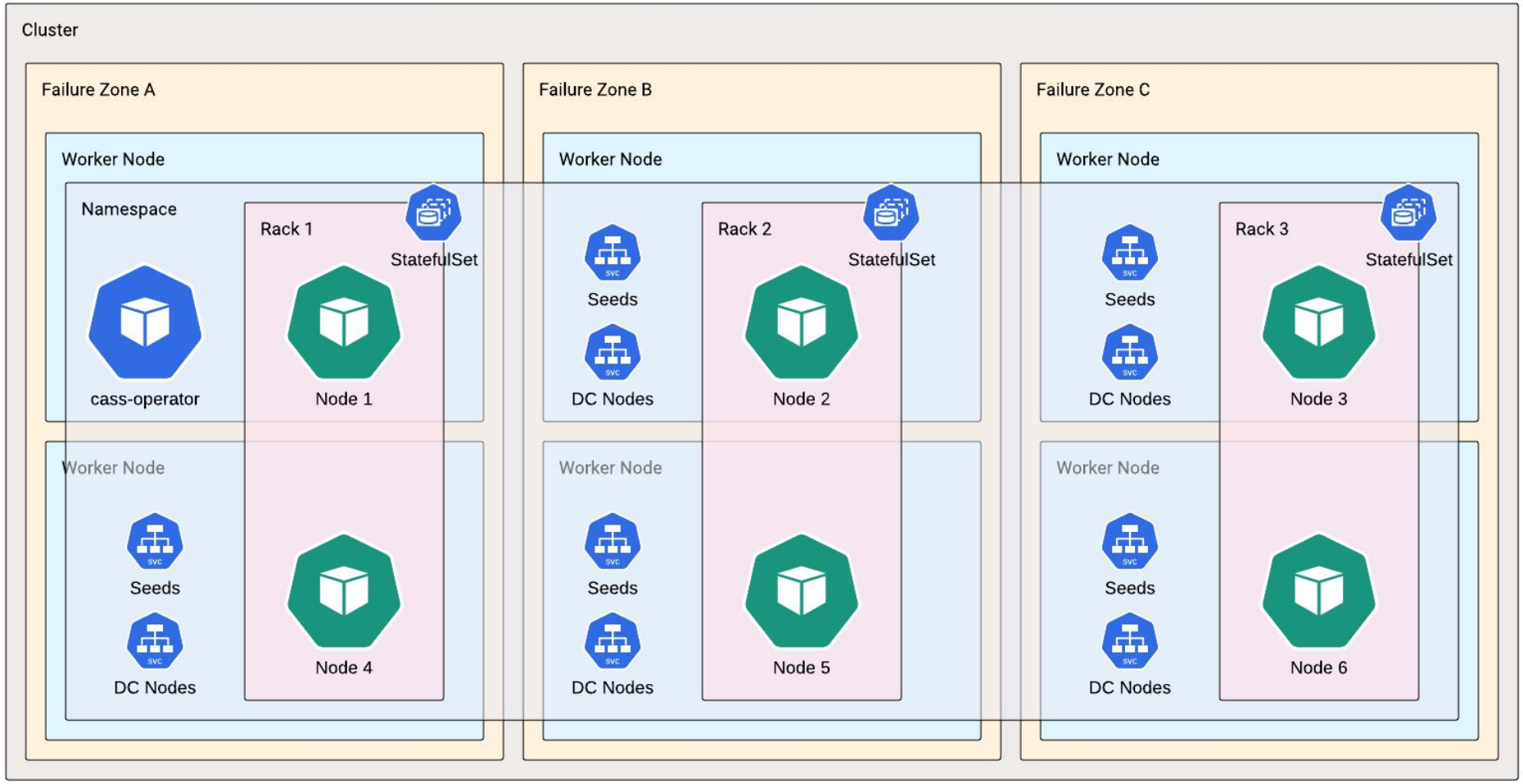
- Setup networking
- Setup storage
- Setup firewalls
- Install Cassandra on node1
- Install Cassandra on node2
- Install Cassandra on node3
- Install Prometheus and Grafana
- Install agents for management
- Configure backups
- Configure repairs
- Connect our app
- Install all user agents
- and SO MUCH MORE!!!!!

**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)
- Scaling up and down simply

Normal K8s Deployment stuff + best practices for C* ops

# Example Deployment Across Multiple K8s Workers

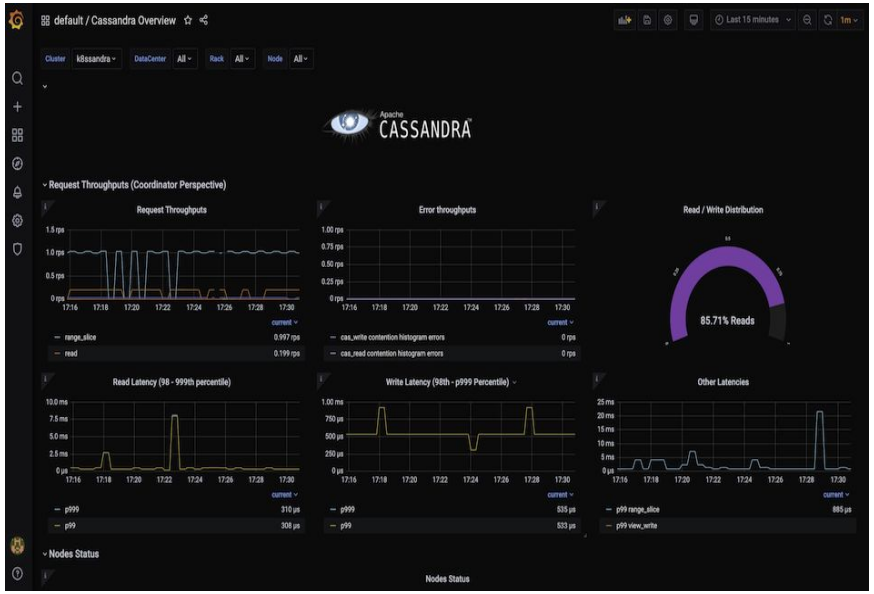


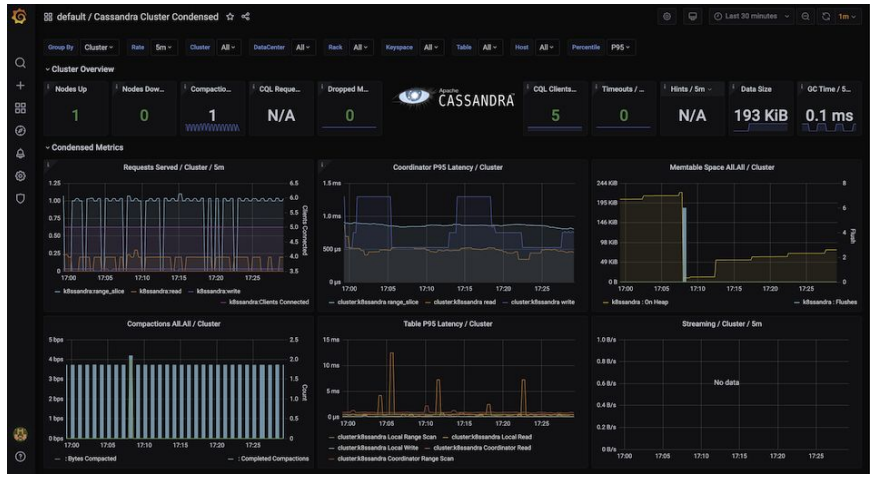
# K8ssandra - Apache Cassandra™ meets Kubernetes!

- Housekeeping and Setup
- Cassandra and Kubernetes Basics
- Introducing K8ssandra
- Monitoring Cassandra
- Working with Data
- Scaling Up and Down
- API Access with Stargate
- Running Cassandra Repairs
- Wrapping up

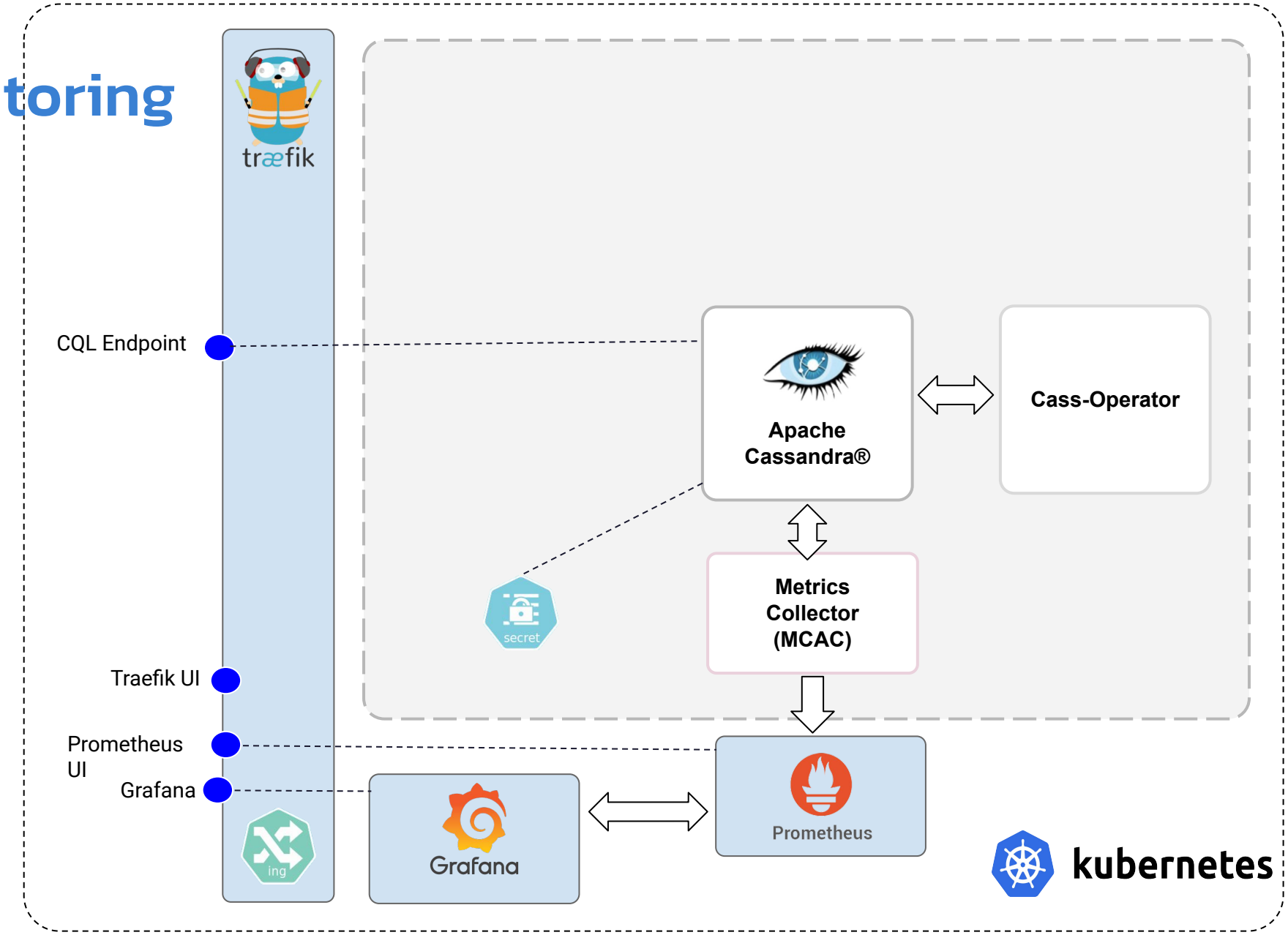
# Monitoring (Kube-prometheus-stack)

https://github.com/datastax/metric-collector-for-apache-cassandra




# Monitoring


træfik
CQL Endpoint
Traefik UI
Prometheus UI
Grafana
ing
Apache Cassandra®
Cass-Operator
secret
Metrics Collector (MCAC)
Grafana
Prometheus
kubernetes

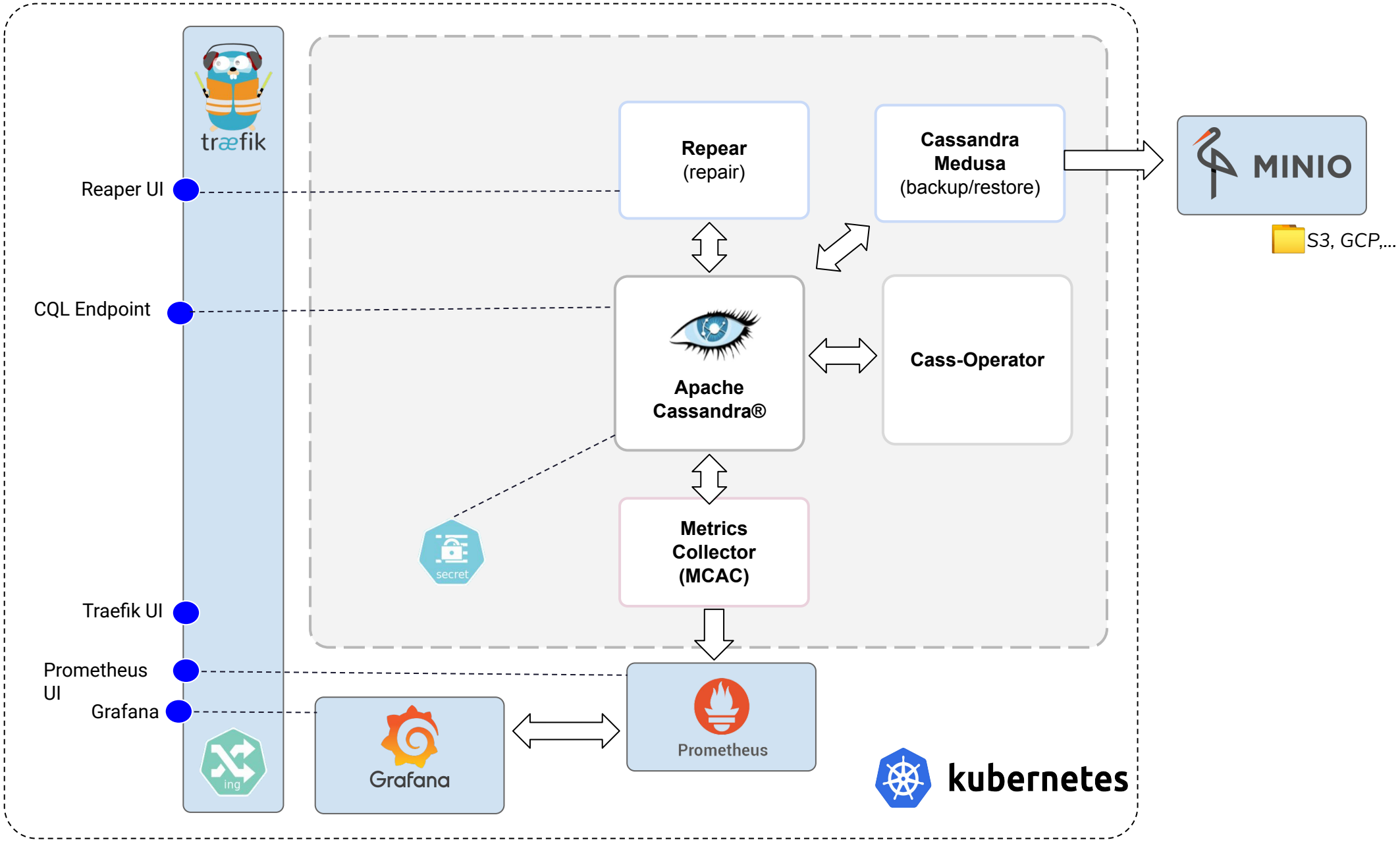
# Demo #2

- Monitoring Cassandra with Prometheus and Grafana

# K8ssandra - Apache Cassandra™ meets Kubernetes!

- Housekeeping and Setup
- Cassandra and Kubernetes Basics
- Introducing K8ssandra
- Monitoring Cassandra
- Working with Data
- Scaling Up and Down
- API Access with Stargate
- Running Cassandra Repairs
- Wrapping up

# Accessing the Data



# Demo #3

- Working with Data

# K8ssandra - Apache Cassandra™ meets Kubernetes!

- Housekeeping and Setup
- Cassandra and Kubernetes Basics
- Introducing K8ssandra
- Monitoring Cassandra
- Working with Data
- Scaling Up and Down
- API Access with Stargate
- Running Cassandra Repairs
- Wrapping up

# Two ways to scale a K8ssandra Cluster

1. Config files

```yaml
# Default values for k8ssandra-cluster.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

name: k8ssandra
clusterName: k8ssandra
datacenterName: dc1
size: 1

reaperOperator:
  enabled: true

reaper:
  enabled: true

  jmx:
    username: ""
    password: ""

k8ssandra:
  enabled: true
```

2. Simple command

```
helm upgrade demo k8ssandra/k8ssandra --set k8ssandra.size=3 --reuse-values
```


https://k8ssandra.io/docs/topics/provision-a-cluster/

# Demo #4

- Scaling Up and Down

# K8ssandra - Apache Cassandra™ meets Kubernetes!

- Housekeeping and Setup
- Cassandra and Kubernetes Basics
- Introducing K8ssandra
- Monitoring Cassandra
- Working with Data
- Scaling Up and Down
- API Access with Stargate
- Running Cassandra Repairs
- Wrapping up

# Stargate Rationale “Start with Why” *Simon Sinek*

## Developers

- Do you like learning query languages?
- Do you care about how your data is stored ?
- Do you like installing and running databases locally?

“Start with Why” Simon Sinek

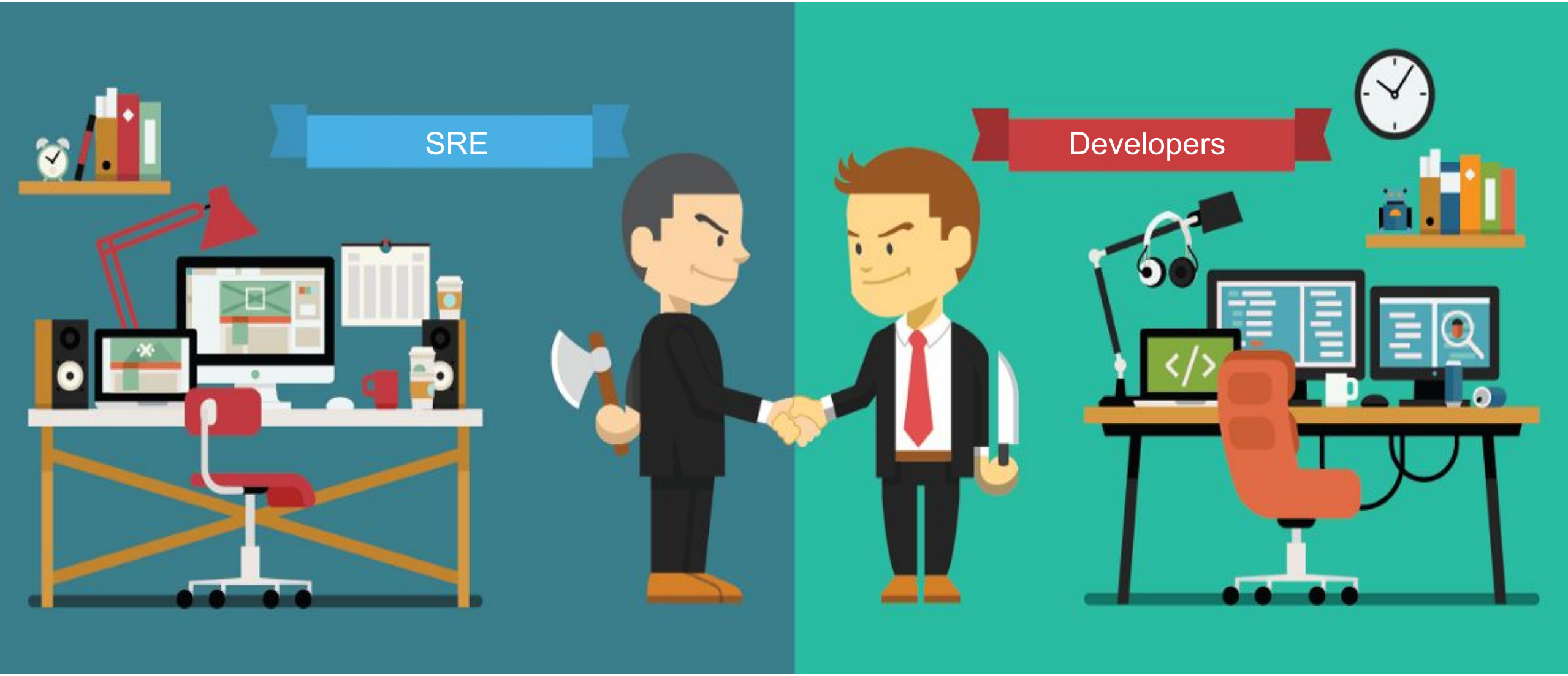
## SREs, Operators, DBAs

- Do you allow developers to execute direct queries against your DB ?
- Do you like opening port ranges like 0-65536 for apps

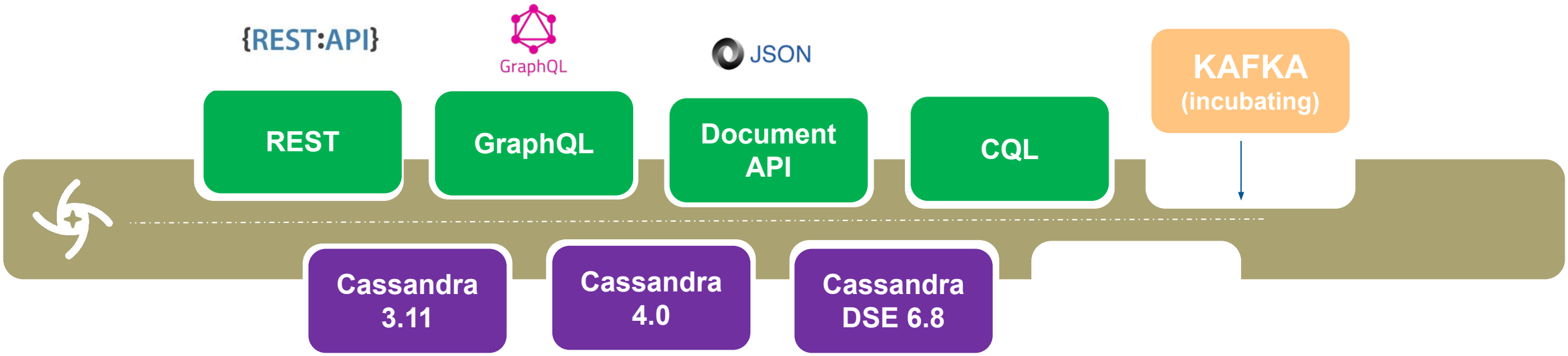
## Architects, CIOs

- Do you like creating dedicated projects and hiring people just to create APIs?

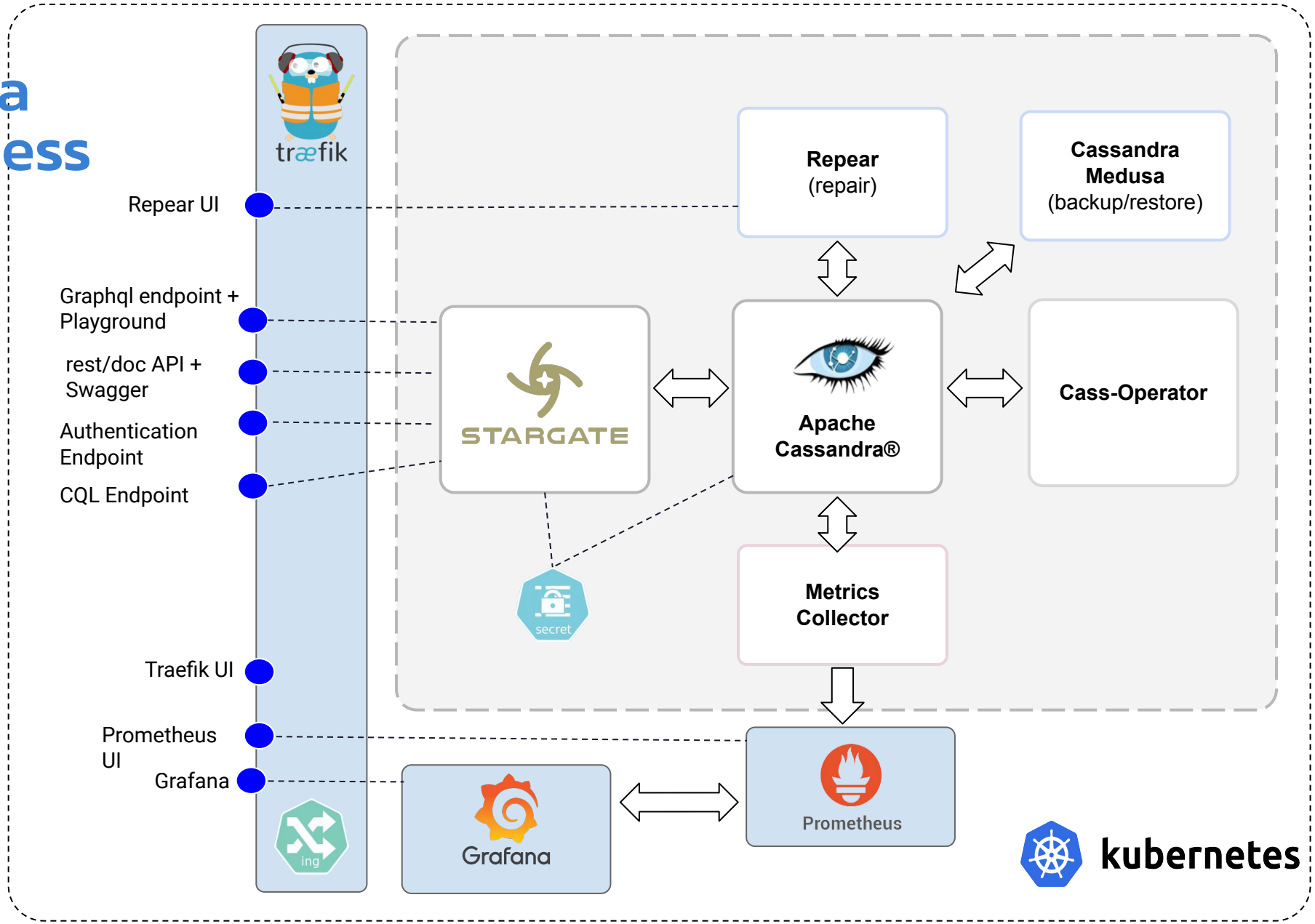
# And when those 2 meet each other...



# Stargate APIs



# Data Access


træfik
Repear UI
Graphql endpoint + Playground
rest/doc API + Swagger
Authentication Endpoint
CQL Endpoint
Traefik UI
Prometheus UI
Grafana
ing
Repear
(repair)
Cassandra
Medusa
(backup/restore)
STARGATE
Apache Cassandra®
Cass-Operator
secret
Metrics Collector
Grafana
Prometheus
kubernetes

# Demo #5

- Stargate

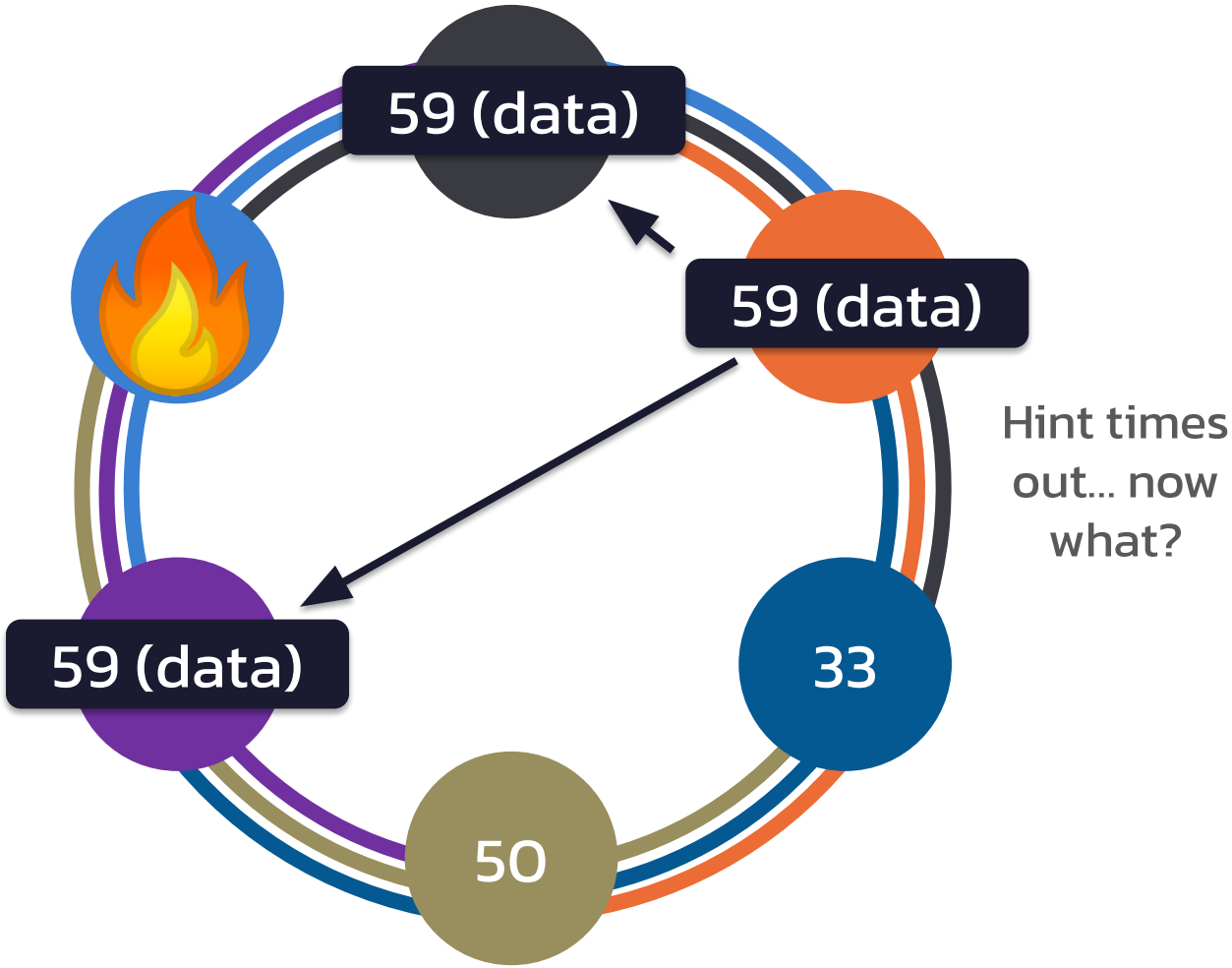
# K8ssandra - Apache Cassandra™ meets Kubernetes!

- Housekeeping and Setup
- Cassandra and Kubernetes Basics
- Introducing K8ssandra
- Monitoring Cassandra
- Working with Data
- Scaling Up and Down
- API Access with Stargate
- Running Cassandra Repairs
- Wrapping up

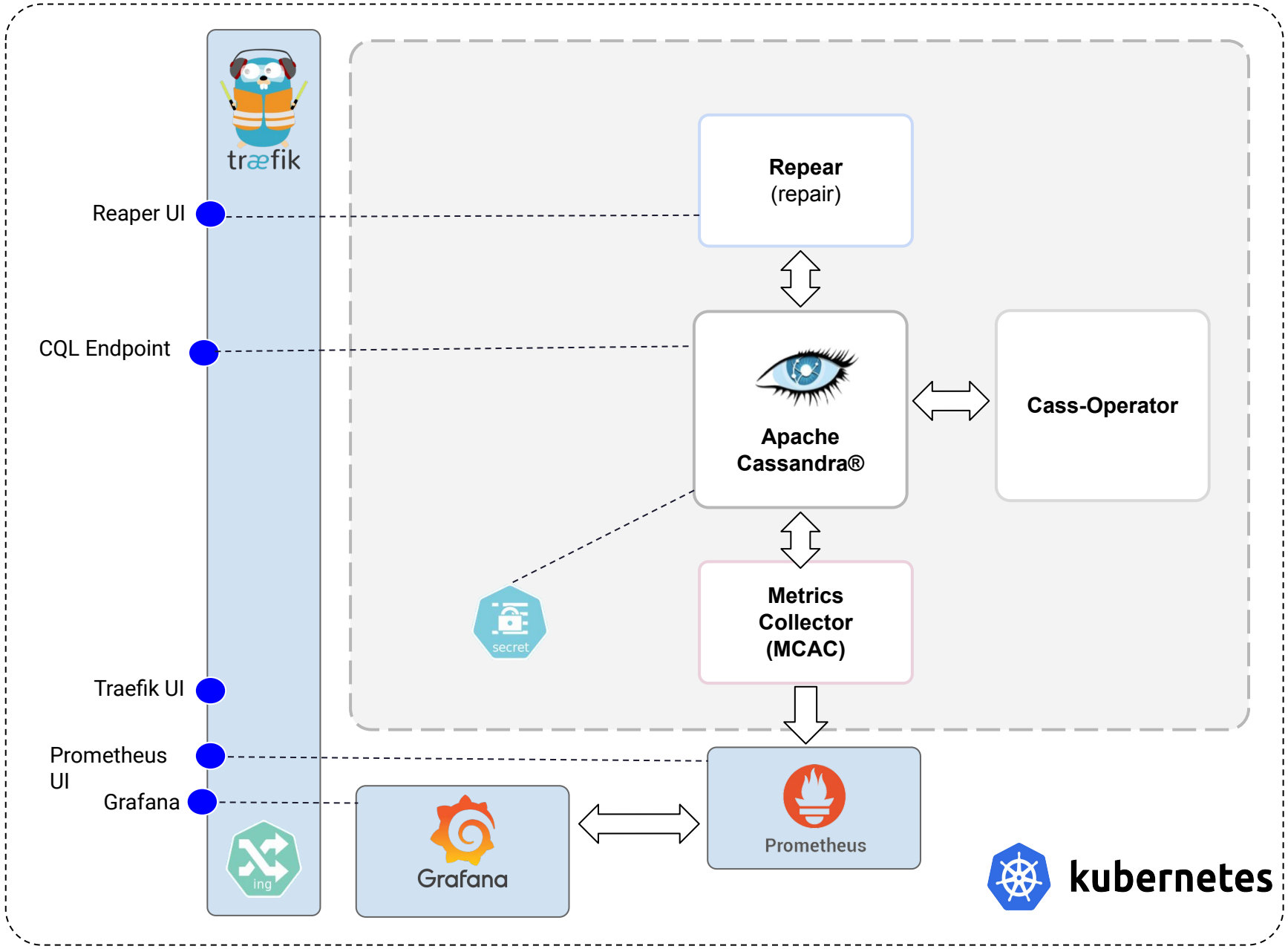
# Repairs in Cassandra

- Repairs are the "last resort" to keep data across a Cassandra cluster in sync
- If repairs don't happen the health of the data will slowly degrade over time due to Entropy

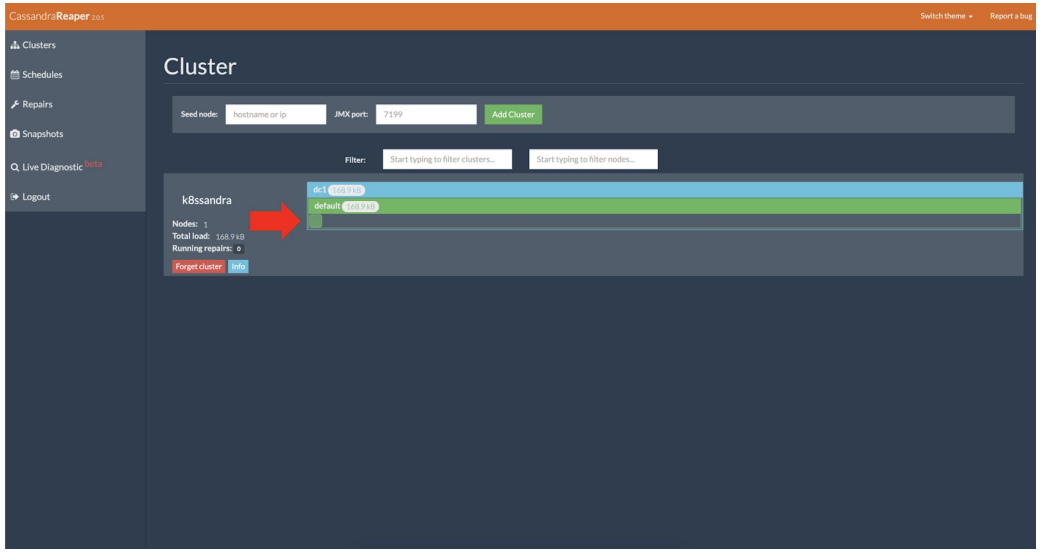
"Without repairs, your database is *hopefully consistent*"
– Vinay Kumar Chella, Principal Cloud Database Architect, Netflix

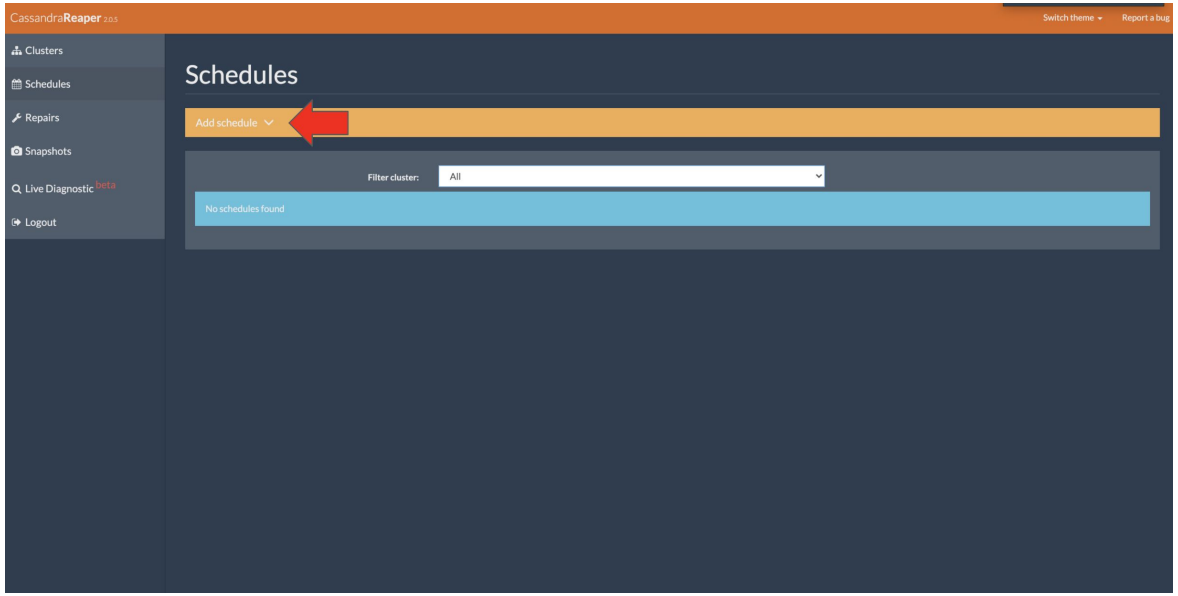


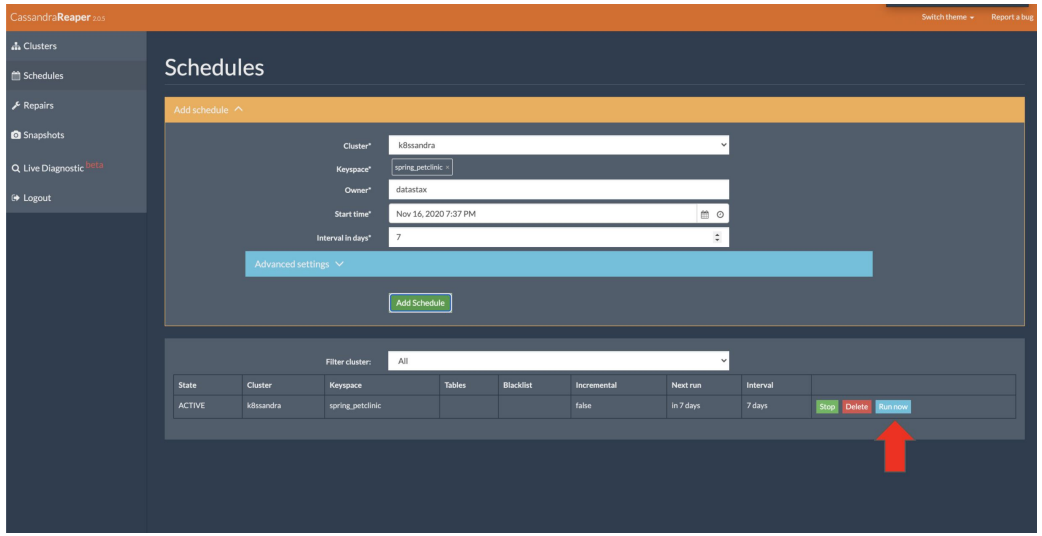
# Repair


træfik
Reaper UI
CQL Endpoint
Traefik UI
Prometheus UI
Grafana
ing
Repear (repair)
Apache Cassandra®
Cass-Operator
Metrics Collector (MCAC)
secret
Grafana
Prometheus
kubernetes

# Cassandra Reaper




## Repair

Start a new repair

Filter cluster : All

Running

| ID | State | Cluster | Keyspace | CFs | Incremental | Repaired | |
|---|---|---|---|---|---|---|---|
| 90bc2910-5022-11e7-b7cc-039e197701f7 | PAUSED | newcluster | test | | false | 162/201 | Activate Abort |
| a8854a70-4ffc-11e7-8025-872a3b38b99e | NOT_STARTED | newcluster | test | | false | 0/201 | Activate Delete |

Done

| ID | State | Cluster | Keyspace | CFs | Incremental | Repaired | |
|---|---|---|---|---|---|---|---|
| 832607a0-46fc-11e7-b3c0-2dc25dfb993e | DONE | twcs308 | booya | | true | 3/3 | Delete |
| c7d63470-4ff7-11e7-a8a1-b957dacf6ec2 | DONE | newcluster | test | | false | 12/12 | Delete |
| 3bd206c0-54c5-11e7-a8f0-7336a93cc91f | DONE | twcs308 | booya | | true | 3/3 | Delete |
| f44c1700-54c4-11e7-a8f0-7336a93cc91f | DONE | twcs308 | booya | | true | 3/3 | Delete |

# Demo #6

- Running Repairs

# K8ssandra - Apache Cassandra™ meets Kubernetes!

- Housekeeping and Setup
- Cassandra and Kubernetes Basics
- Introducing K8ssandra
- Monitoring Cassandra
- Working with Data
- Scaling Up and Down
- API Access with Stargate
- Running Cassandra Repairs
- Backup and Restore
- Wrapping up

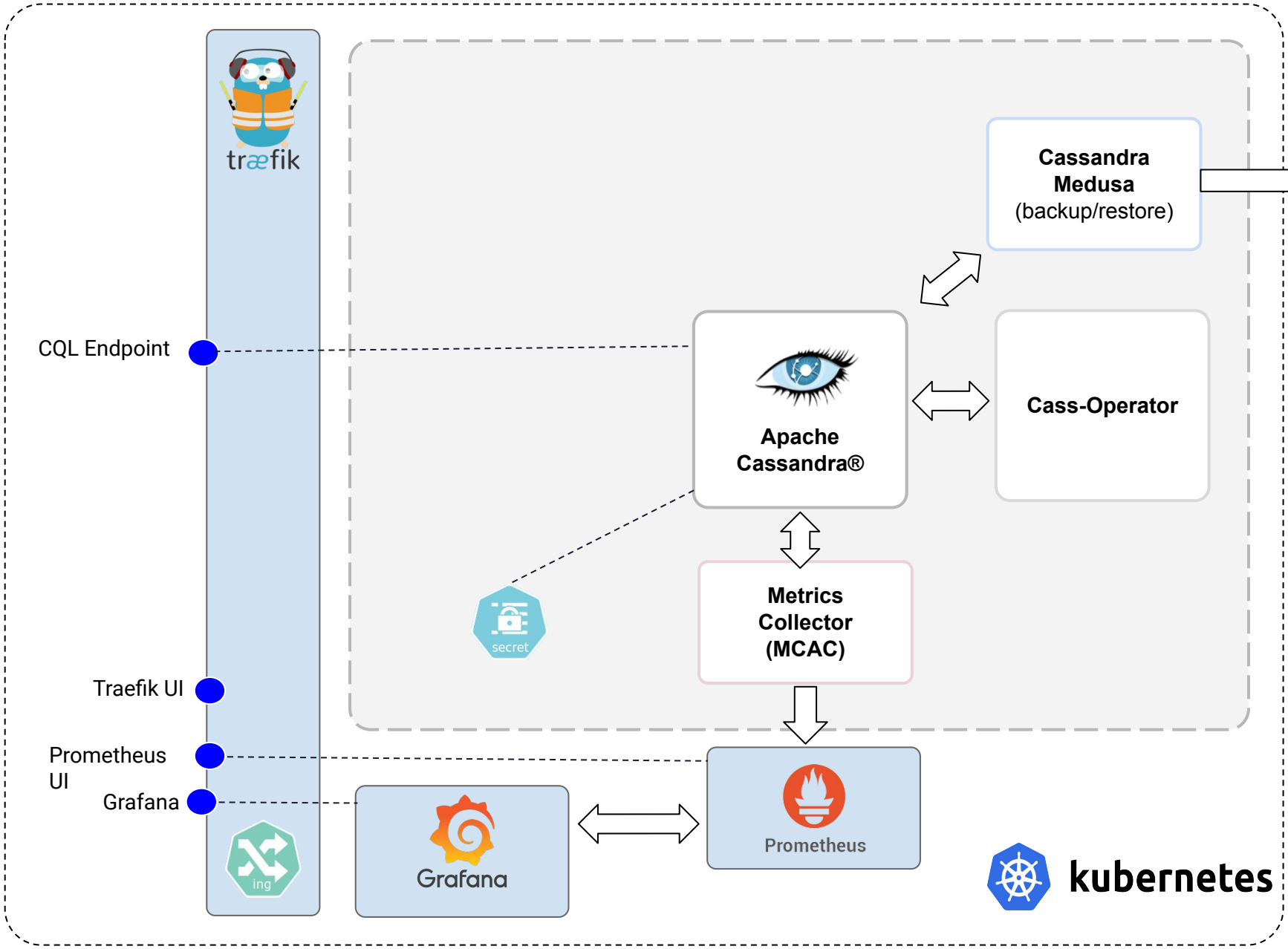
# Backup and restore

Backups are a work in progress but getting close.  In their current form they are executed by kicking off a helm install job.  The following is an example command.

```
helm install k8ssandra-backup datastax/k8ssandra-backup --set name=backup
```

The first backups in development have already happened and changes will be merged soon.  If you want to try them out in beta they will be accessible in the upcoming weeks.

## Backups

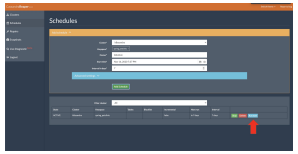

træfik
CQL Endpoint
Traefik UI
Prometheus UI
Grafana
ing
Cassandra Medusa (backup/restore)
MINIO
S3, GCP,...
Apache Cassandra®
Cass-Operator
secret
Metrics Collector (MCAC)
Grafana
Prometheus
kubernetes

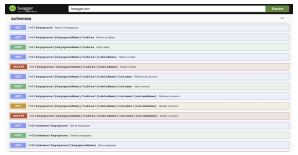
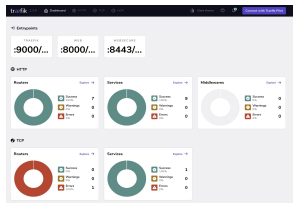
# Demo #7

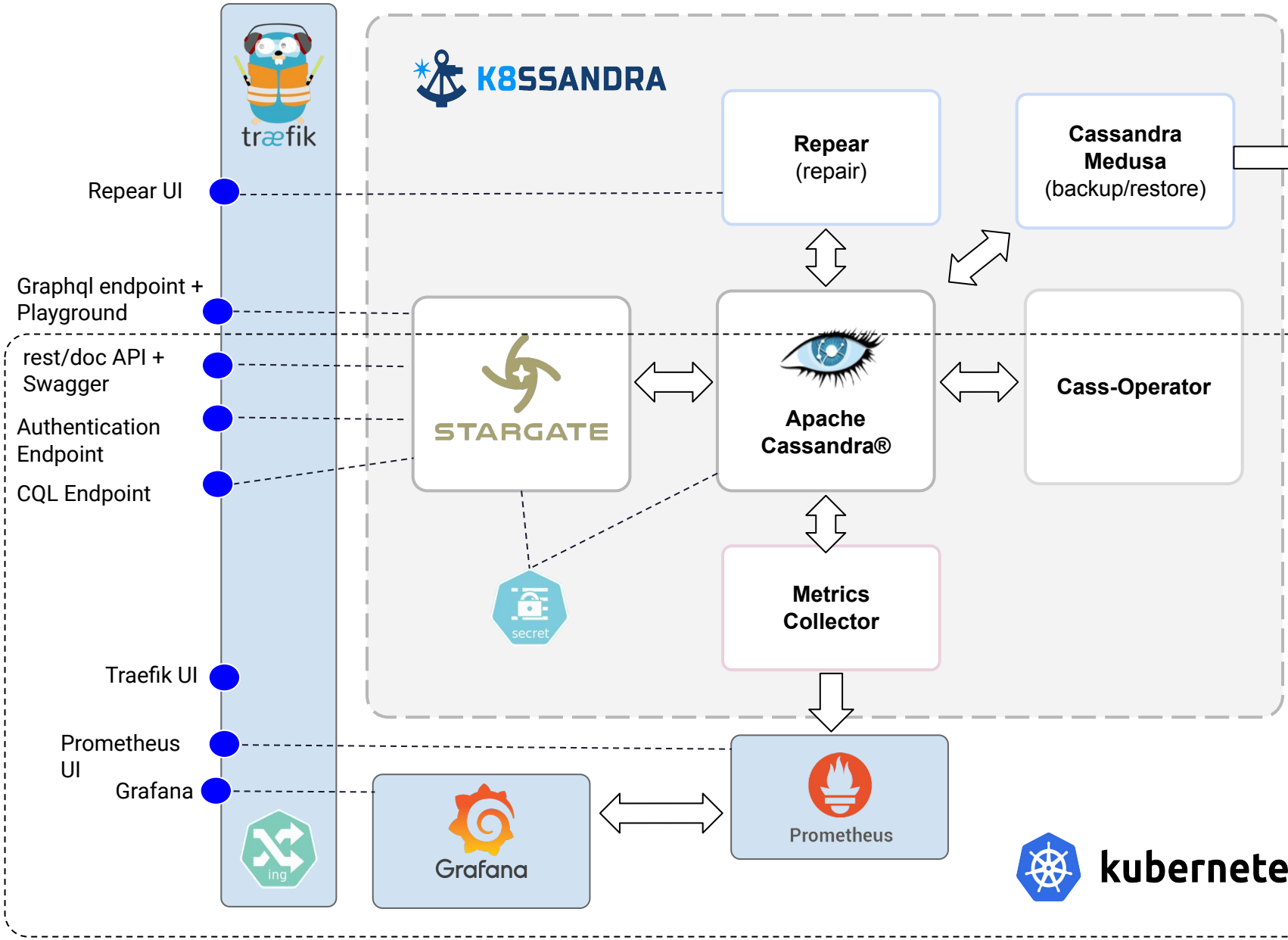
- Backups

# K8ssandra - Apache Cassandra™ meets Kubernetes!

- Housekeeping and Setup
- Cassandra and Kubernetes Basics
- Introducing K8ssandra
- Monitoring Cassandra
- Working with Data
- Scaling Up and Down
- API Access with Stargate
- Running Cassandra Repairs
- Wrapping up

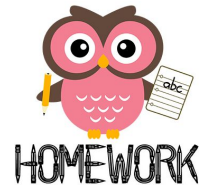




træfik
K8SSANDRA
Repear
(repair)
Cassandra
Medusa
(backup/restore)
MINIO
S3, GCP,...
Repear UI
Graphql endpoint +
Playground
rest/doc API +
Swagger
Authentication
Endpoint
CQL Endpoint
STARGATE
Apache
Cassandra®
Cass-Operator
secret
Metrics
Collector
Traefik UI
Prometheus
UI
Grafana
ing
Grafana
Prometheus
kubernetes


K8SSANDRA

# Kubernetes Workshop – Homework




https://github.com/datastaxdevs/k8ssandra-workshop/wiki

Search or jump to... / Pull requests Issues Marketplace Explore

datastaxdevs / k8ssandra-workshop

forked from DataStax-Academy/kubecon2020

Watch 0 Star 0 Fork 47

<> Code Issues Pull requests Actions Wiki Security Insights Settings

## Home

Edit New Page

Aleks Volochnev edited this page 1 hour ago · 7 revisions

## Practice

The theory alone is not enough. Follow the practice steps to power up your newly learned skills!

### Before starting

To follow along with the hands-on exercises during the workshop, there 2 possibilities:

- Your own [local] installation
- Provided cloud instance (contact Aleks in case of any questions)

Notice that training cloud instances will be available only during the workshop and will be terminated **24 hours later**. If you are in our workshop we recommend using the provided cloud instance, you can relax as we have you covered: prerequisites are installed already. If you are doing this on your own using **your own computer** or your own cloud node, please check the requirements and install the missing tools as explained Here. You need to have a docker-ready machine with at least a 4-

Pages 9

1. Setting Up Cassandra
2. Monitoring Cassandra
3. Working with Data
4. Scaling Up and Down
5. Stargate!
6. Running Repairs
7. Backups

Clone this wiki locally

https://github.com/datastaxdev

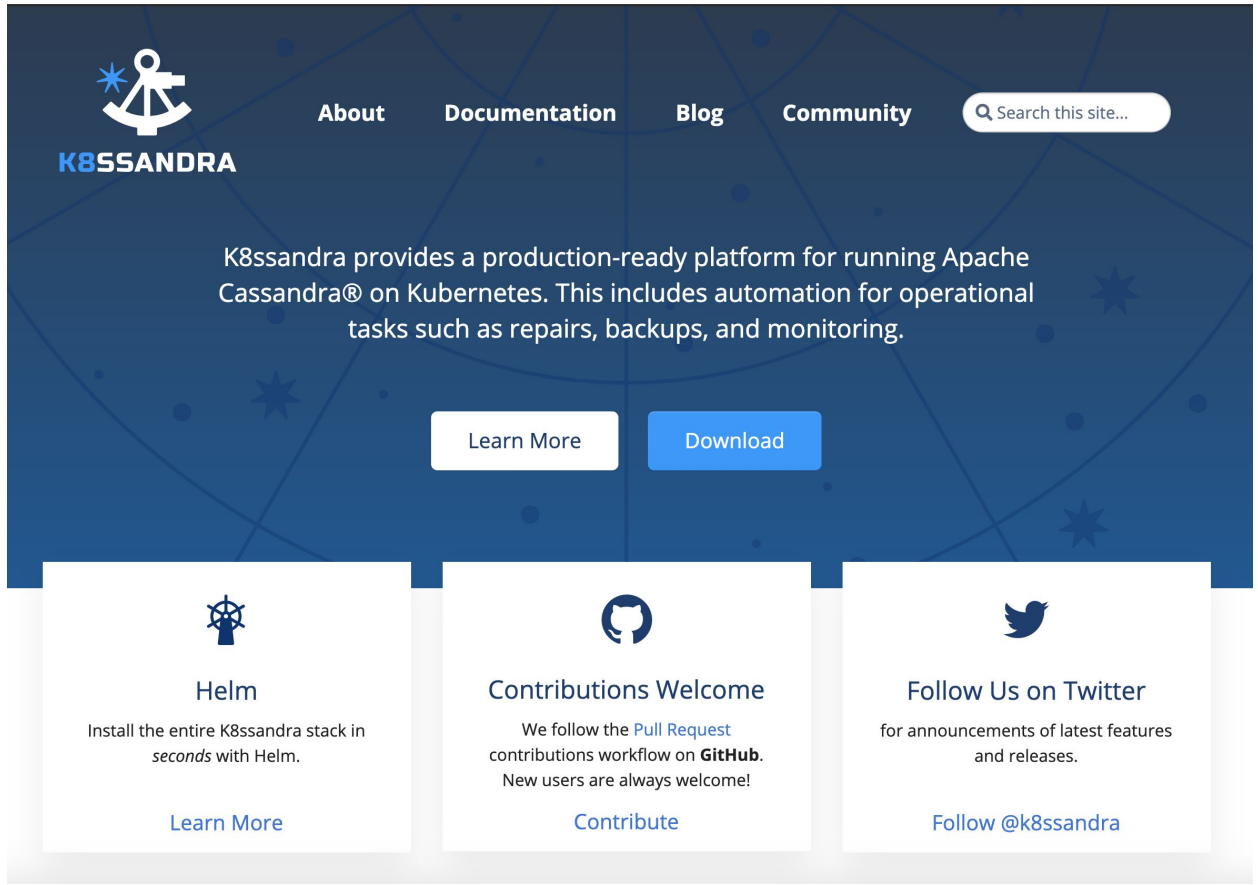
# Join the K8ssandra Community!

@k8ssandra

github.com/k8ssandra

Forums coming soon!

k8ssandra.io



# Cassandra on Kubernetes Certification!

Sign up for news and updates at https://datastax.com/dev/certifications

Cassandra 3.x Developer Certification


Cassandra 3.x Administrator Certification


Cassandra on Kubernetes Certification

**Coming Soon!**

# We'd love to hear from you

**Aleks Volochnev**
Developer Advocate at DataStax

@hadesarchitect

https://calendly.com/aleks-volochnev

**Jeff Carpenter**
Developer Adoption at DataStax

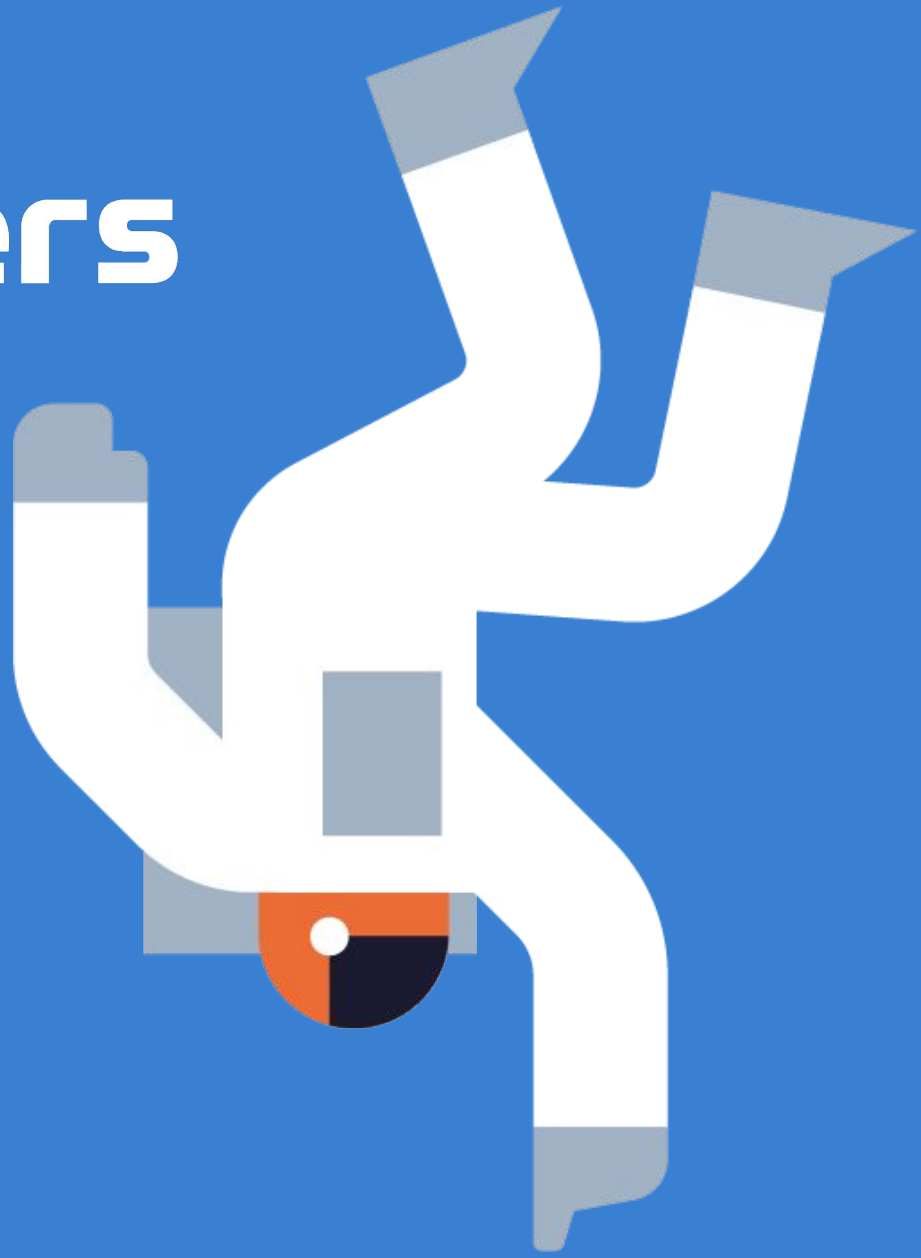
@jeffreyscarpenter

@jscarp

https://calendly.com/jeffrey-carpenter


menti.com
Mentimeter
What is the most desirable feature of your ideal workplace?
Flexible time
Social factor
Big paycheck
Available on the iPhone
App Store
GET IT ON
Google play

# DataStax Developers



## Thank you!

GitHub @hadesarchitect
@jeffreyscarpenter

@hadesarchitect
@jscarp

# DataStax Developers

## Thank you!