



Cloud Native Cassandra

Deploying on Kubernetes with cass-operator



KubeCon



CloudNativeCon

Europe 2020

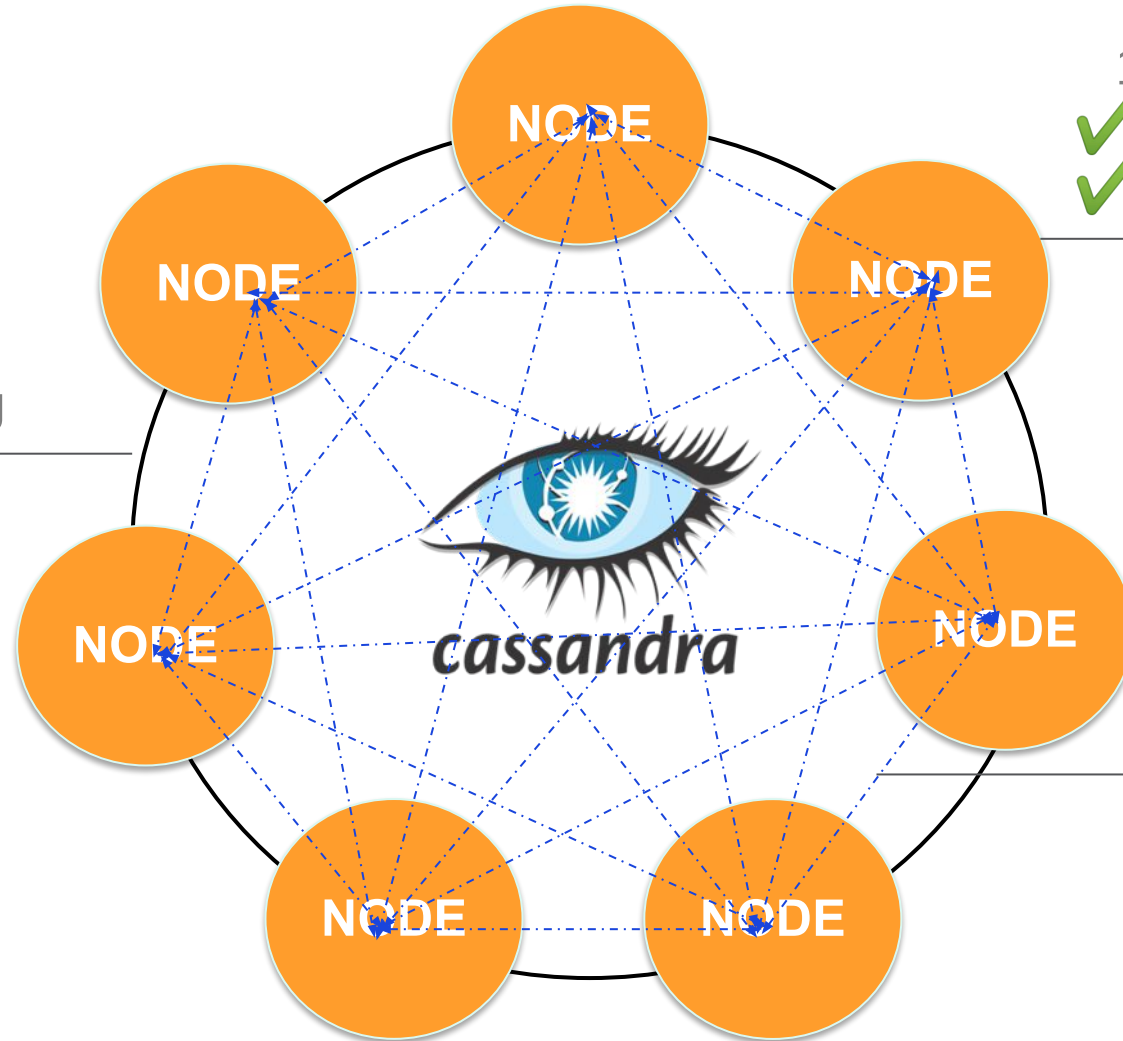


cassandra

Apache Cassandra™ = NoSQL Distributed Database

- 1 Installation = 1 NODE
- ✓ Capacity: ± 1TB
- ✓ Throughput: 3000 Tx/sec/core

DataCenter | Ring



- Communication:
✓ Gossiping

Understanding Use Cases

Scalability

High Throughput
High Volume



Heavy Writes
Heavy Reads



Event Streaming	Log Analytics
Internet of Things	Other Time Series

Availability

Mission-Critical



No Data Loss
Always-on



Caching	Pricing
Market Data	Inventory

Distributed

Global Presence
Workload Mobility



Compliance / GDPR



Banking	Retail
Tracking / Logistics	Customer Experience

Cloud-native

Modern Cloud Applications

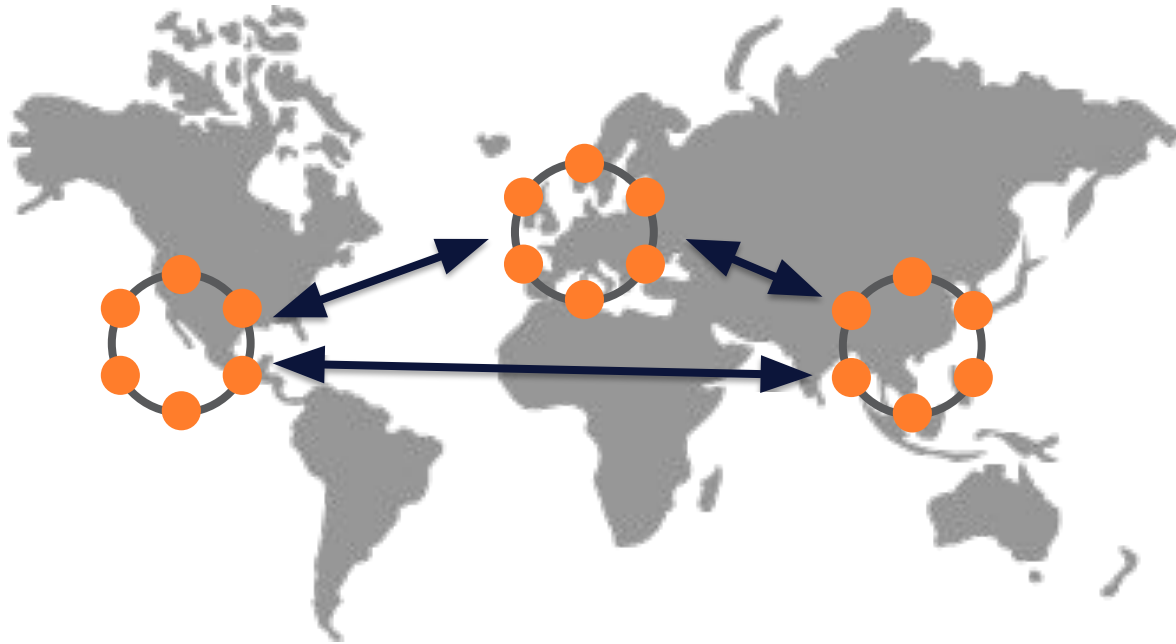


API Layer	Hybrid-cloud
Enterprise Data Layer	Multi-cloud

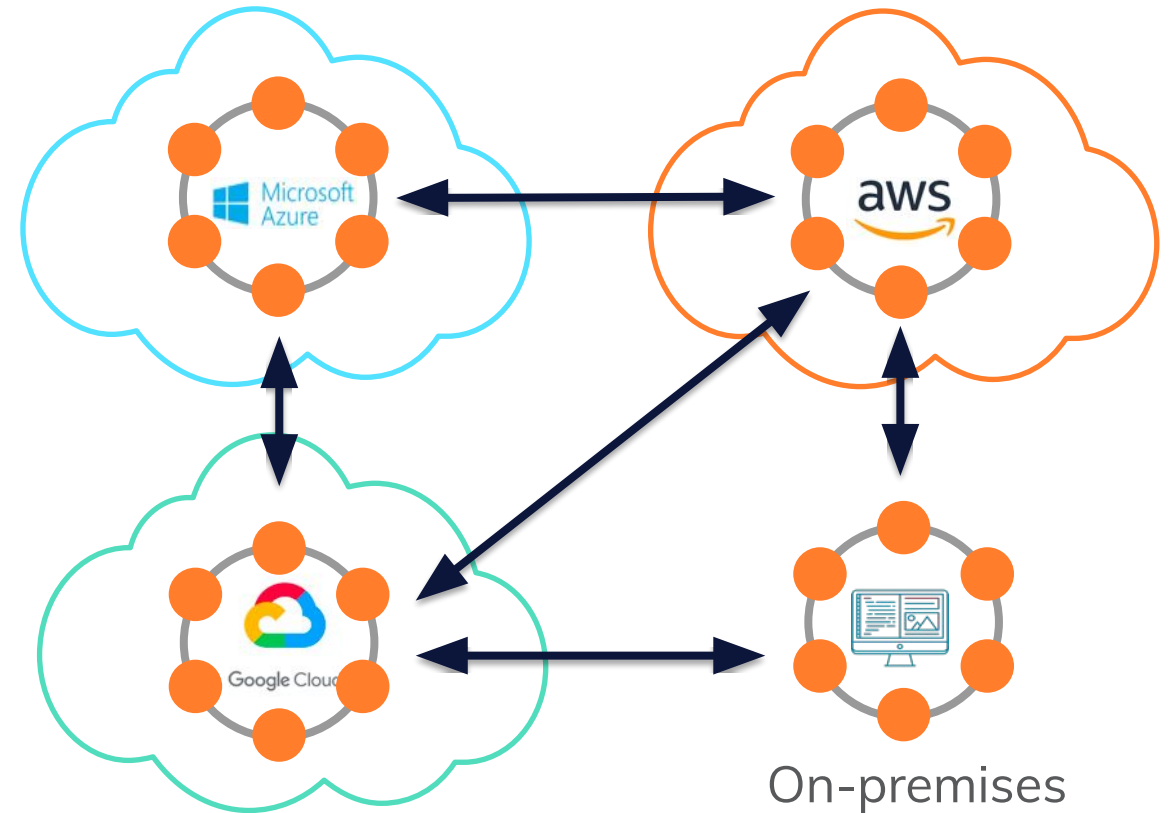


Apache Cassandra™ is a NoSQL Distributed Database cassandra

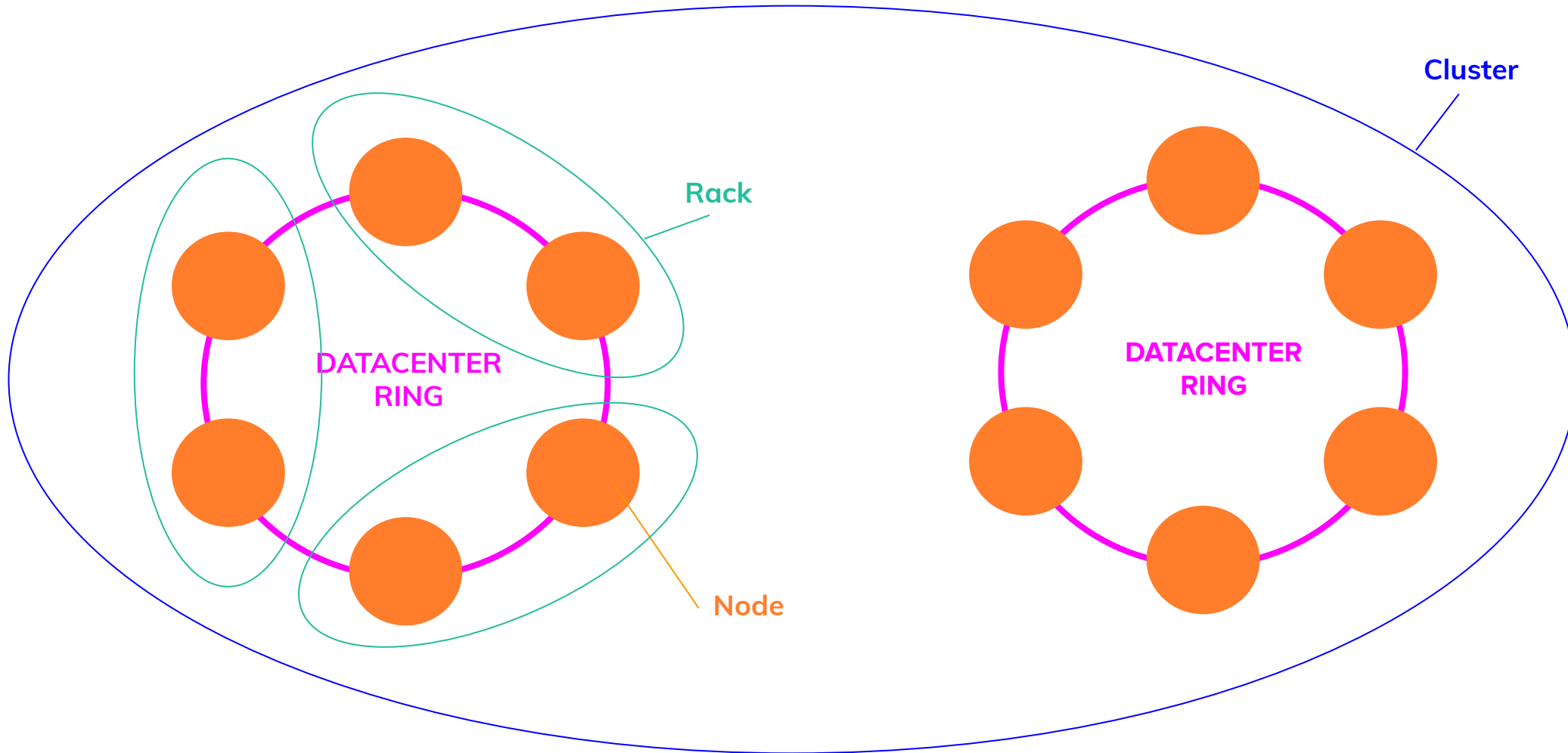
Geographic Distribution



Hybrid-Cloud and Multi-Cloud



Apache Cassandra™ Vocabulary





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

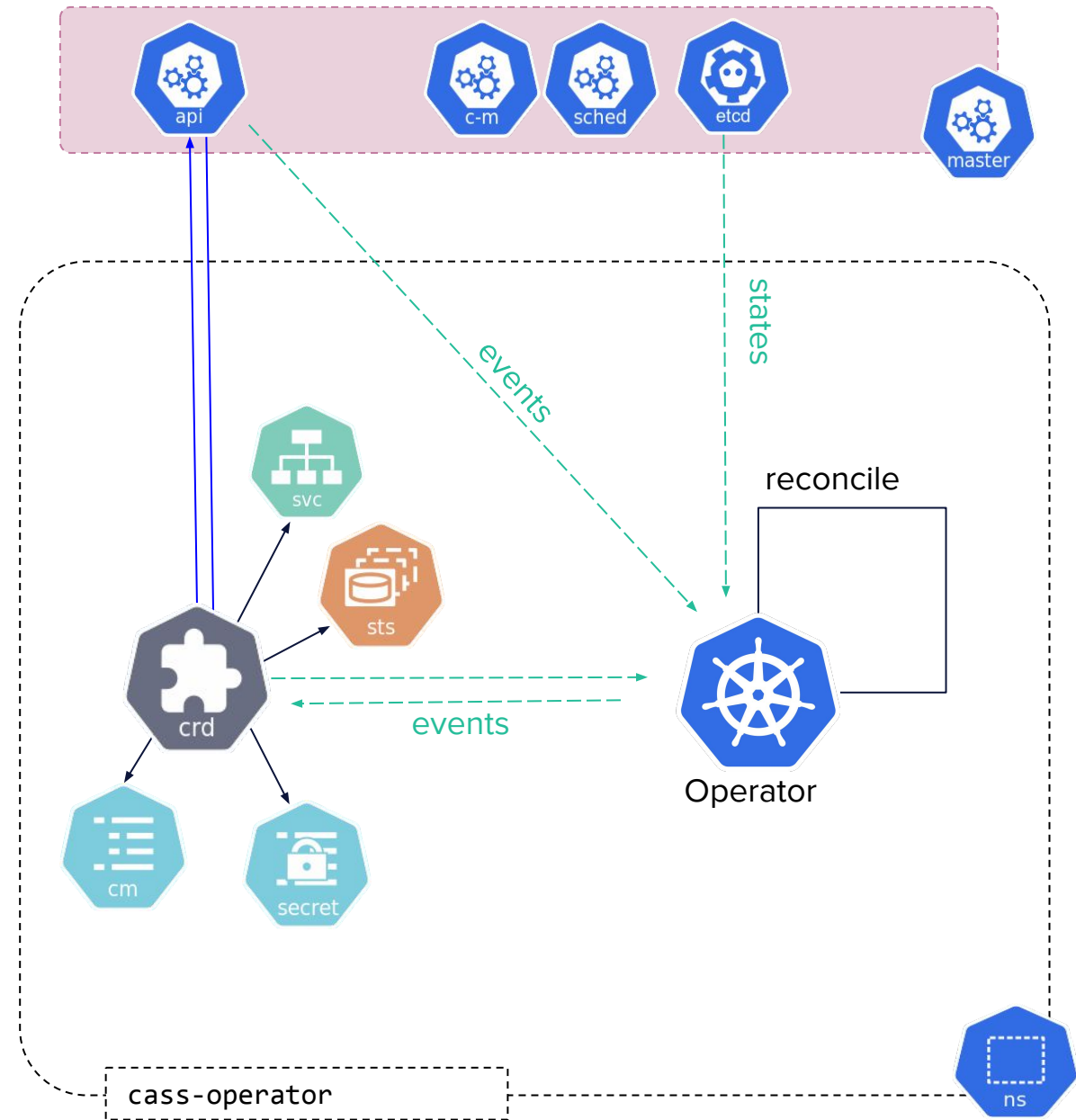
  cassandra-node:
    image: cassandra: 3.11.6
    command: /bin/bash -c "echo 'Waiting for seed node' && sleep 30 && /docker-entrypoint.sh cassandra -f"
    environment:
      - "CASSANDRA_SEEDS=cassandra-seed-node"
    depends_on:
      - "cassandra-seed"
```

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

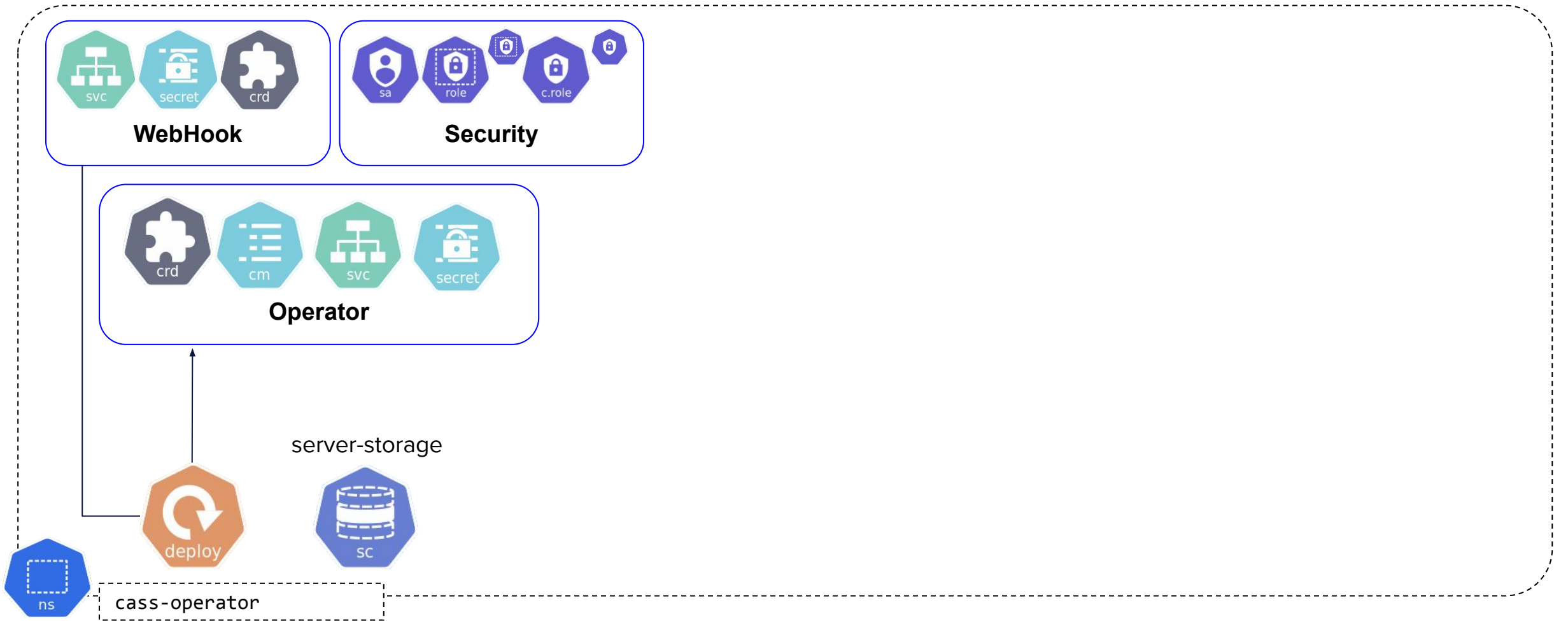
- **Reconcile** CRD instances which states defined within the “spec” attribute.
- **Listen events** and **status evolution** to react accordingly.



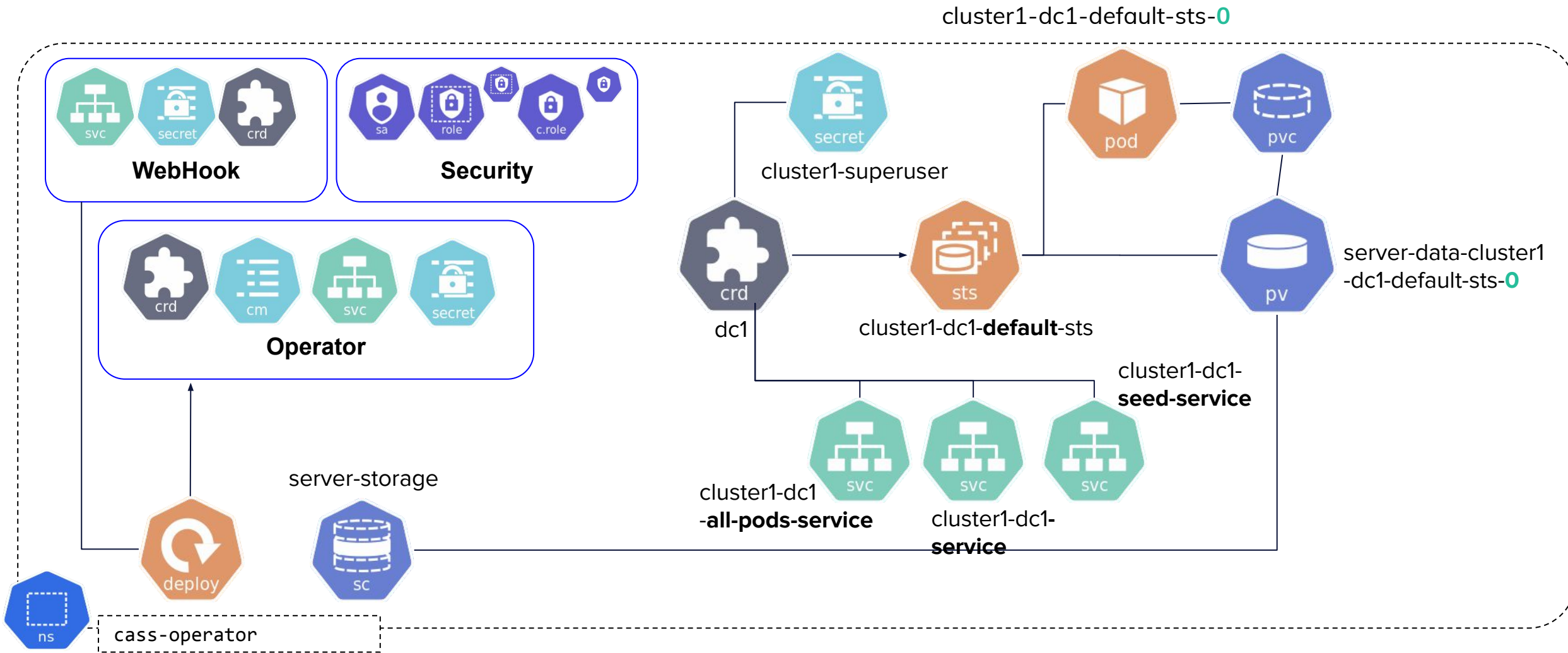


Demo

Installing the Cass Operator Manifest

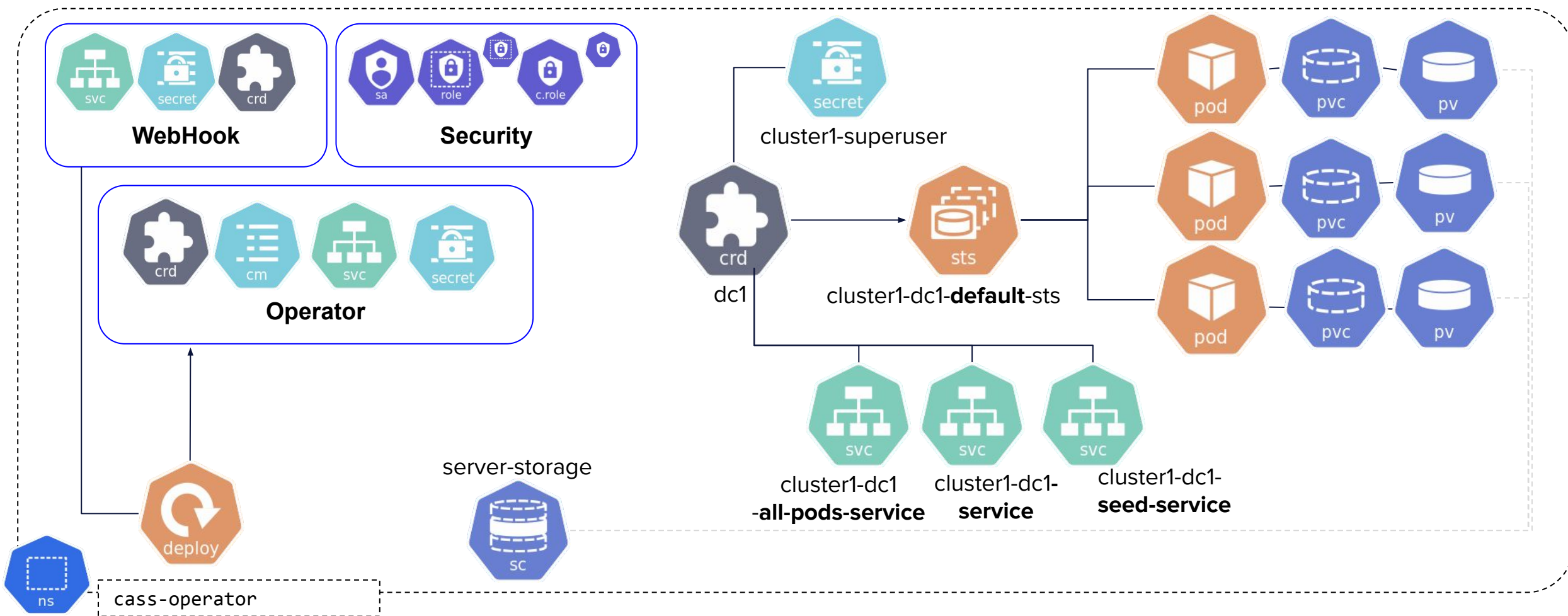


Creating DataCenter dc1

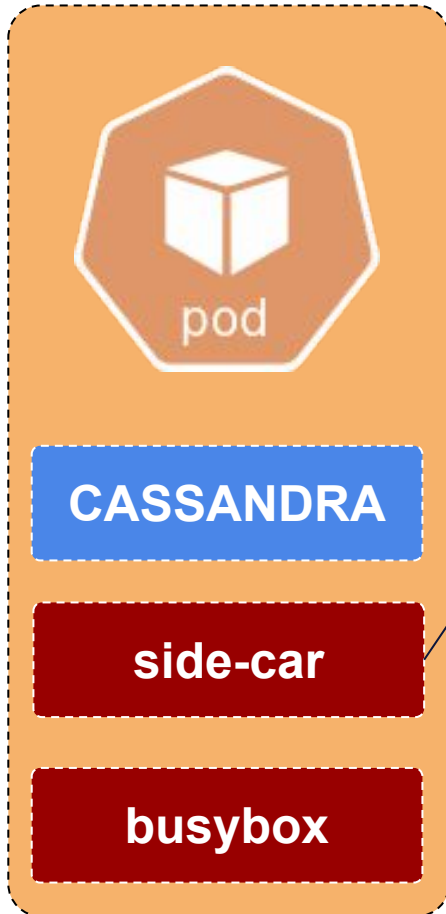


Scale up DataCenter dc1

cluster1-dc1-default-sts-0
cluster1-dc1-default-sts-1
cluster1-dc1-default-sts-2



Our Pods



Cassandra Management API Service

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

Management API for Apache Cassandra ^{0.1} ^{OAS3}

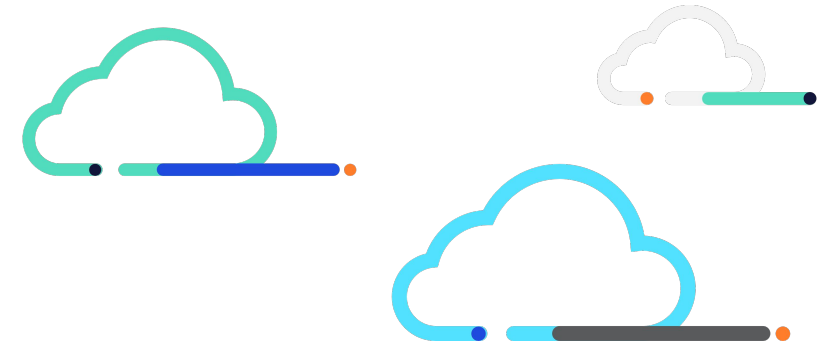
<https://raw.githubusercontent.com/datastax/management-api-for-apache-cassandra/master/management-api-server/doc/openapi.json>

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



Thank you !

To play the demo



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