

Week 5

Cassandra in Kubernetes



Cassandra Cloud-Native Workshop Series

Building Cloud-Native apps with Cassandra Expertise



Cassandra & Kubernetes



DataStax Developer Special Unit !



David
Jones-Gilardi



Eric
Zietlow



Erick
Ramirez



Cédrick
Lunven



Bettina
Swynnerton



Jack
Fryer



Aleksandr
Volochnev



MATERIALS



bit.ly/CassandraWorkshopMaterials

Developer Workshop Series **Week5**



What we will cover:

- Housekeeping
- Kubernetes Basics
- Kubernetes Operators
- Cass Operator in Deep
- Dashboarding + Grafana | Prometheus
- Resources



MATERIALS



bit.ly/CassandraWorkshopMaterials

Livestreams

YouTube



Twitch

The central column displays five screenshots illustrating the DataStax ecosystem:

- YouTube:** A thumbnail for a video titled "3 HOURS" featuring a speaker.
- Astra Cluster Management:** A screenshot of the Astra web interface showing cluster metrics and database details.
- DataStax Academy:** A screenshot of the DataStax Academy course "cassandra-workshop-series" on GitHub.
- GitHub:** A screenshot of the GitHub repository "DataStax-Academy/cassandra-workshop-series" showing code commits.
- Cassandra Workshop Series:** A screenshot of the "Cassandra Workshop Series" landing page.

Astra.datastax.com



Live Questions

YouTube



Discord



Materials & Help



GitHub



Gitpod



MATERIALS → bit.ly/CassandraWorkshopMaterials

menti.com

58 55



Available on the iPhone
App Store

GET IT ON
Google play

Developer Workshop Series **Week5**



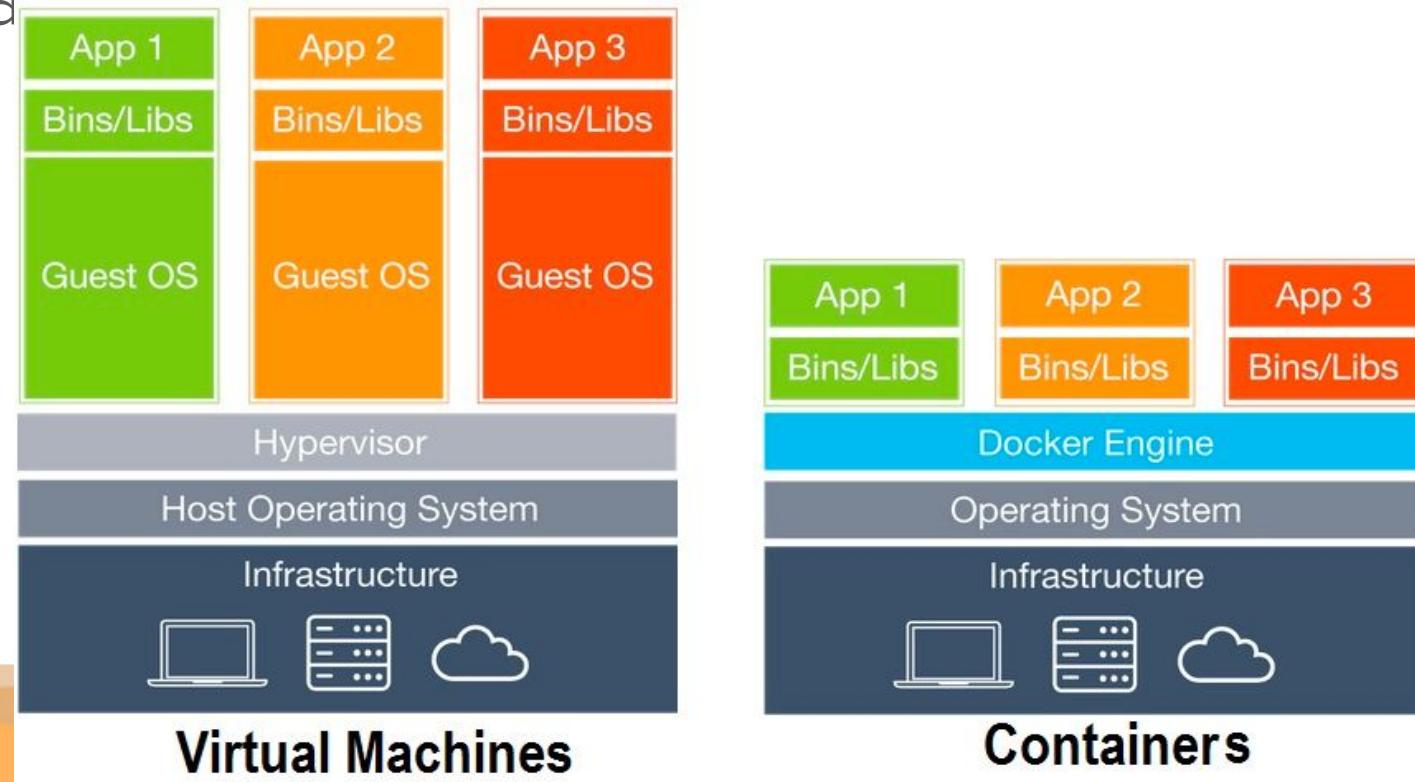
- Housekeeping
- **Kubernetes Basics**
- Kubernetes Operators
- Cass Operator in Deep
- Dashboarding + Grafana | Prometheus
- Resources

MATERIALS

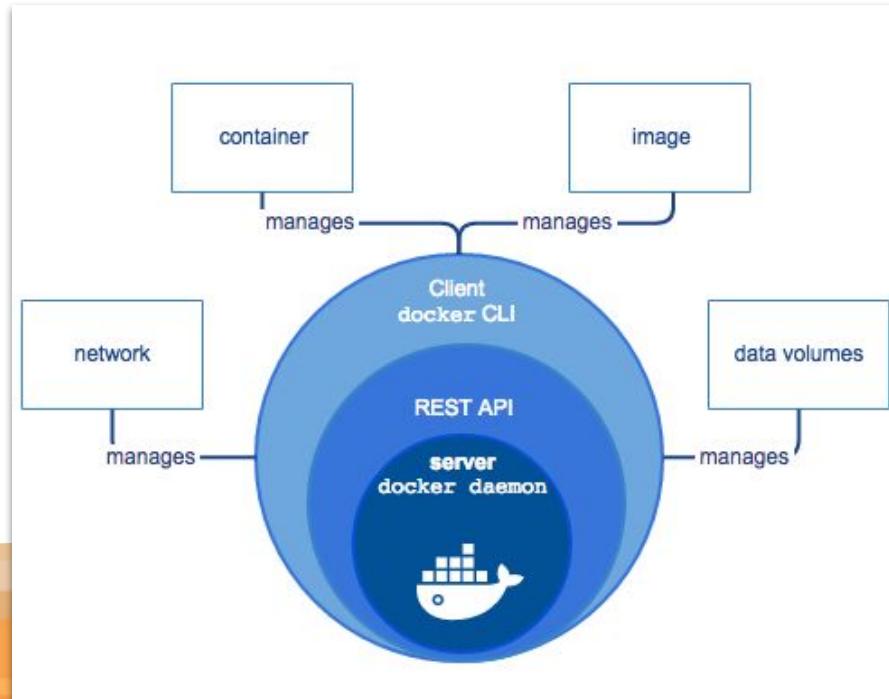
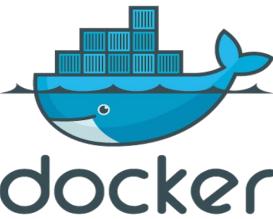


bit.ly/CassandraWorkshopMaterials

Virtual Machines vs. Containers



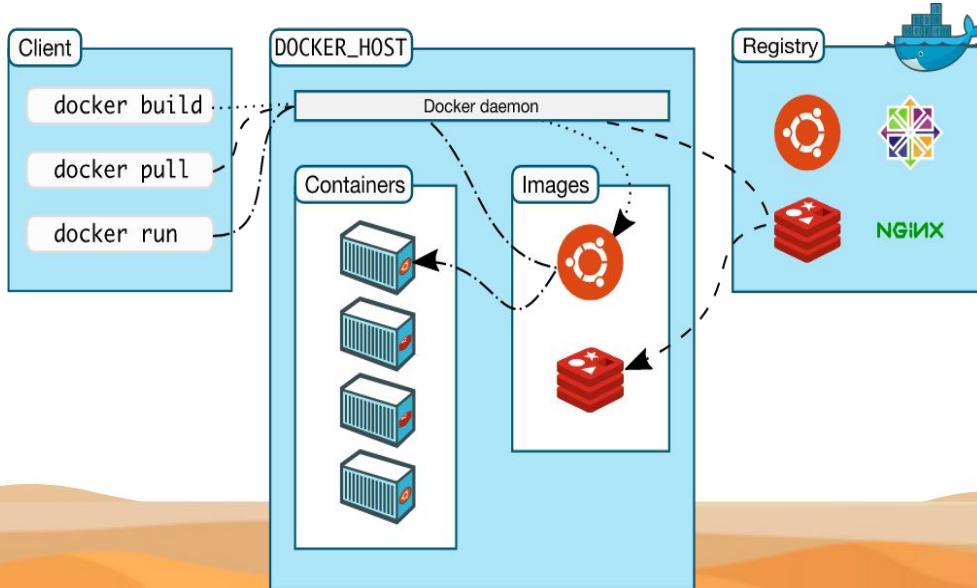
MATERIALS → bit.ly/CassandraWorkshopMaterials



Docker Engine is Client-server application with :

- A server which is a type of long-running program called a daemon process (**the `dockerd` command**).
- **A REST API** which specifies interfaces that programs can use to talk to the daemon and instruct it what to do.
- **A command line interface (CLI)** client (the `docker` command).

Docker Architecture



Some Docker client commands

- We move immutable **images** (build/pull/push)
- Docker Daemon start containers based on images (run)
 - *Adding Env Var*
 - *Adding Storage (volumes)*
 - *Adding Network*
 - *Exposing port*
- We interact with **running containers** (ps, exec...)

MATERIALS



bit.ly/CassandraWorkshopMaterials



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
```

MATERIALS



bit.ly/CassandraWorkshopMaterials



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"
```

MATERIALS



bit.ly/CassandraWorkshopMaterials





kubernetes

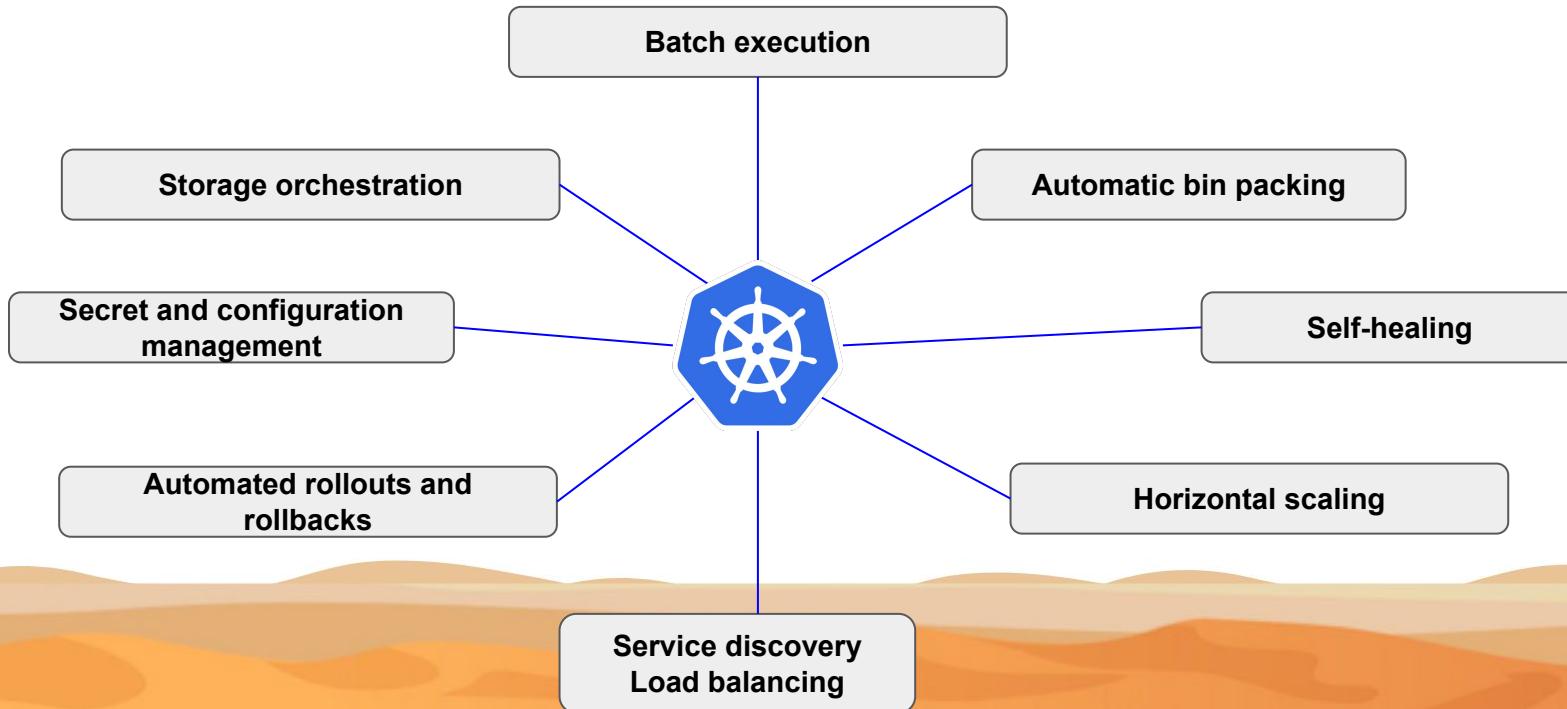
*"Kubernetes is an open-source system for
automating deployment, scaling, and management
of containerized applications."*

MATERIALS



bit.ly/CassandraWorkshopMaterials

Kubernetes Features



Kubernetes Infrastructure



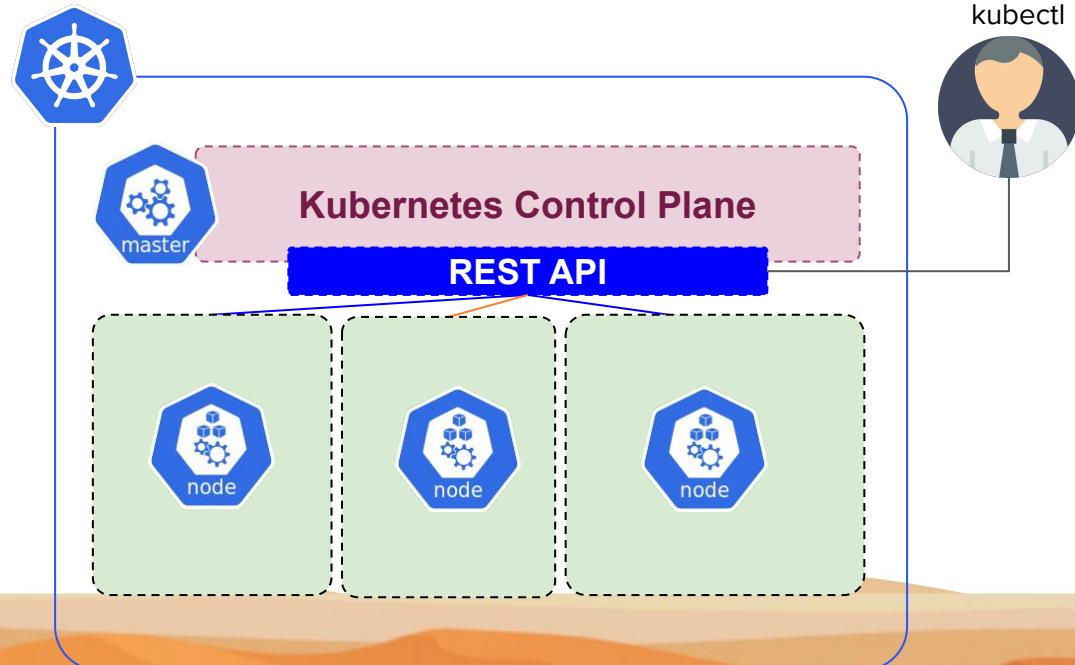
Cluster:
Kubernetes cluster.



Master:
Kubernetes Control Plane.



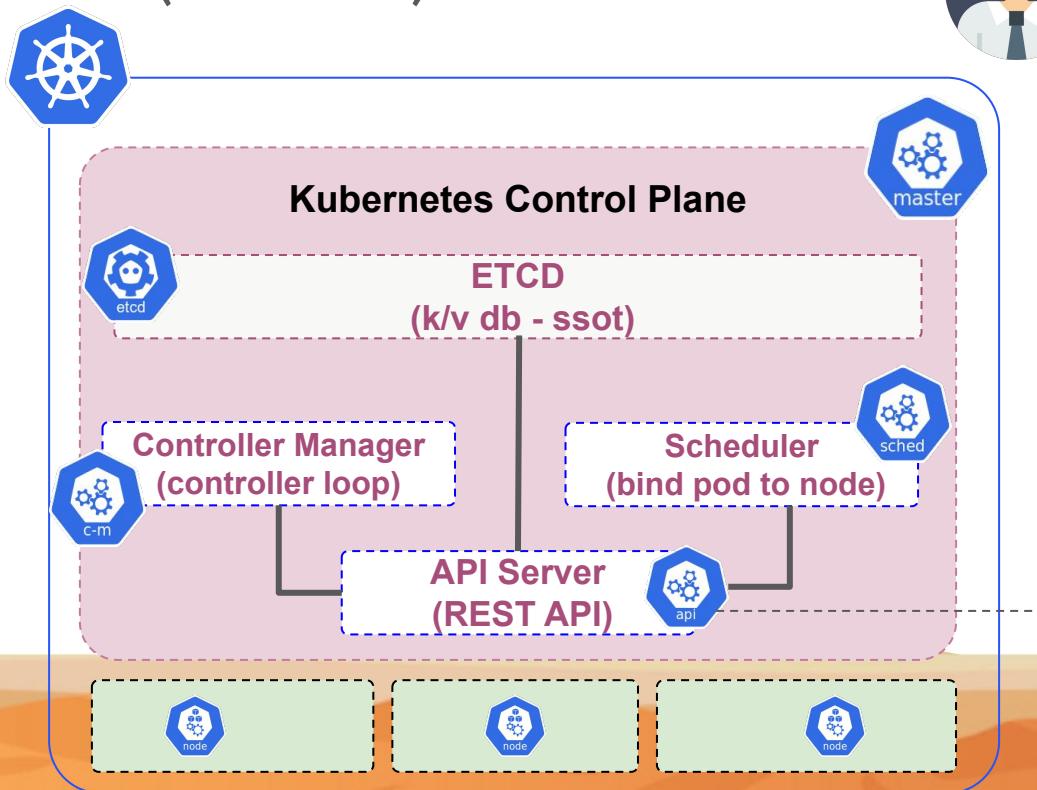
Node:
Worker machine in Kubernetes cluster.



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.



MATERIALS



bit.ly/CassandraWorkshopMaterials

Kubernetes Worker (Node, Minions)



Kubelet:

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



Kube-proxy

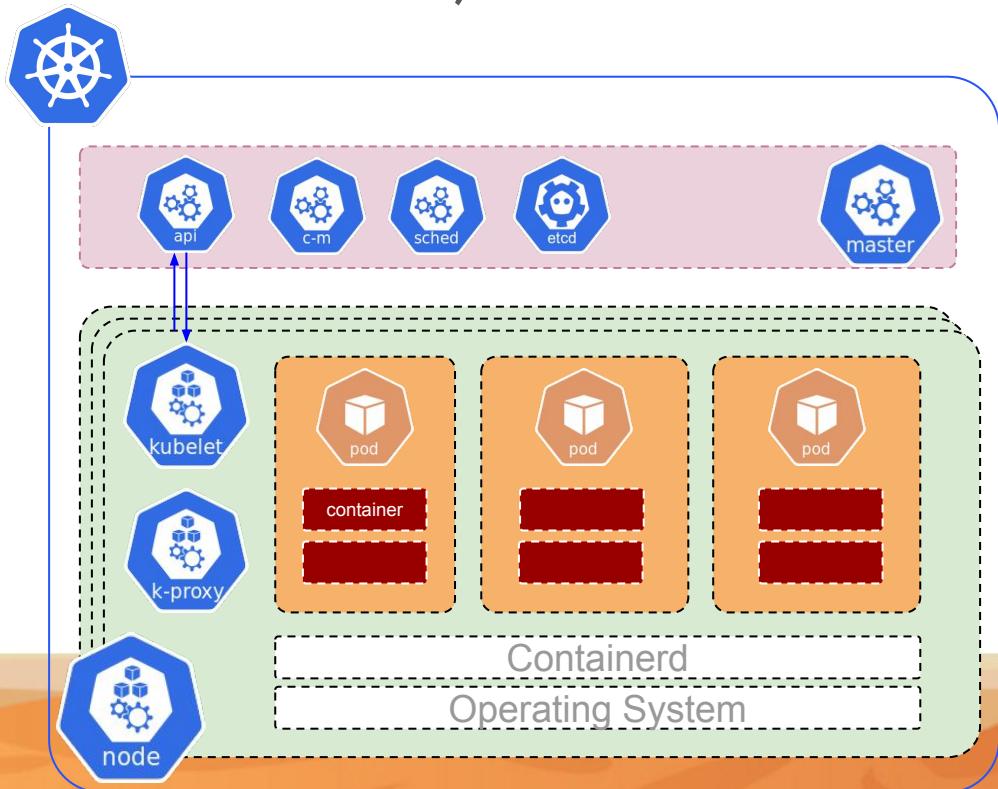
The Kubernetes network proxy runs on each node. This reflects services as defined in the Kubernetes API on each node.



POD

Collection of containers that can run on a host.

This resource is created by clients and scheduled onto hosts.

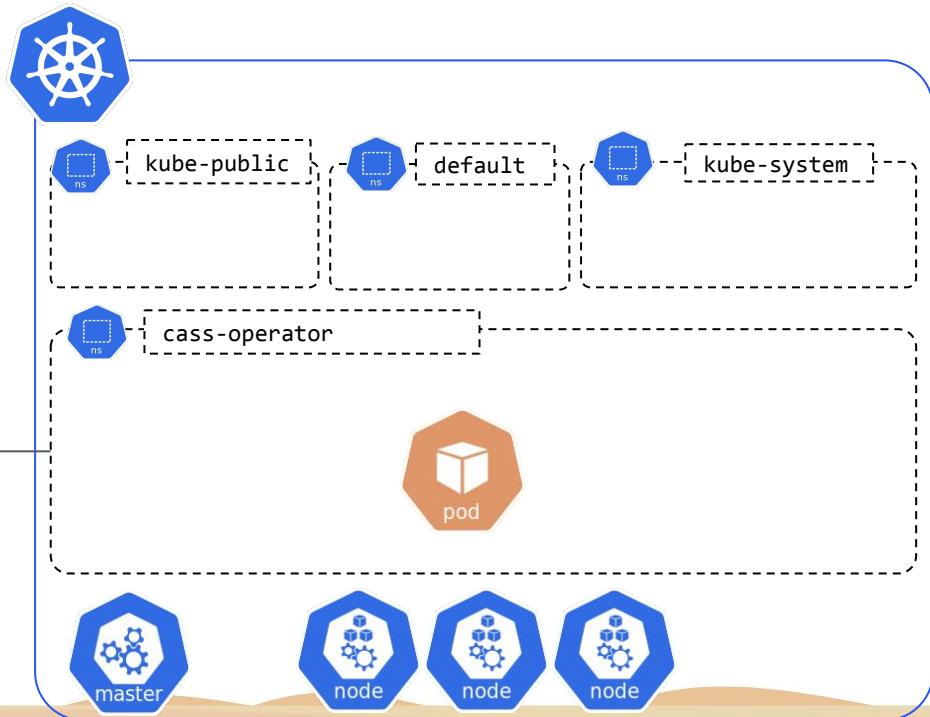


Kubernetes Namespace



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.

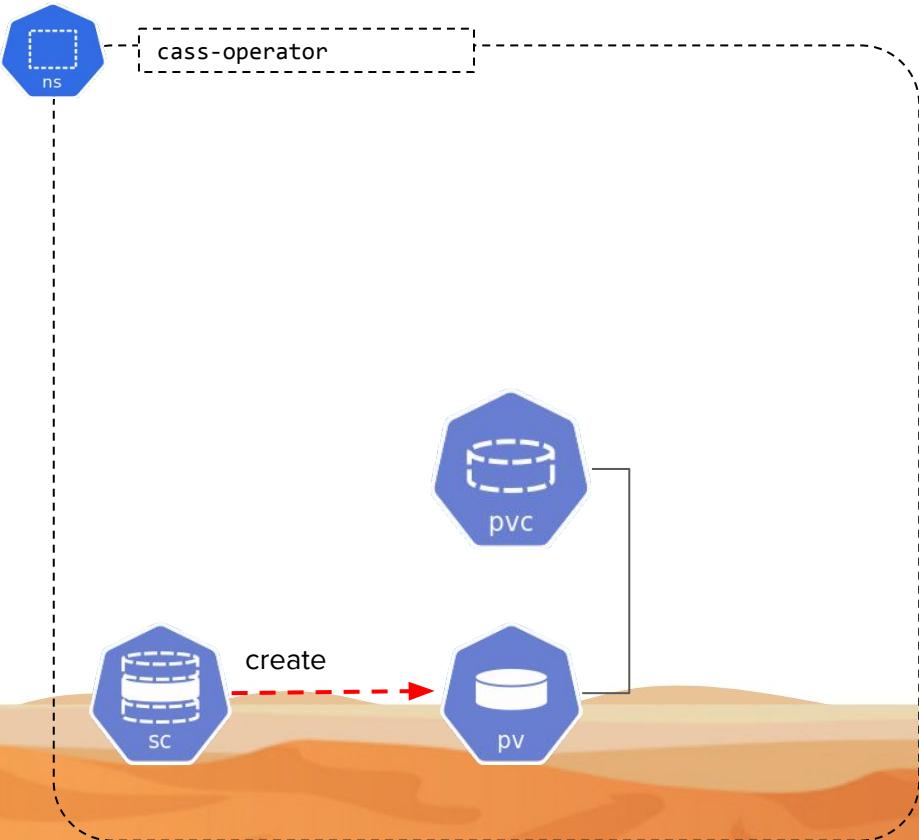


PersistentVolumeClaim:

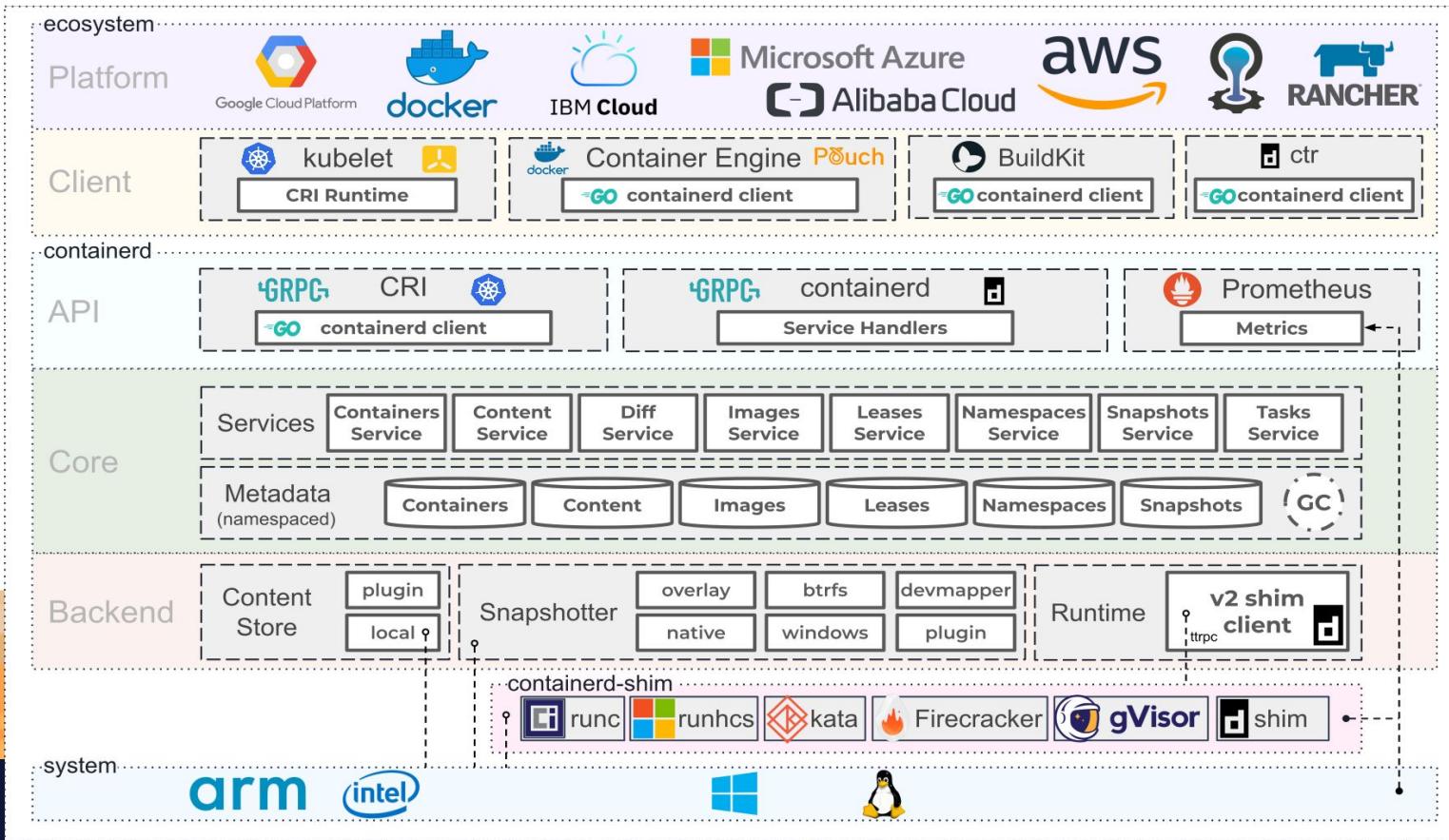
PersistentVolumeClaim is a user's request for and claim to a persistent volume.

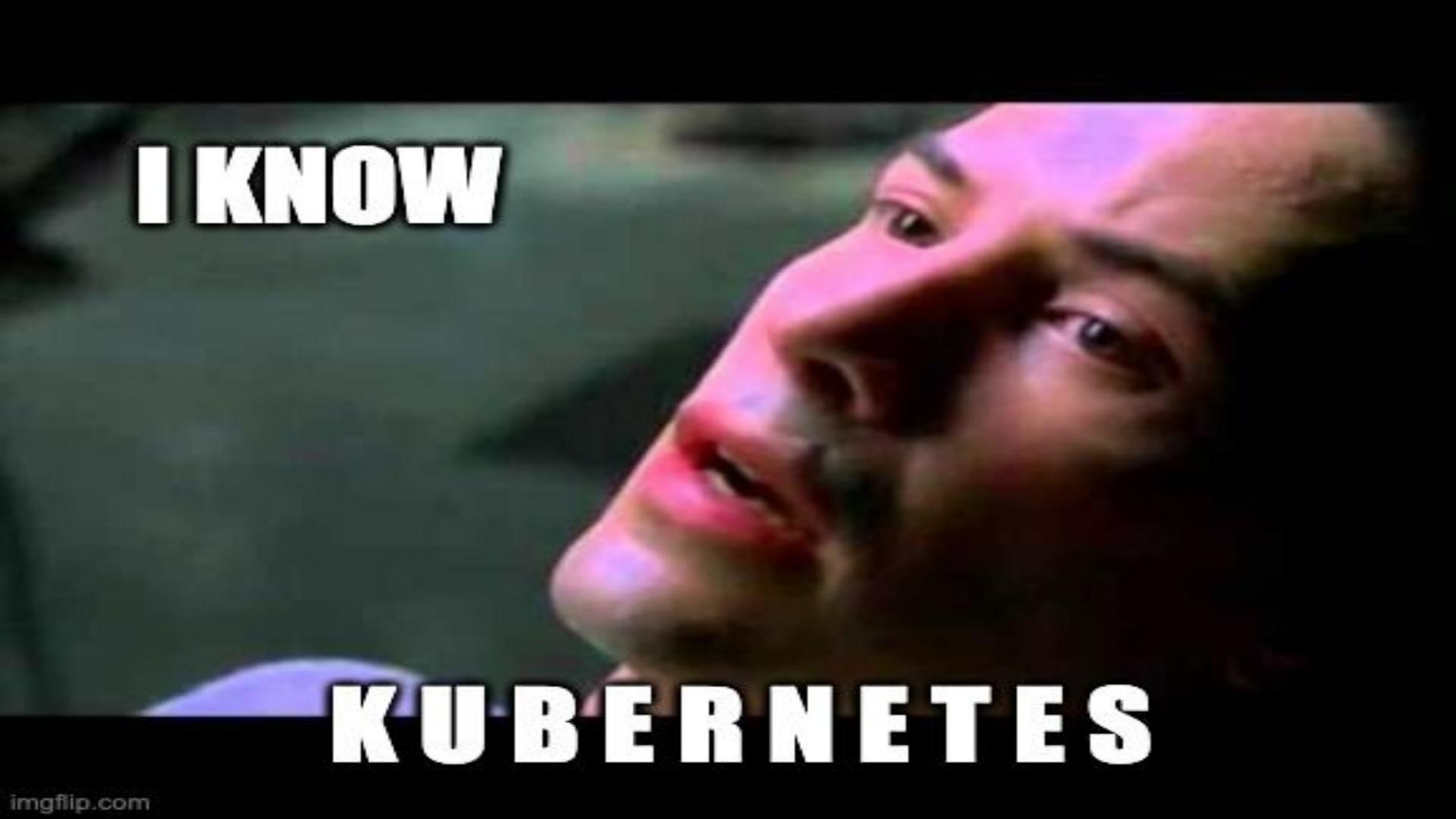


StorageClass: StorageClass describes the parameters for a class of storage for which *PersistentVolumes* can be dynamically provisioned.



Open specification they said...





I KNOW

KUBERNETES

Kubernetes Distributions

Clouds



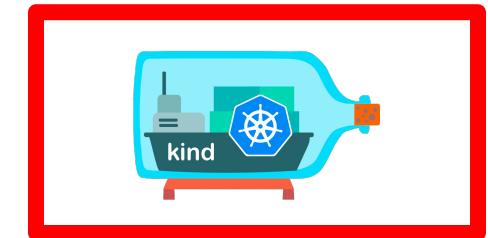
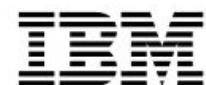
- Azure Kubernetes Service (AKS)



Amazon
EKS



- Google Kubernetes Engine (GKE)



MATERIALS → bit.ly/CassandraWorkshopMaterials



YOUR LAPTOP

Your Laptop



docker



kubernetes



CLOUD INSTANCE



Cloud Env



kubernetes



docker



Exercise

Installation and prerequisites

<https://github.com/DataStax-Academy/kubernetes-workshop-online/blob/master/setup-your-cluster/README.MD>

5. Install Kind

kind (kind)is a tool for running local Kubernetes clusters using Docker container "nodes". kind was primarily designed for testing Kubernetes itself, but may be used for local development or CI. Please refer to [Reference Documentation](#) for more detailed instructions.

 : To install on windows please download the [executable](#) and place it on the PATH. You can also use [Chocolatey](#) very clever package manager for windows.

```
choco install kind
```

 : To install on MAC OS please use the following [homebrew](#) commands:

```
brew install kubectl
```

 To install on linux (centOS) you can use the following commands

```
curl -Lo ./kind https://github.com/kubernetes-sigs/kind/releases/download/v0.7.0/kind-$uname-amd64  
chmod +x ./kind  
sudo mv ./kind /usr/local/bin/kind
```

Check that the installation is successful. Starting for now all command will be the same on each platform, as a such we will keep providing a single command. We will mark with a blue book the command () and a green book () to show expected result.

MATERIALS

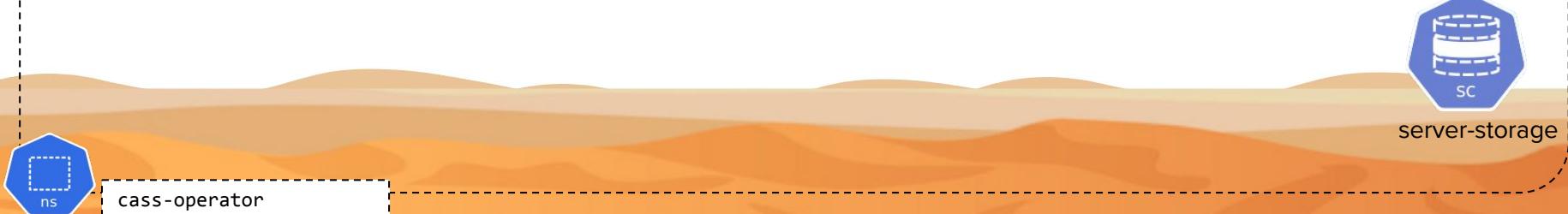


bit.ly/CassandraWorkshopMaterials



Setup your cluster

Nothing Here :)



Developer Workshop Series **Week5**



- Housekeeping
- Kubernetes Basics
- **Kubernetes Operators**
- Cass Operator in Deep
- Dashboarding + Grafana | Prometheus
- Resources

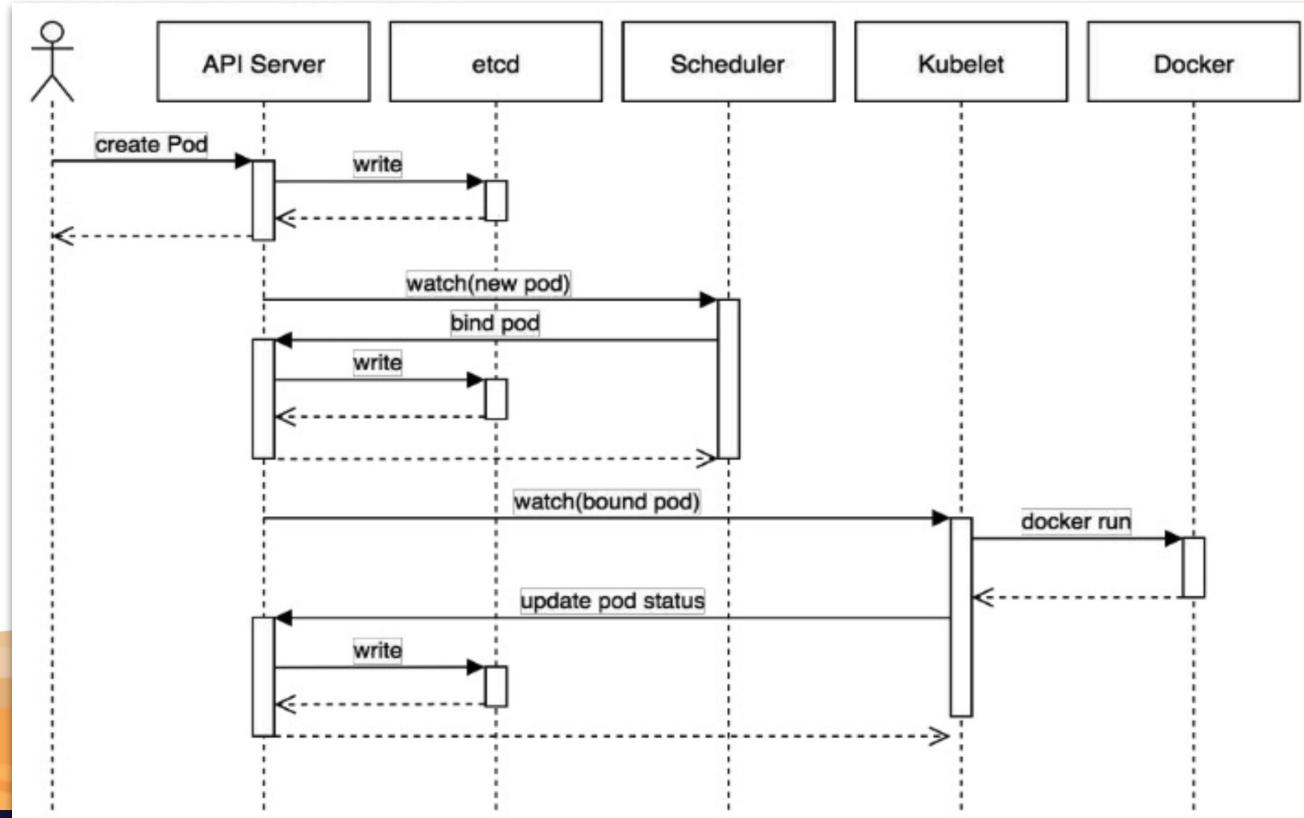
MATERIALS



bit.ly/CassandraWorkshopMaterials



Pod Lifecycle



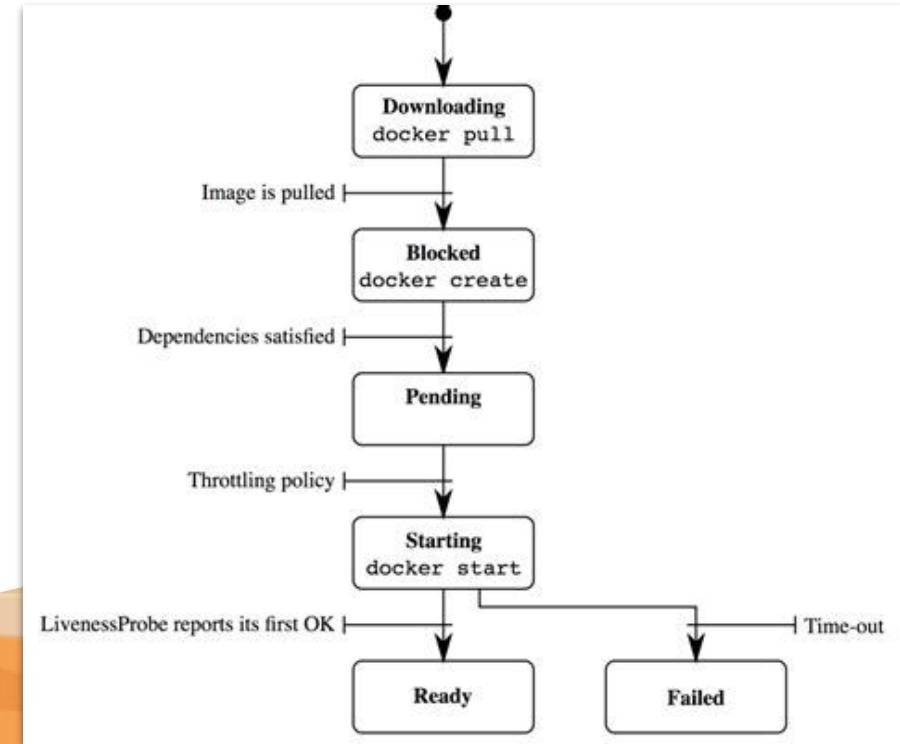
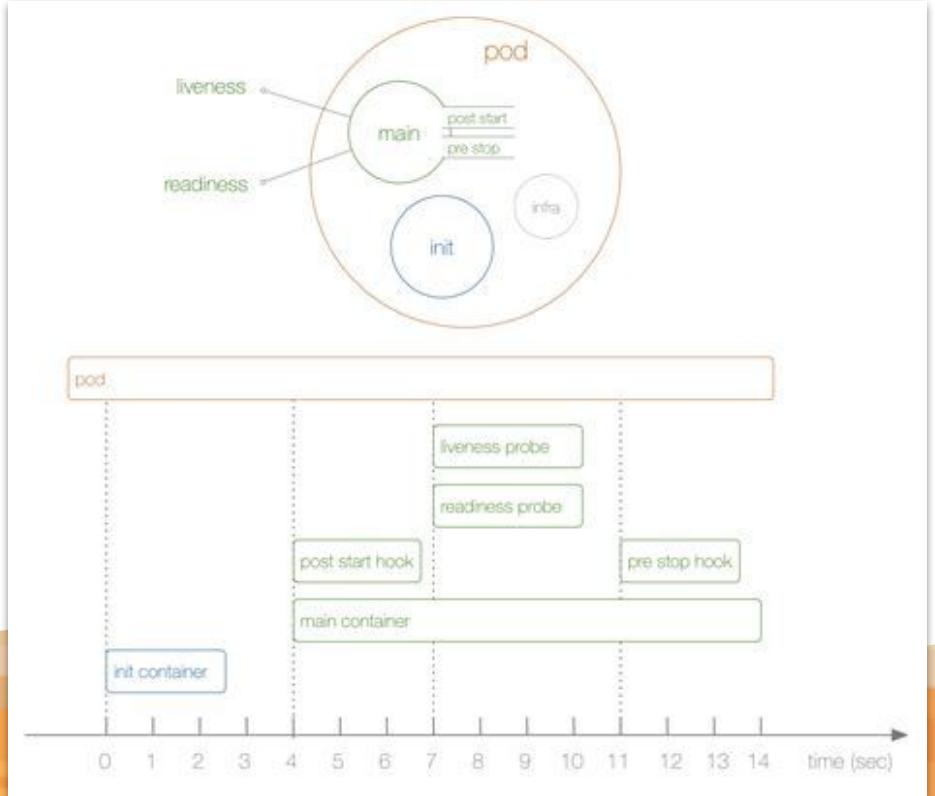
MATERIALS



bit.ly/CassandraWorkshopMaterials



Pod Lifecycle

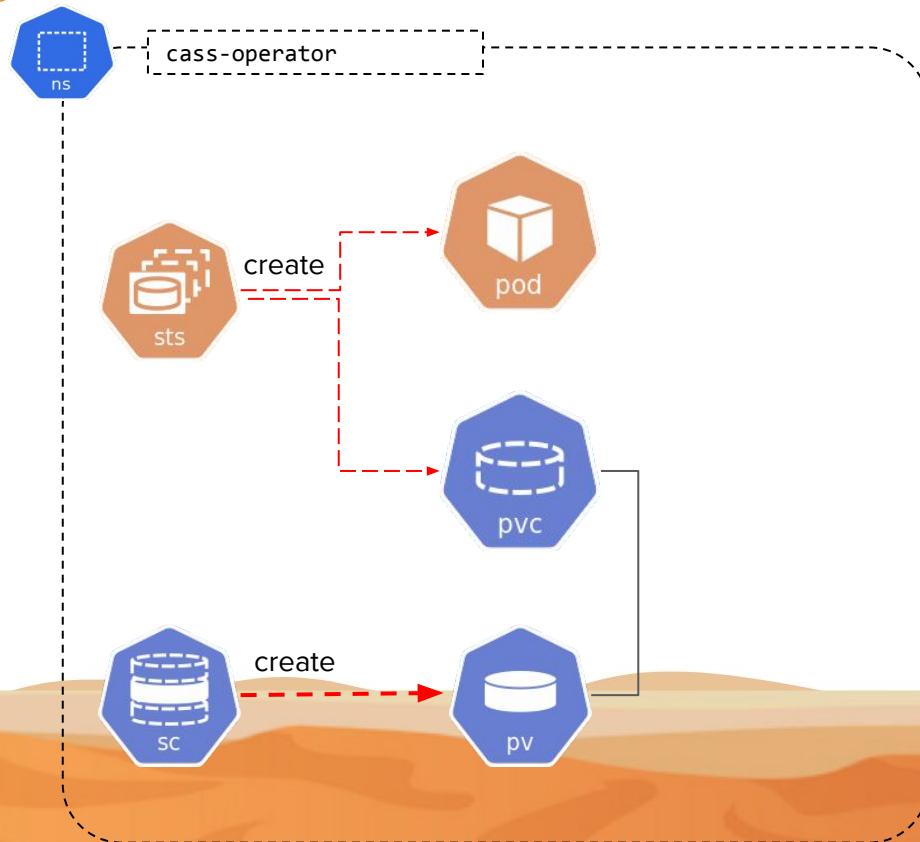


MATERIALS → bit.ly/CassandraWorkshopMaterials

K8s Primitives : StatefulSet



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



MATERIALS



bit.ly/CassandraWorkshopMaterials

K8s Primitives : Service



Ingress is a collection of rules that allow inbound connections to reach the endpoints defined by a backend.



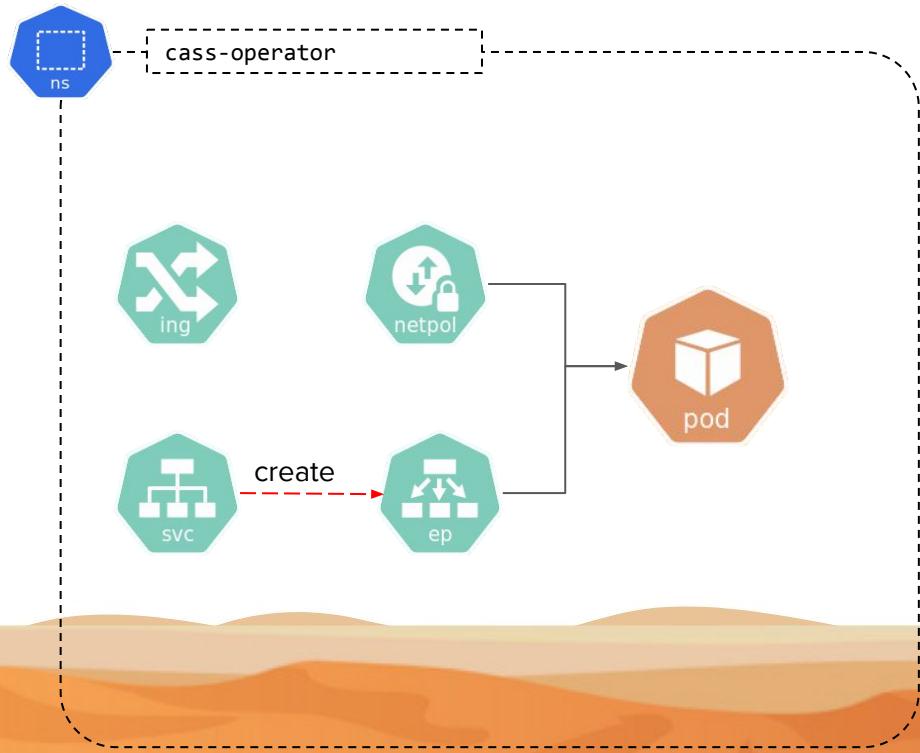
Service is a named abstraction of software service with ports to listen on and selector to determine which pods will answer requests.



EndPoint is a collection of endpoints that implement the actual service..



NetworkPolicy: Describes what network traffic is allowed for a set of Pods.



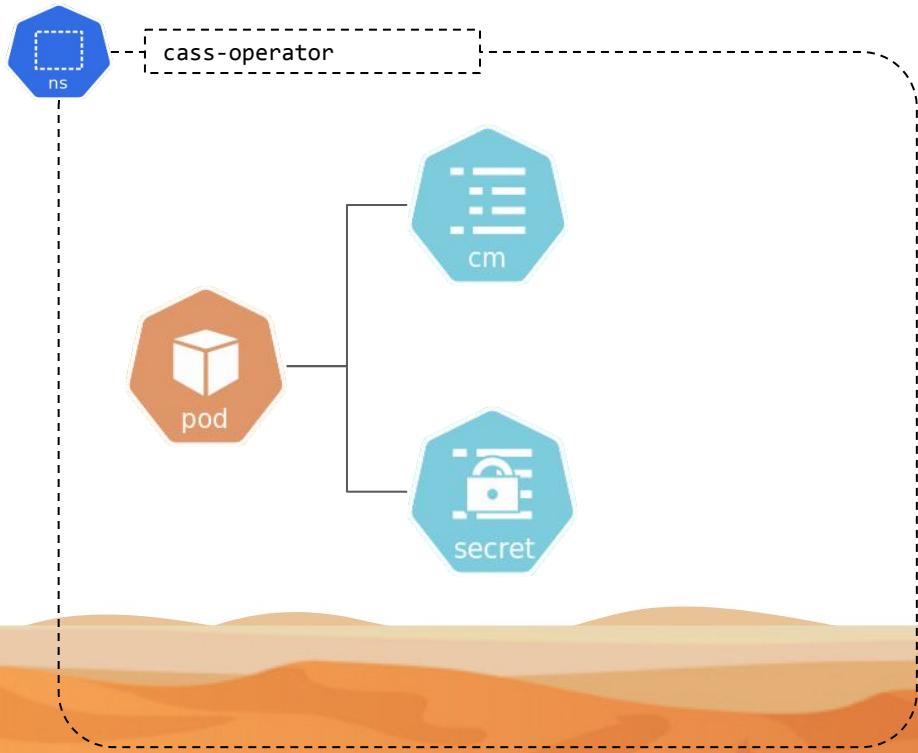
K8s Primitives : Configuration



ConfigMap: ConfigMap holds configuration data for pods to consume..



Secret: Secret holds secret data of a certain type..



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

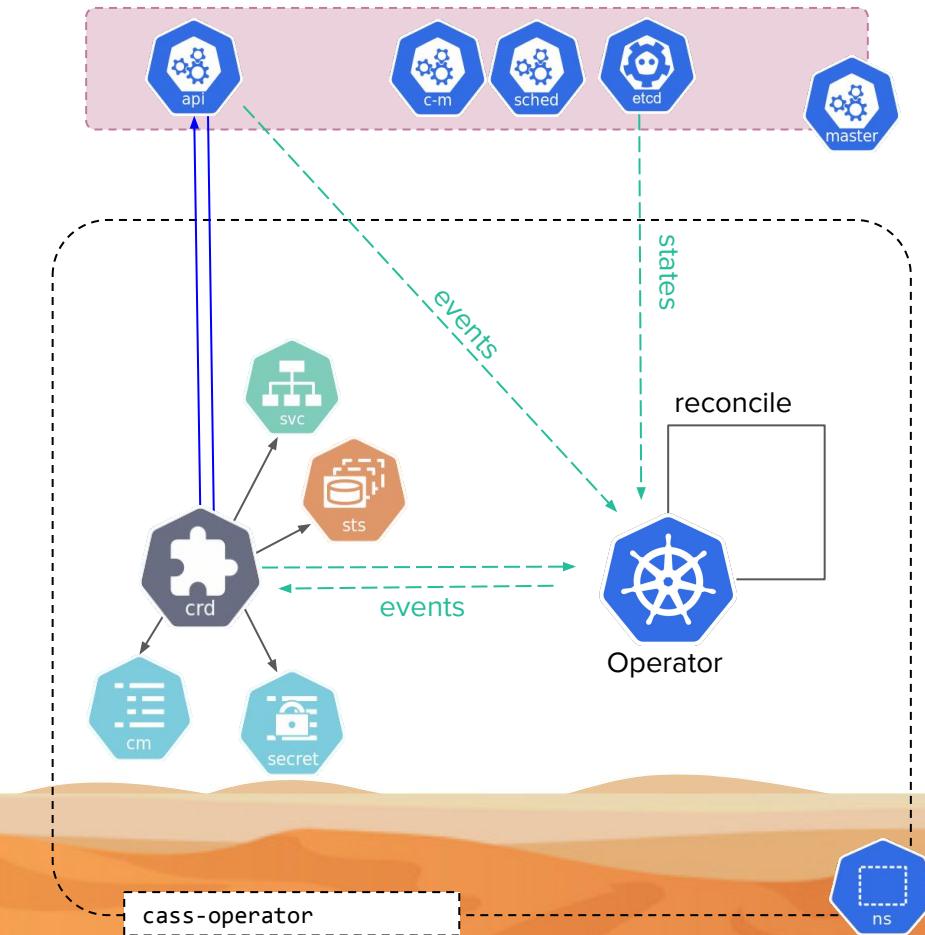
```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: crontabs.stable.example.com
spec:
  # group name to use for REST API: /apis/<group>/<version>
  group: stable.example.com
  # list of versions supported by this CustomResourceDefinition
  versions:
    - name: v1
      # 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:
        openAPI3Schema:
          type: object
          properties:
            spec:
              type: object
              properties:
                cronSpec:
                  type: string
                image:
                  type: string
                replicas:
                  type: integer
  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.



menti.com

58 55



Developer Workshop Series **Week5**



What we will cover:

- Housekeeping
- Kubernetes Basics
- Kubernetes Operators
- Cass Operator in Deep
- Dashboarding + Grafana | Prometheus
- Resources



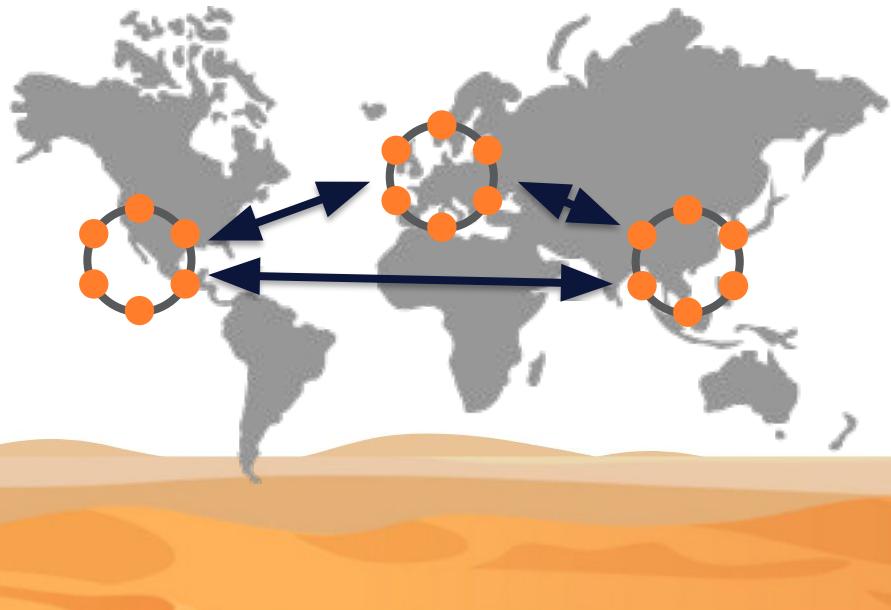
MATERIALS



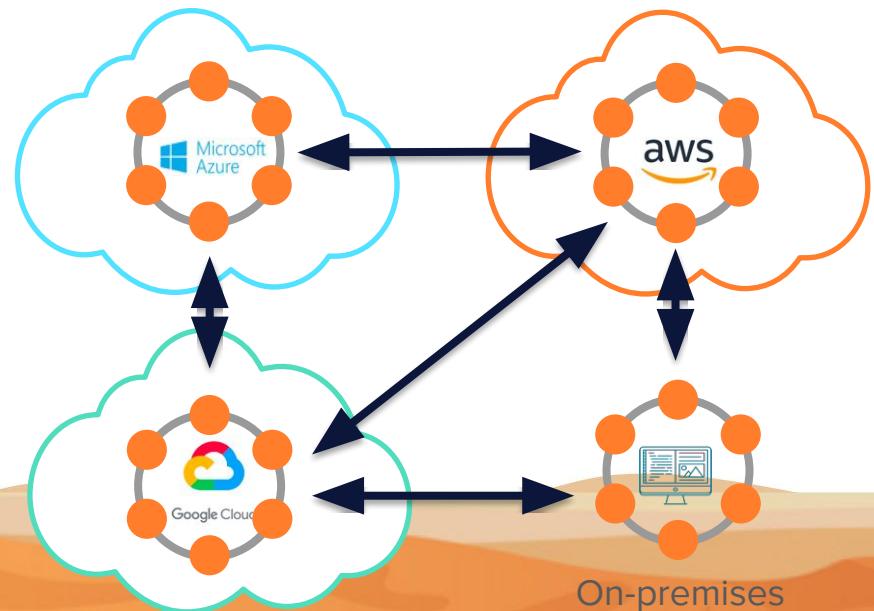
bit.ly/CassandraWorkshopMaterials

Apache Cassandra™ is a **NoSQL Distributed** Database

Geographic Distribution



Hybrid-Cloud and Multi-Cloud



On-premises

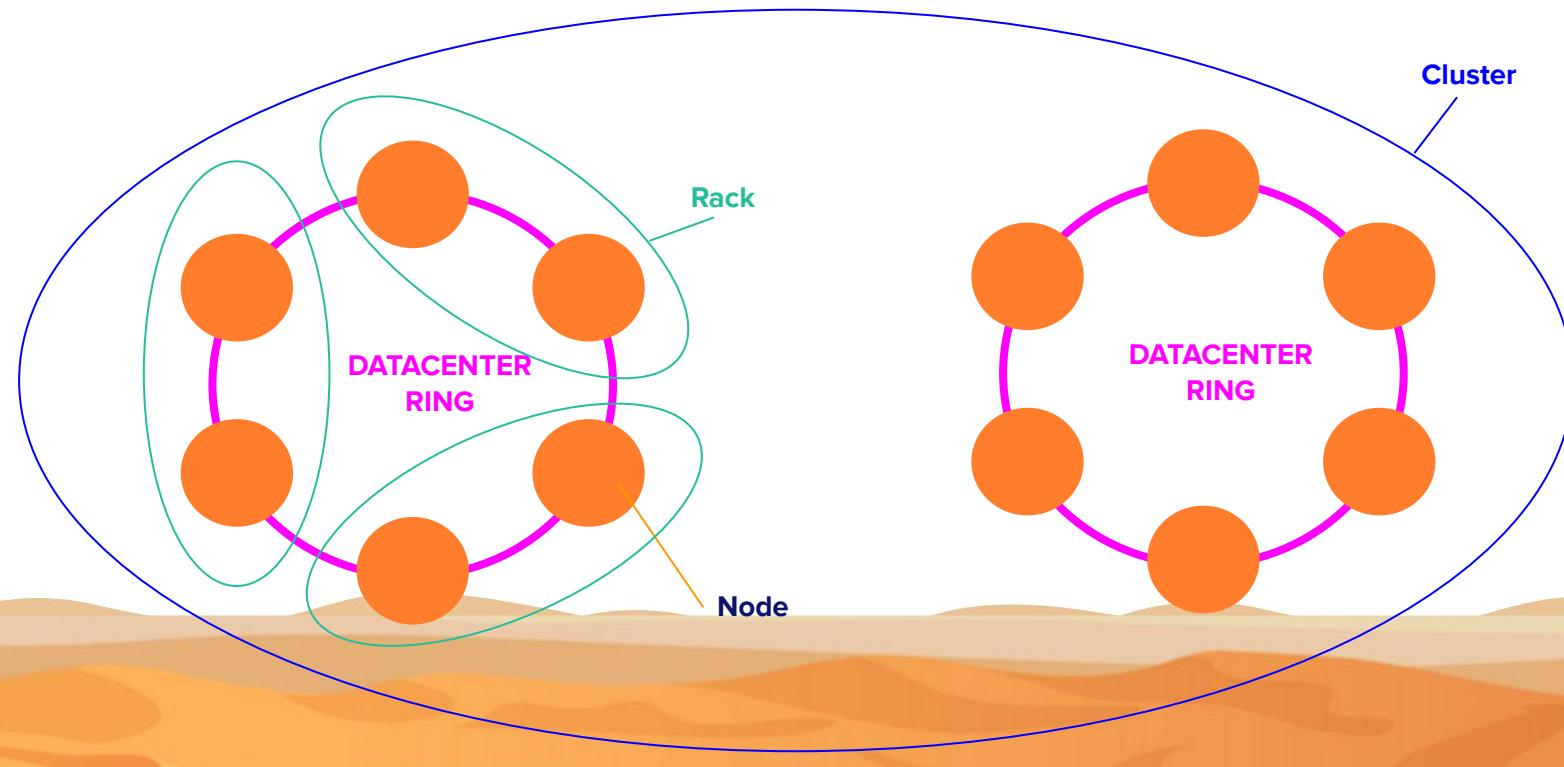
MATERIALS



bit.ly/CassandraWorkshopMaterials



Apache Cassandra™ Vocabulary

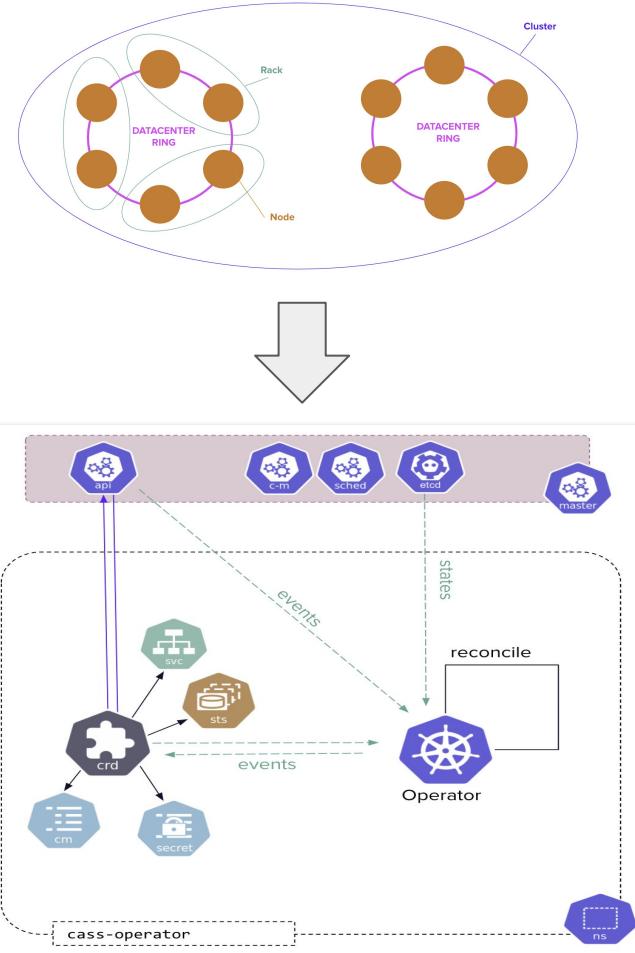


MATERIALS



bit.ly/CassandraWorkshopMaterials

YOUR MISSION SHOULD YOU CHOOSE TO ACCEPT IT



MATERIALS



bit.ly/CassandraWorkshopMaterials

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)

Exercise

Cass Operator

<https://github.com/DataStax-Academy/kubernetes-workshop-online/blob/master/1-cassandra/README.MD>

2. Create a single node cluster

Apply this file via `kubectl` and watch the list of pods as the operator deploys them. Completing a deployment may take several minutes per node.

2a. Create the cluster

```
kubectl -n cass-operator apply -f ./1-cassandra/12-cassandra-cluster-1nodes.yaml
```

2b. Watch progression

```
watch kubectl -n cass-operator get pod
```

Expected output

NAME	READY	STATUS	RESTARTS	AGE
cass-operator-657cb5c695-q9psl	1/1	Running	0	5m22s
cluster1-dc1-default-sts-0	1/2	Running	0	50s

2c. Execute the command to describe the datacenter

```
kubectl -n cass-operator describe cassdc dc1
```

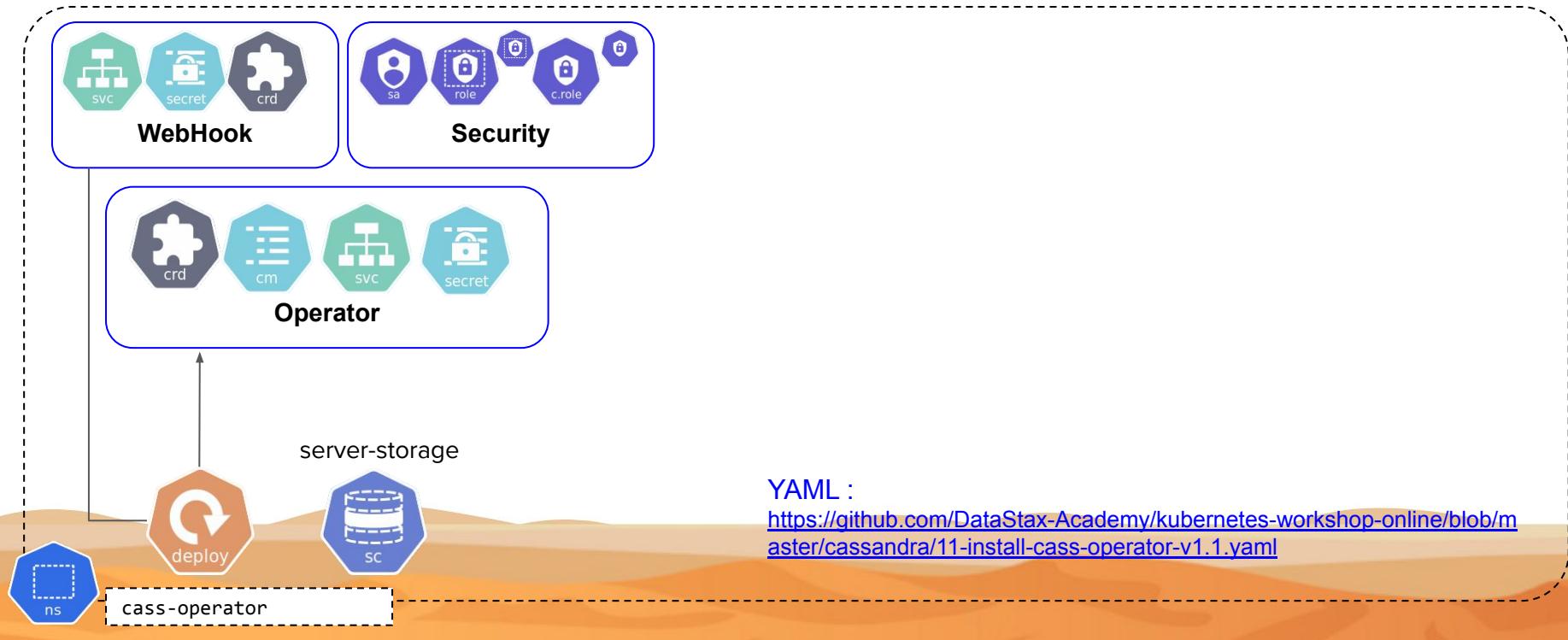
Expected output

MATERIALS



bit.ly/CassandraWorkshopMaterials

Installing the Cass Operator Manifest



MATERIALS



bit.ly/CassandraWorkshopMaterials

CRD CassandraDataCenter

```
apiVersion: cassandra.datastax.com/v1beta1
kind: CassandraDatacenter
metadata:
  name: dc1
spec:
  clusterName: cluster1
  serverType: cassandra
  serverVersion: "3.11.6"
  managementApiAuth:
    insecure: {}
  size: 1
  storageConfig:
    cassandraDataVolumeClaimSpec:
      storageClassName: server-storage
      accessModes:
        - ReadWriteOnce
      resources:
        requests:
          storage: 5Gi
  config:
    cassandra-yaml:
      authenticator: org.apache.cassandra.auth.PasswordAuthenticator
      authorizer: org.apache.cassandra.auth.CassandraAuthorizer
      role_manager: org.apache.cassandra.auth.CassandraRoleManager
  jvm-options:
    initial_heap_size: "800M"
    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
```

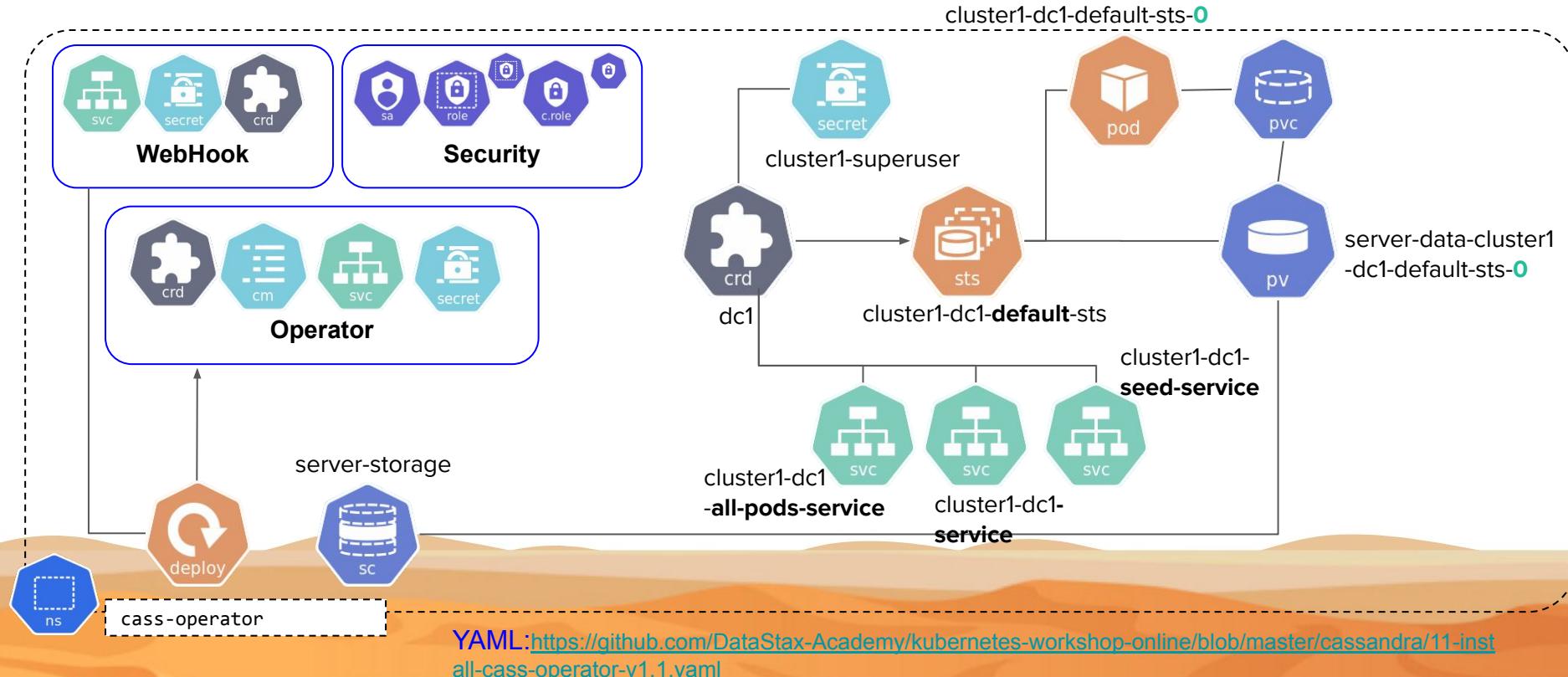
<https://github.com/DataStax-Academy/kubernetes-workshop-online/blob/master/cassandra/12-cassandra-cluster-1nodes.yaml>

MATERIALS



bit.ly/CassandraWorkshopMaterials

Creating DataCenter dc1



MATERIALS



bit.ly/CassandraWorkshopMaterials

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)

MATERIALS



bit.ly/CassandraWorkshopMaterials

Cassandra Management API SideCar

<https://petstore.swagger.io/>

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

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

The Management API is a sidecar service layer that attempts to build a well supported set of operational actions on Cassandra® nodes that can be administered centrally. It currently works with official Apache Cassandra® 3.11.x and 4.0 via a drop in java agent.

- Lifecycle Management
 - Start Node
 - Stop Node
- Configuration Management (alpha)
 - Change YAML
 - Change jvm-opts
- Health Checks
 - Kubernetes liveness/readiness checks
 - Consistency level checks
- Per node actions
 - All nodetool commands

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

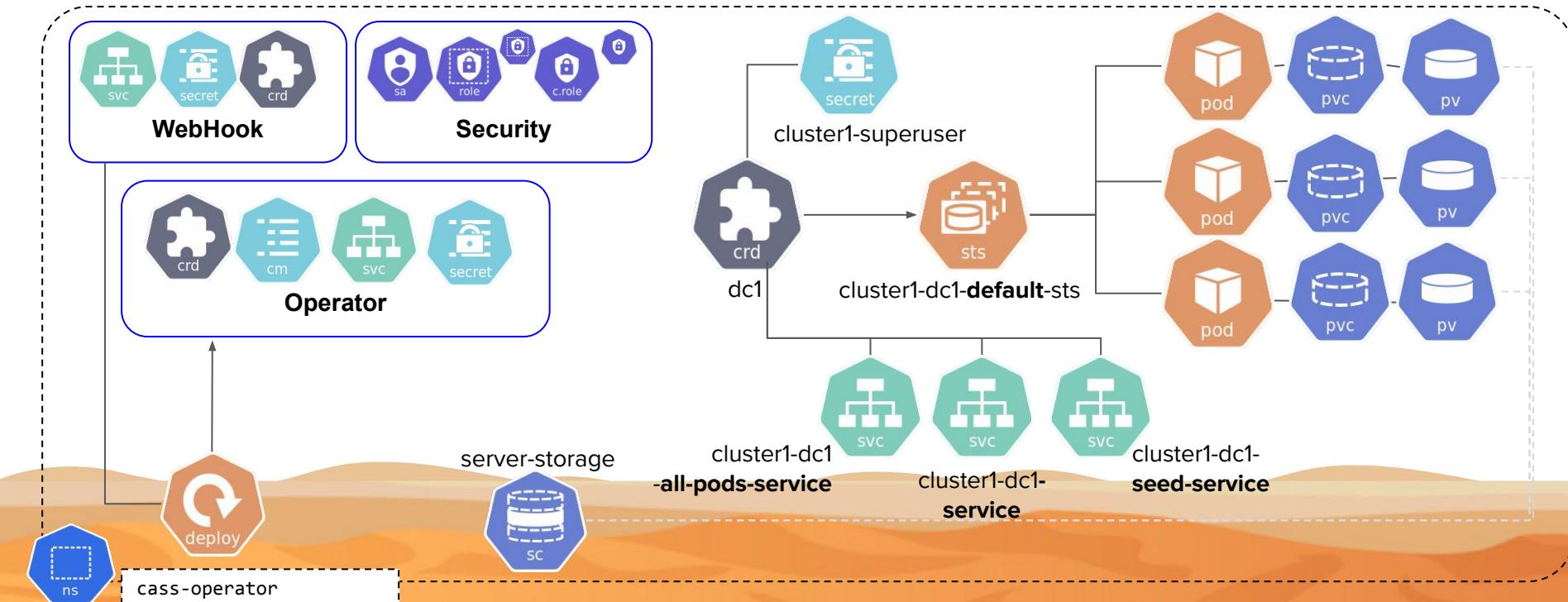
Method	Path	Description
POST	/api/v0/ops/auth/role	Creates a new user role
GET	/api/v0/probes/liveness	Indicates whether this service is running
GET	/api/v0/probes/readiness	Indicates whether the Cassandra service is ready to service requests
GET	/api/v0/probes/cluster	Indicated whether the Cassandra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

MATERIALS



bit.ly/CassandraWorkshopMaterials

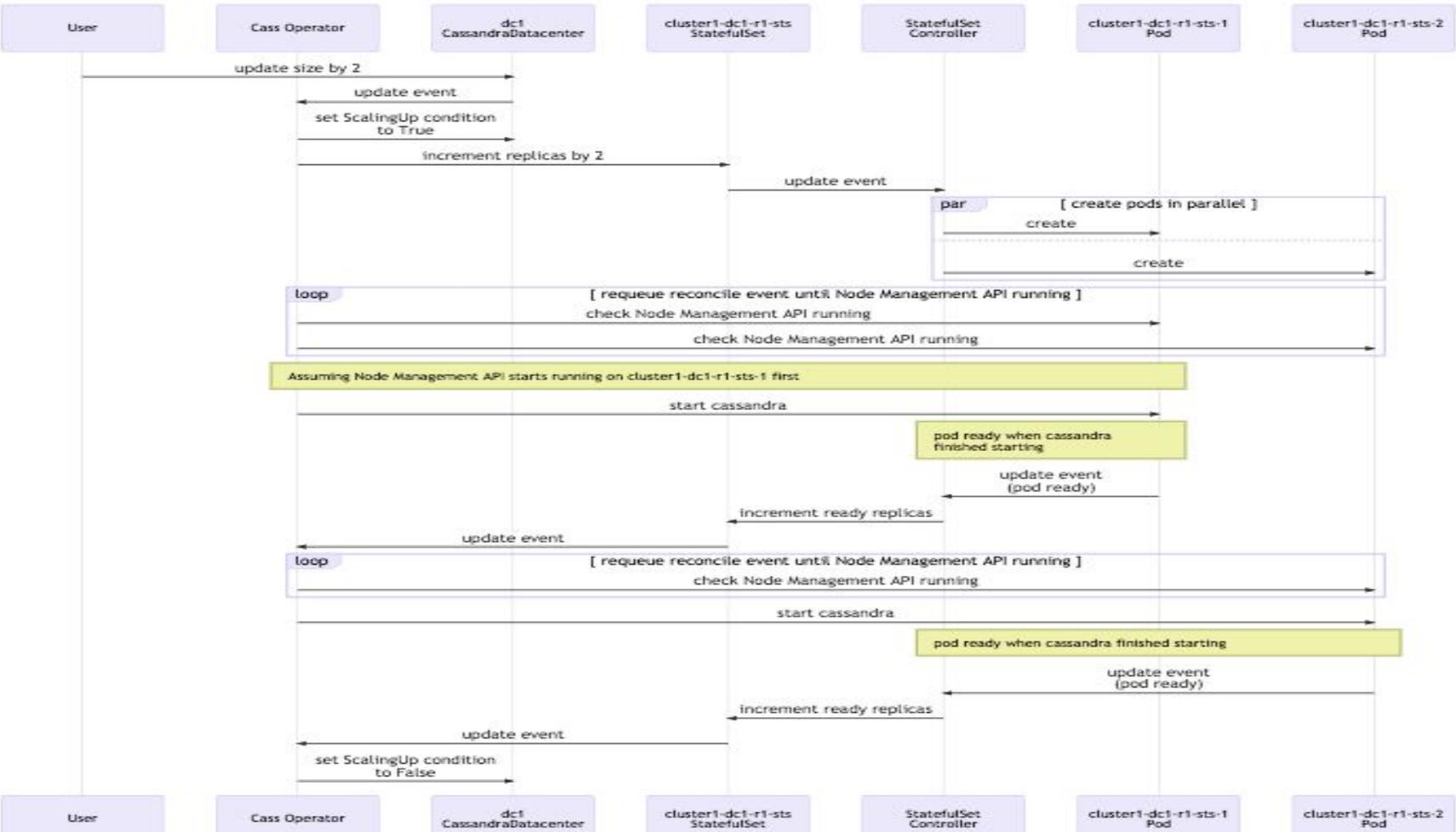
Scale up DataCenter dc1



MATERIALS



bit.ly/CassandraWorkshopMaterials



User

Cass Operator

dc1 CassandraDatacenter

cluster1-dc1-r1-sts StatefulSet

StatefulSet Controller

cluster1-dc1-r1-sts-1 Pod

cluster1-dc1-r1-sts-2 Pod

MATERIALS

<https://bit.ly/CassandraWORKSHOPMaterials>

DATASTAX® Apache Cassandra®



Developer Workshop Series **Week5**

What we will cover:

- Housekeeping
- Kubernetes Basics
- Kubernetes Operators
- Cass Operator in Deep
- Dashboarding + Grafana | Prometheus
- Resources



MATERIALS



bit.ly/CassandraWorkshopMaterials

Exercise Dashboard

[https://github.com/DataStax-Academy/
kubernetes-workshop-online/blob/master/dashboard/README.MD](https://github.com/DataStax-Academy/kubernetes-workshop-online/blob/master/dashboard/README.MD)

Name	Labels	Pods	Age	Images
cassandra-operator	-	1 / 1	an hour	datastax/cassandra-operator:1.0

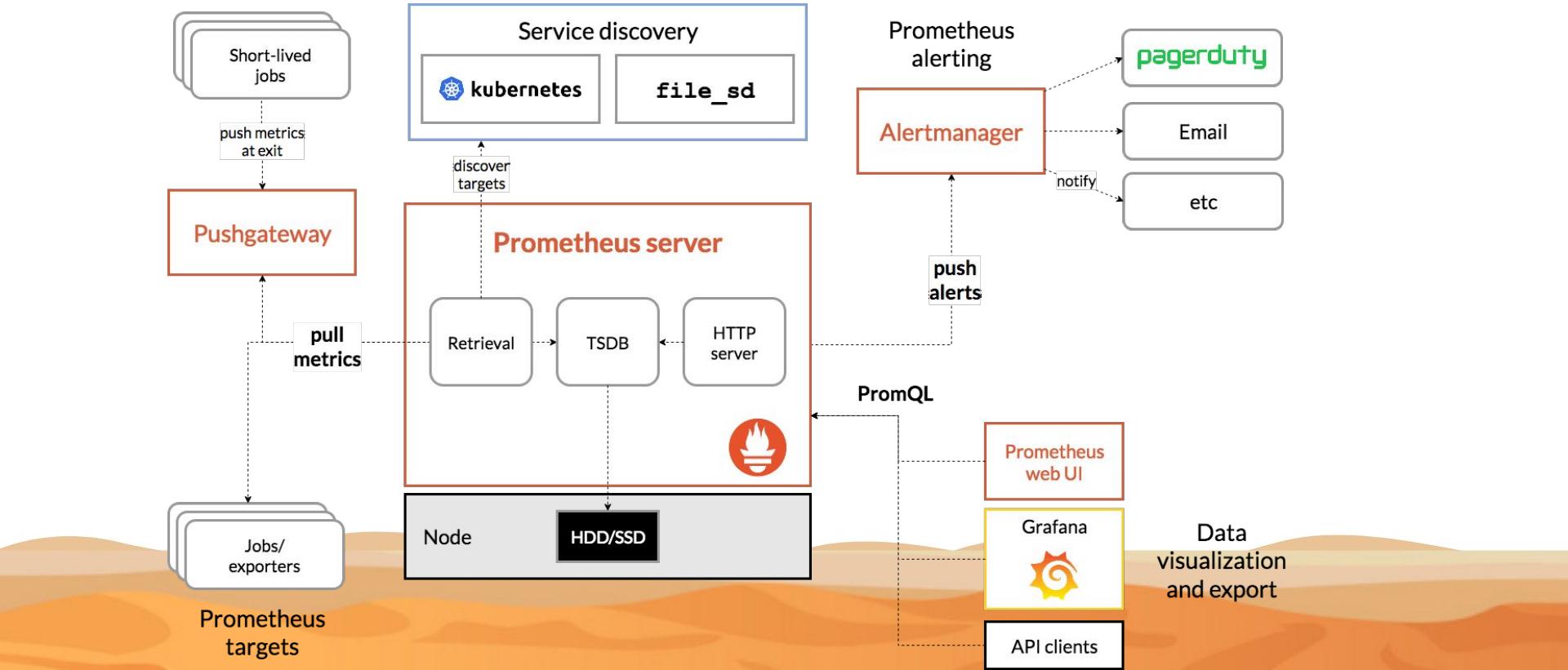
Name	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Age
cluster1-dc1-default-sts-0	app.kubernetes.io/managed-by: cassandra-operator	cassandra.datastax.com/cluster: cluster1	Running	0	-	-	32 minutes
cluster1-dc1-default-sts-1	app.kubernetes.io/managed-by: cassandra-operator	cassandra.datastax.com/cluster: cluster1	Running	0	-	-	33 minutes
cluster1-dc1-default-sts-2	app.kubernetes.io/managed-by: cassandra-operator	cassandra.datastax.com/cluster: cluster1	Running	0	-	-	34 minutes

MATERIALS



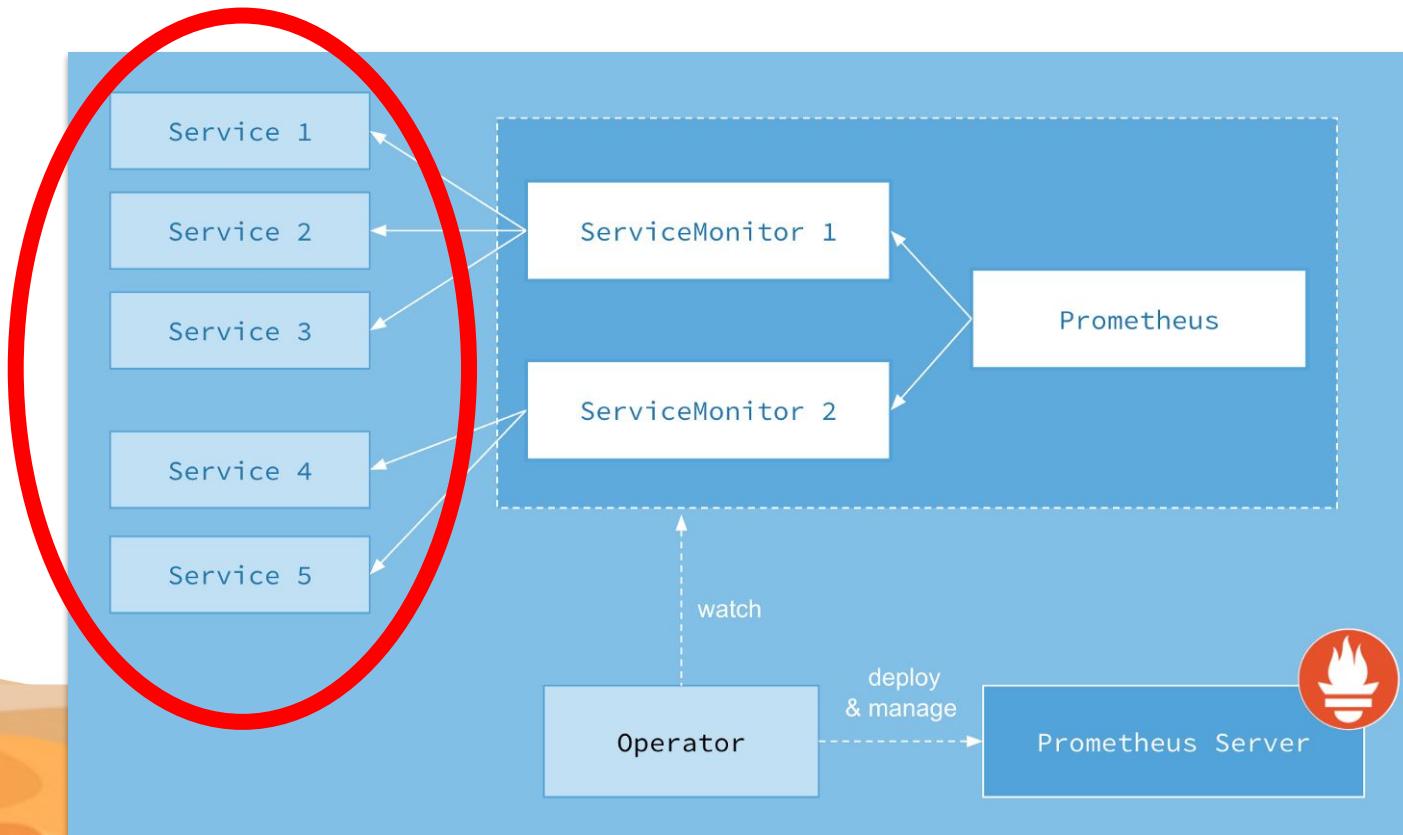
bit.ly/CassandraWorkshopMaterials

Prometheus + Grafana



MATERIALS → bit.ly/CassandraWorkshopMaterials

Prometheus metrics collection



MATERIALS



bit.ly/CassandraWorkshopMaterials

Dse Metrics Exporter

Reference Documentation

https://docs.datastax.com/en/dse/6.7/dse-dev/datastax_enterprise/tools/metricsCollector/mcIntroduction.html

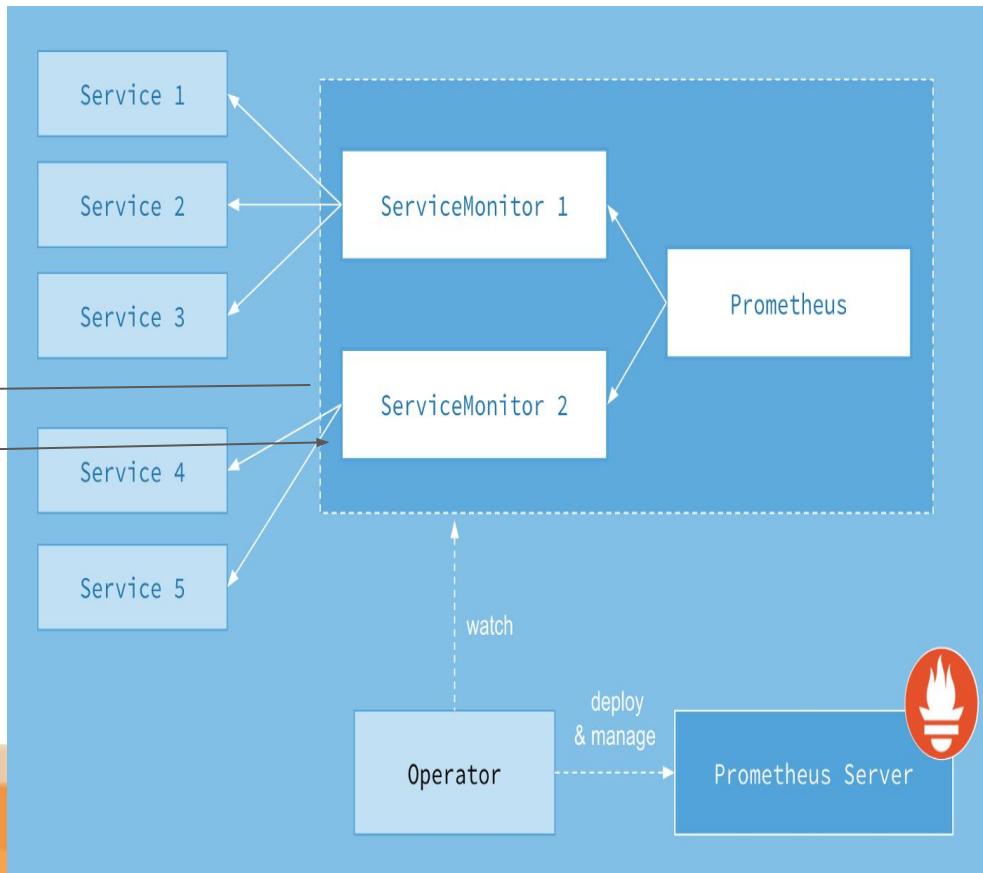
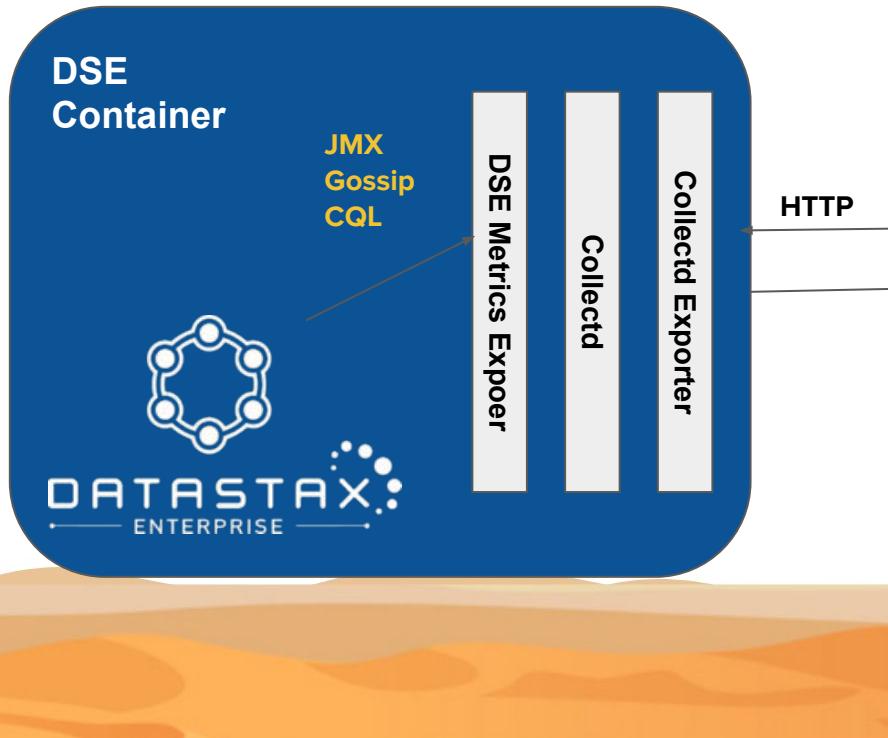
- DSE Metrics Collector aggregates DataStax Enterprise (DSE) metrics and integrates with existing monitoring solutions to facilitate problem resolution and remediation.
- **DSE Metrics Collector is built on collectd**, a popular, well-supported, open source metric collection agent. With [over 90 plugins](#), you can tailor the solution to collect metrics most important to your organization.
- When [DSE Metrics Collector](#) is enabled, DSE sends metrics and other structured events to DSE Metrics Collector.

`/etc/dse/collectd.conf tmpl`

```
LoadPlugin load
LoadPlugin memory
LoadPlugin swap
LoadPlugin uptime
LoadPlugin processes
LoadPlugin tcpconns
```



All Together



MATERIALS



bit.ly/CassandraWorkshopMaterials



Exercise

Grafana Prometheus



https://github.com/DataStax-Academy/kubernetes-workshop-online/blob/master/prometheus_grafana/README.MD

MATERIALS



bit.ly/CassandraWorkshopMaterials



Developer Workshop Series **Week5**



- Housekeeping
- Kubernetes Basics
- Kubernetes Operators
- Cass Operator in Deep
- Dashboarding + Grafana | Prometheus
- Resources

MATERIALS



bit.ly/CassandraWorkshopMaterials

Homework Week 5 (cf community)

1. Learn

- a. Watch talk introducing Kafka and Cassandra (29min)
- b. Short course Kafka and Cassandra on academy (2H)



2. Practice

- a. Finish workshop exercises, if needed following guidance on Github.
- b. (optional) Quarkus example
 - i. <https://github.com/datastax/cassandra-quarkus/tree/master/quickstart>

3. Validation form of the week



MATERIALS



bit.ly/CassandraWorkshopMaterials



Thank You





What we will
cover:

MATERIALS → bit.ly/CassandraWorkshopMaterials



MATERIALS → bit.ly/CassandraWorkshopMaterials



The Crew



MATERIALS



bit.ly/CassandraWorkshopMaterials



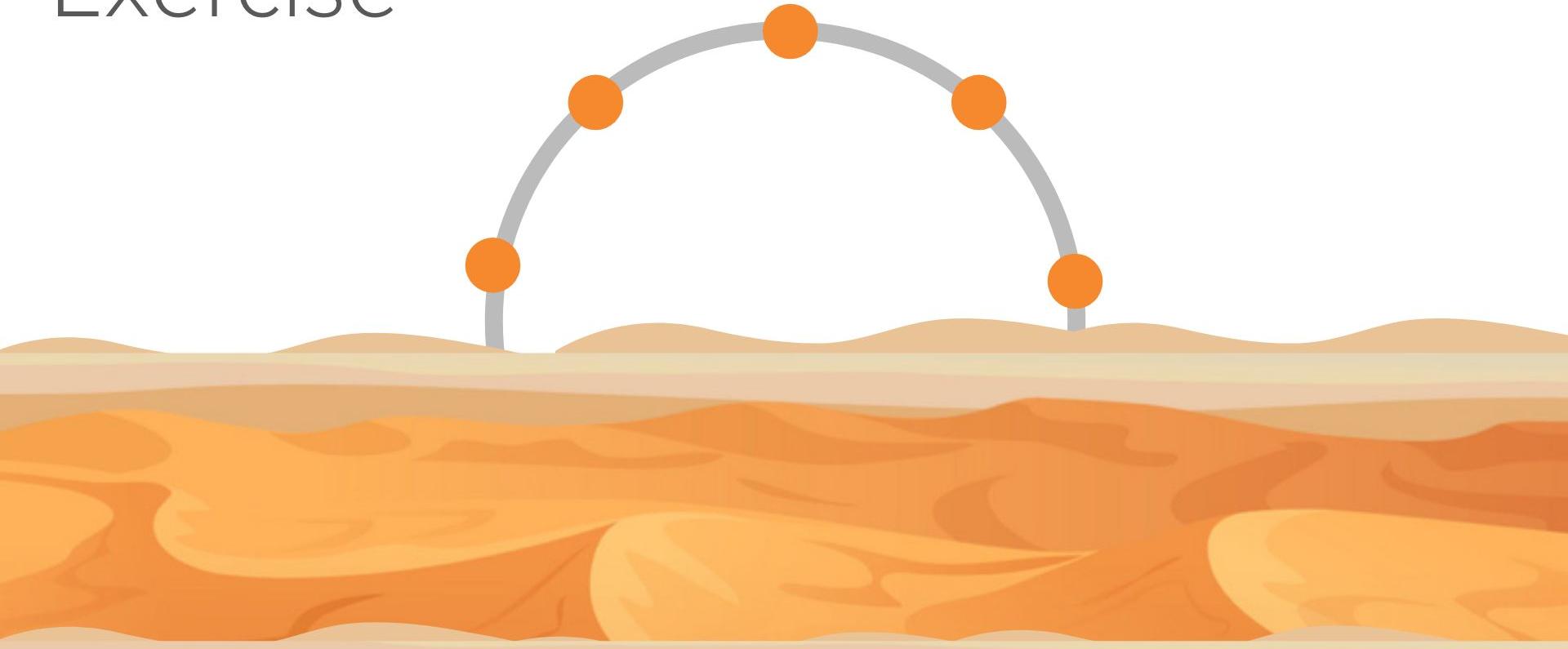
MATERIALS → bit.ly/CassandraWorkshopMaterials





MATERIALS → bit.ly/CassandraWorkshopMaterials

Exercise



MATERIALS → bit.ly/CassandraWorkshopMaterials