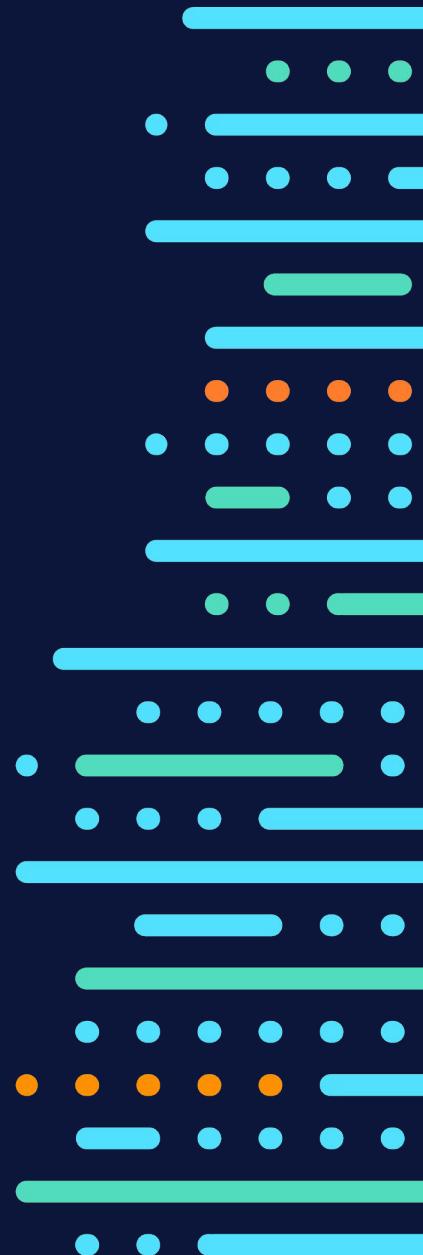




# Workshop online Series

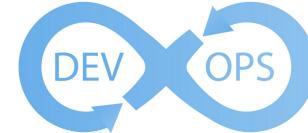
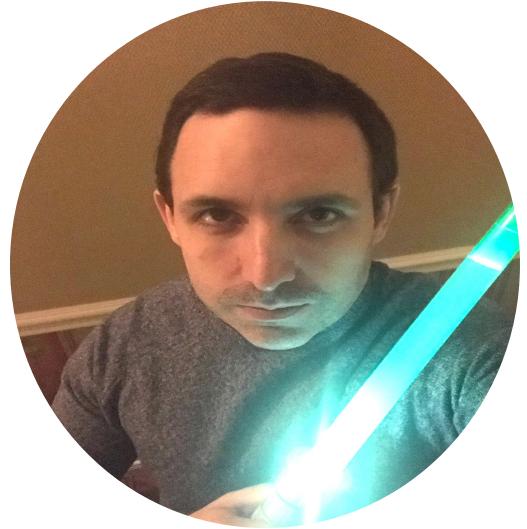
## Apache Cassandra™ with Kubernetes



# Your hosts

Cedrick Lunven

Developer Advocate



Aleksandr Volochnev

Developer Advocate

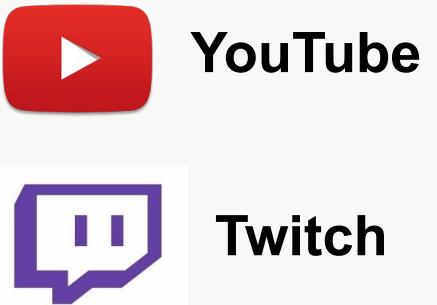
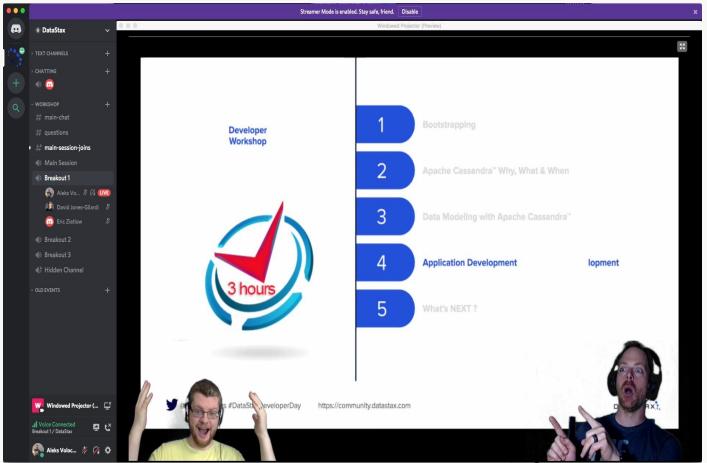


Jack Fryer

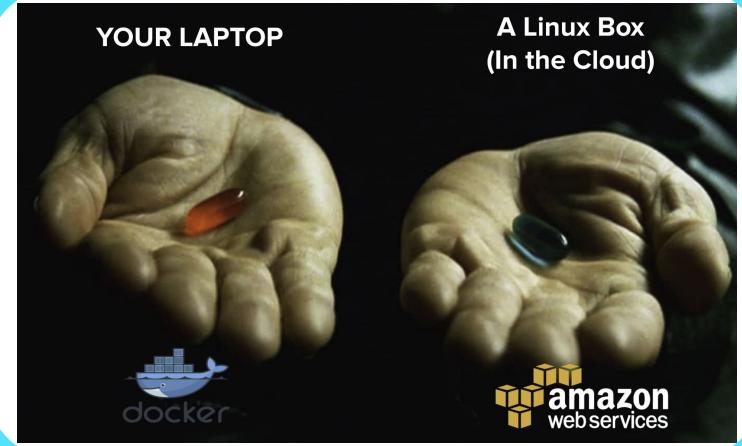
Community Manager



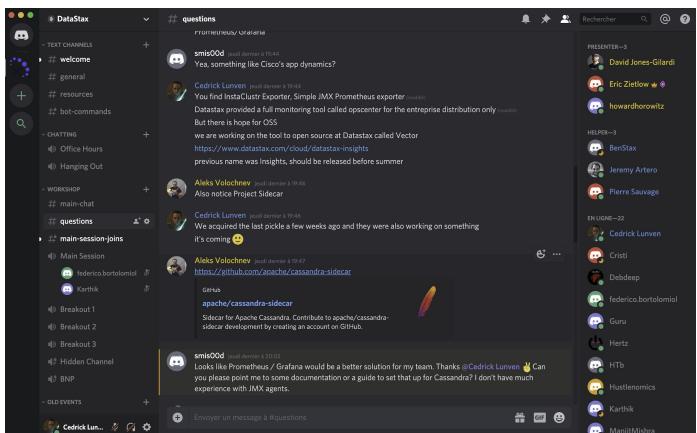
## STREAMS



## RUNTIME

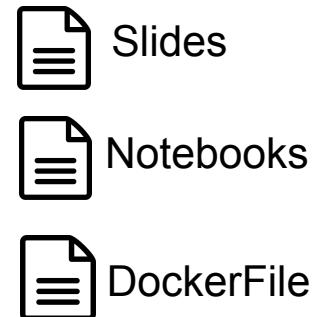


## QUESTIONS



Discord

## MATERIALS



# Apache Cassandra™ with Kubernetes

1

**Cassandra Why, What, When**

2

**Kubernetes Reminders + Setup**

3

**Cassandra Operator + *Hands-On***

4

**Dashboard UI + *Hands-On***

5

**Grafana Prometheus + *Hands-On***

6

**Resources**

# menti.com

# 36 63 61

## 6 questions



Available on the iPhone  
**App Store**

GET IT ON  
**Google play**

# Apache Cassandra™ with Kubernetes

1

**Cassandra Why, What, When**

2

**Kubernetes Reminders + Setup**

3

**Cassandra Operator + *Hands-On***

4

**Dashboard UI + *Hands-On***

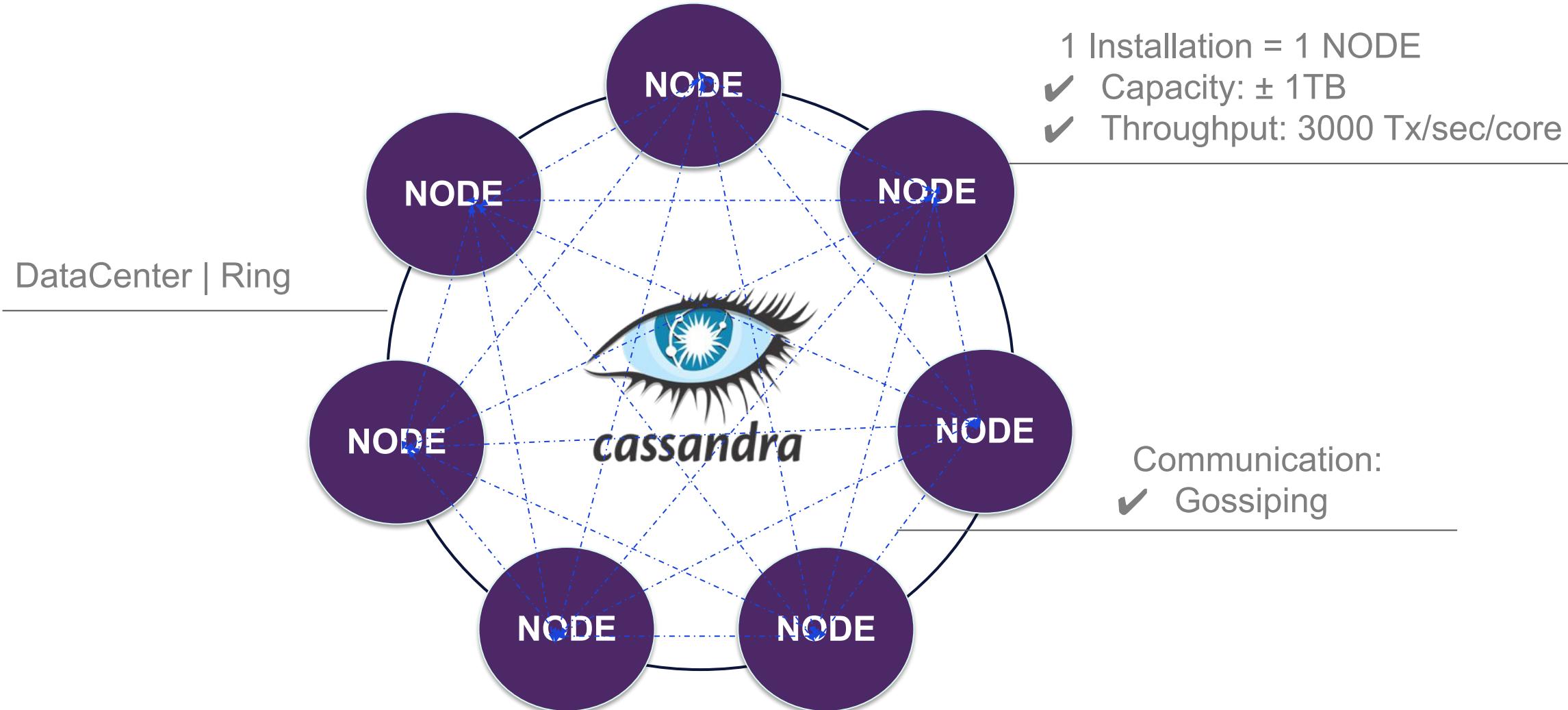
5

**Grafana Prometheus + *Hands-On***

6

**Resources**

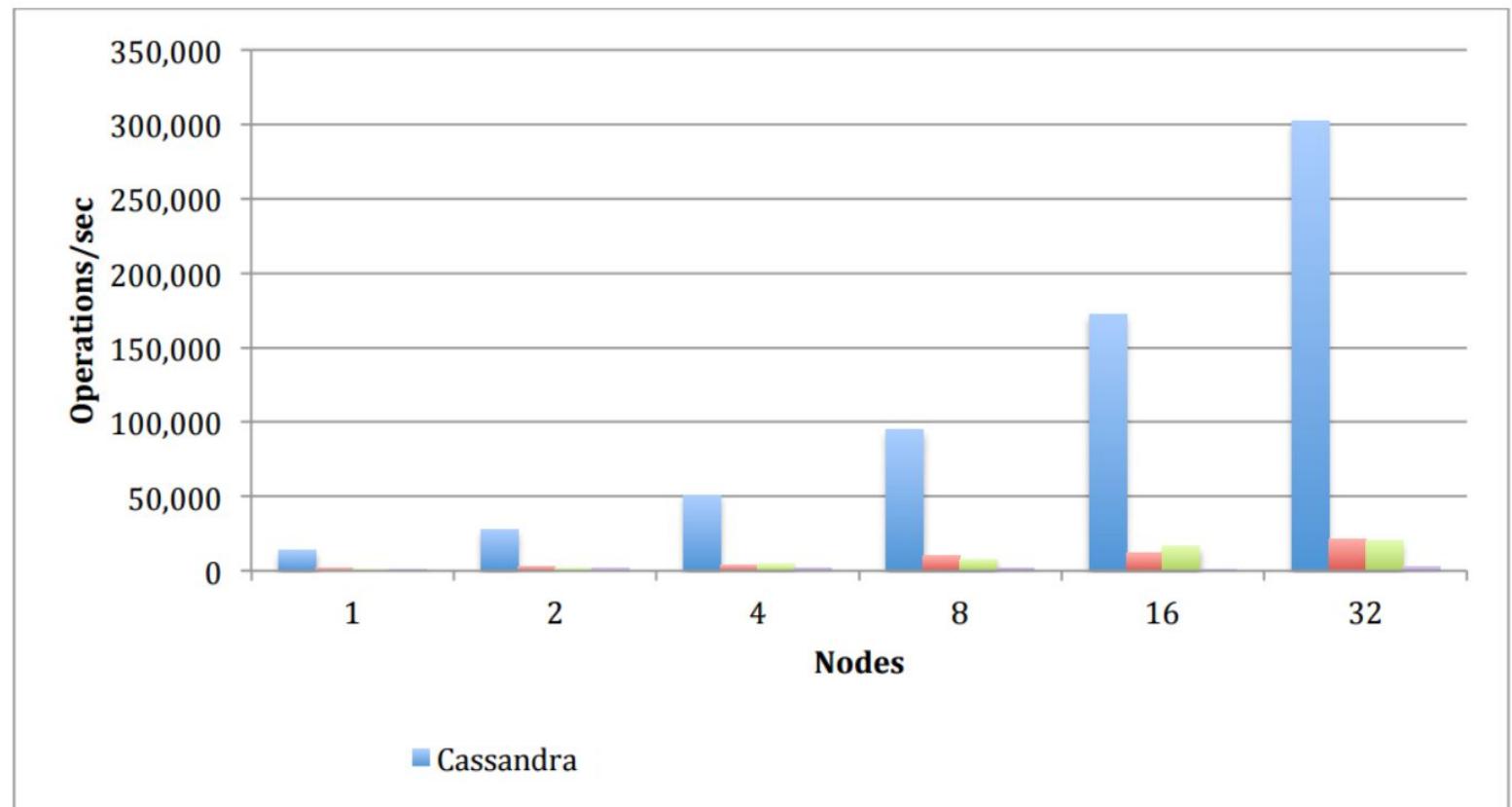
# Apache Cassandra™ = NoSQL Distributed Database



# Scales Linearly

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

Balanced Read/Write Mix



# 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

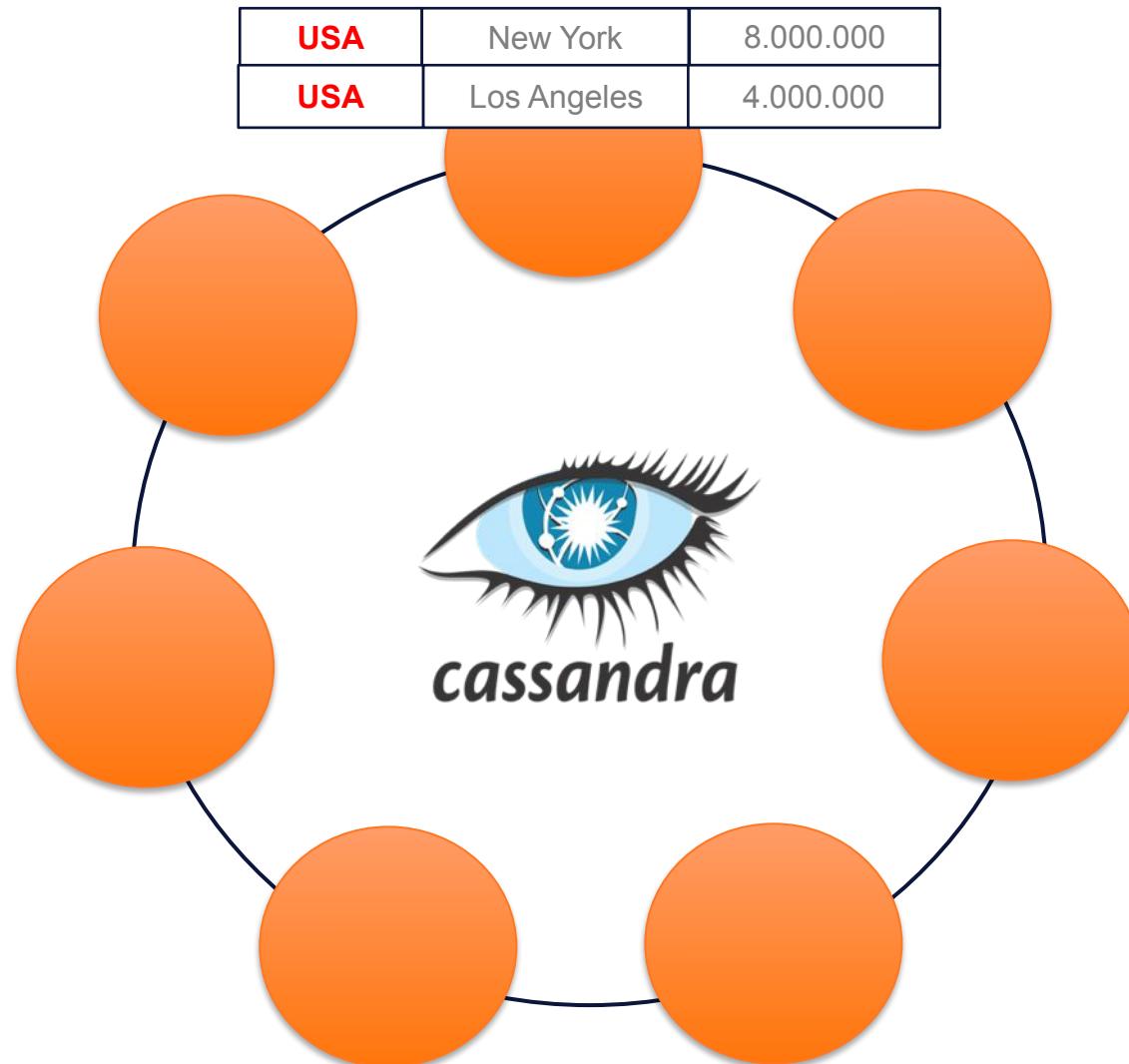


@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



# Data is Distributed



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

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

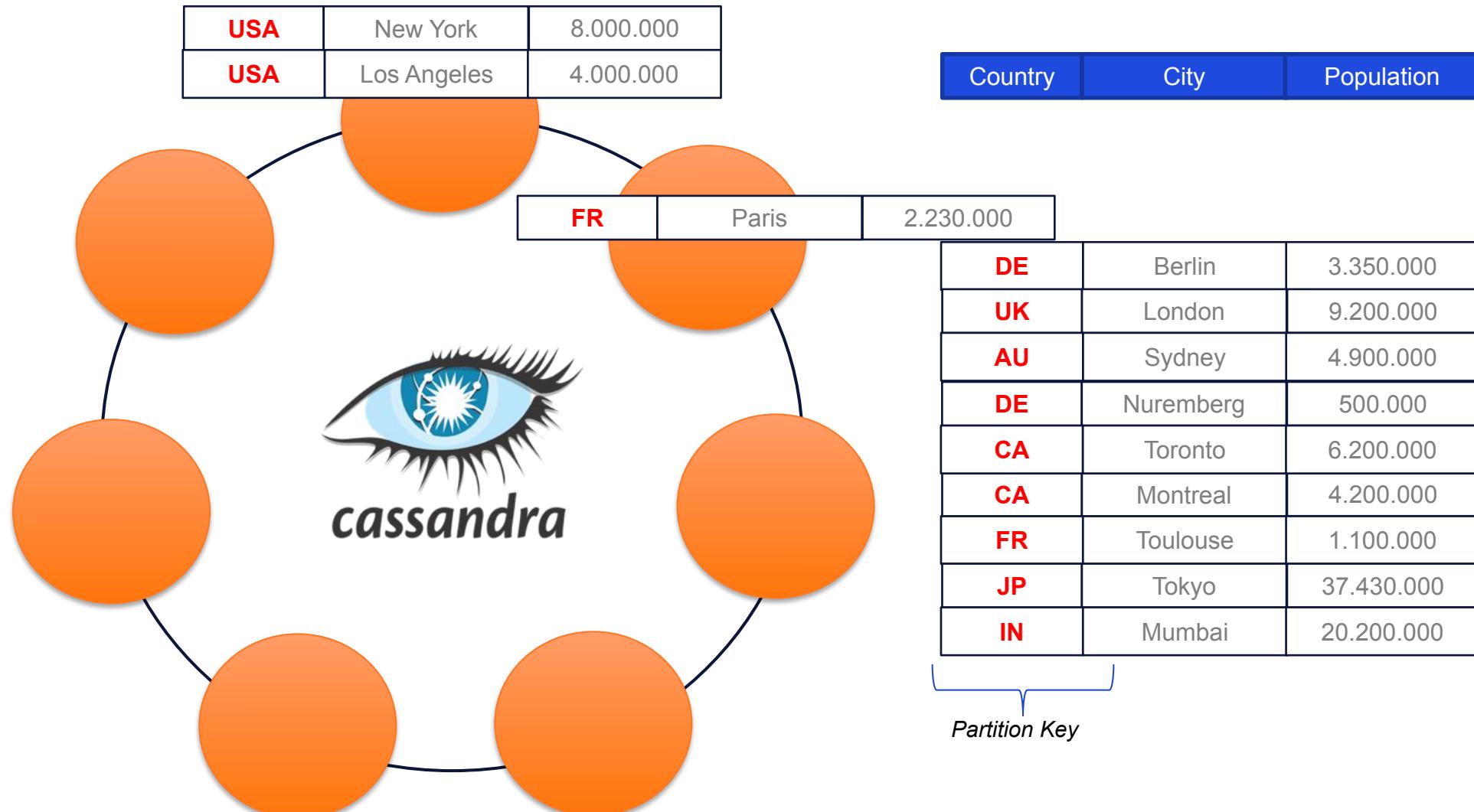


@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



# Data is Distributed

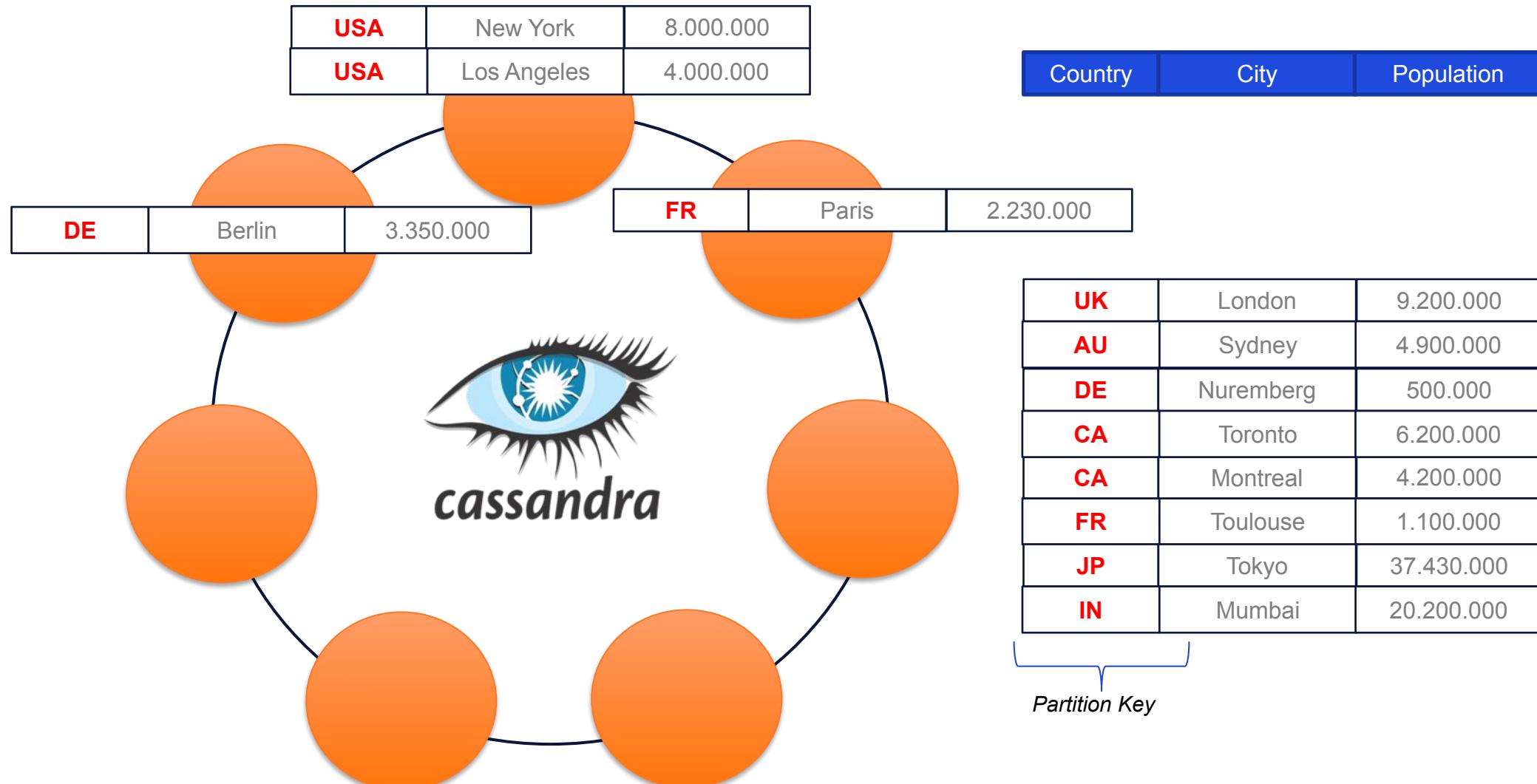


@DataStaxDevs #DataStaxDeveloperDay

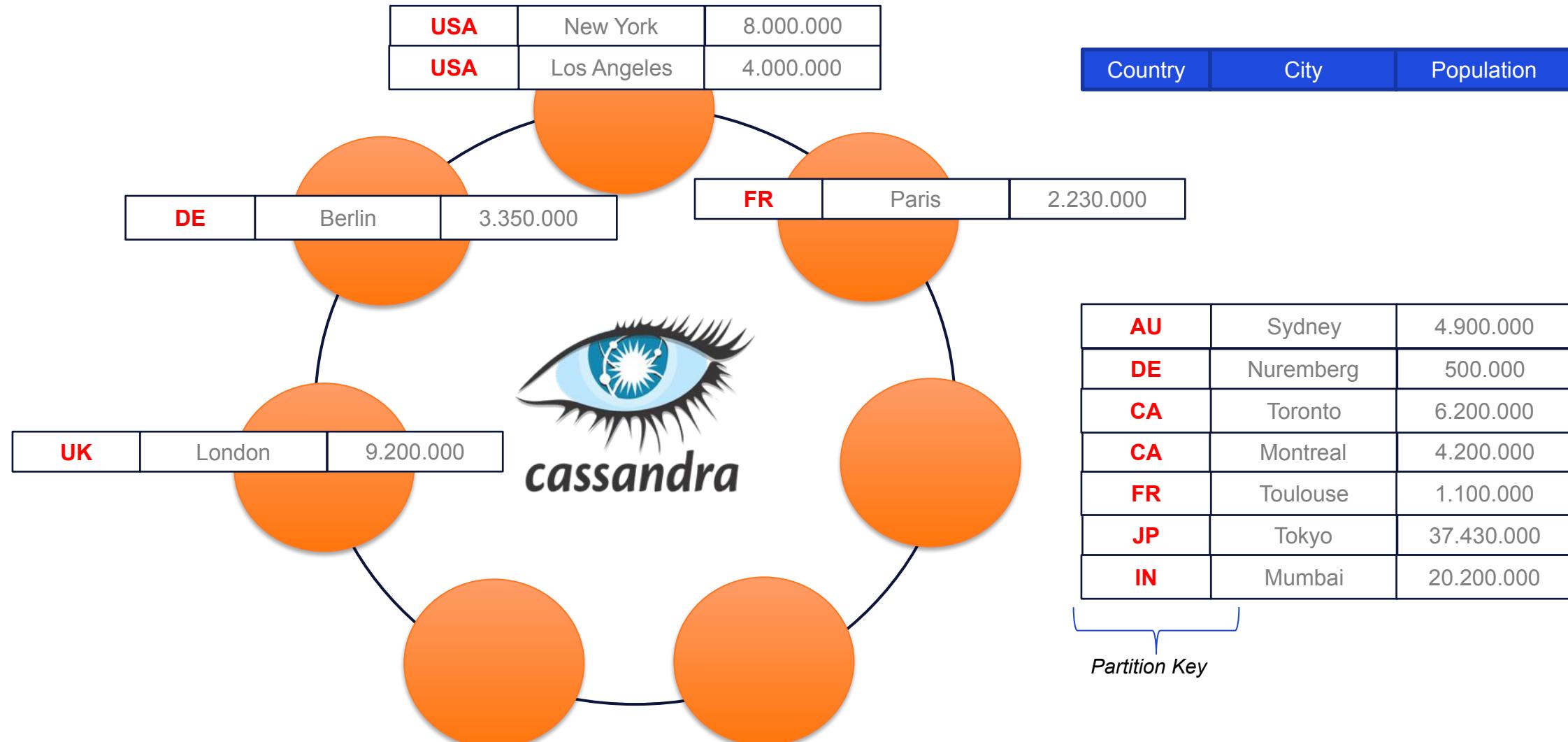
<https://community.datastax.com>



# Data is Distributed



# Data is Distributed

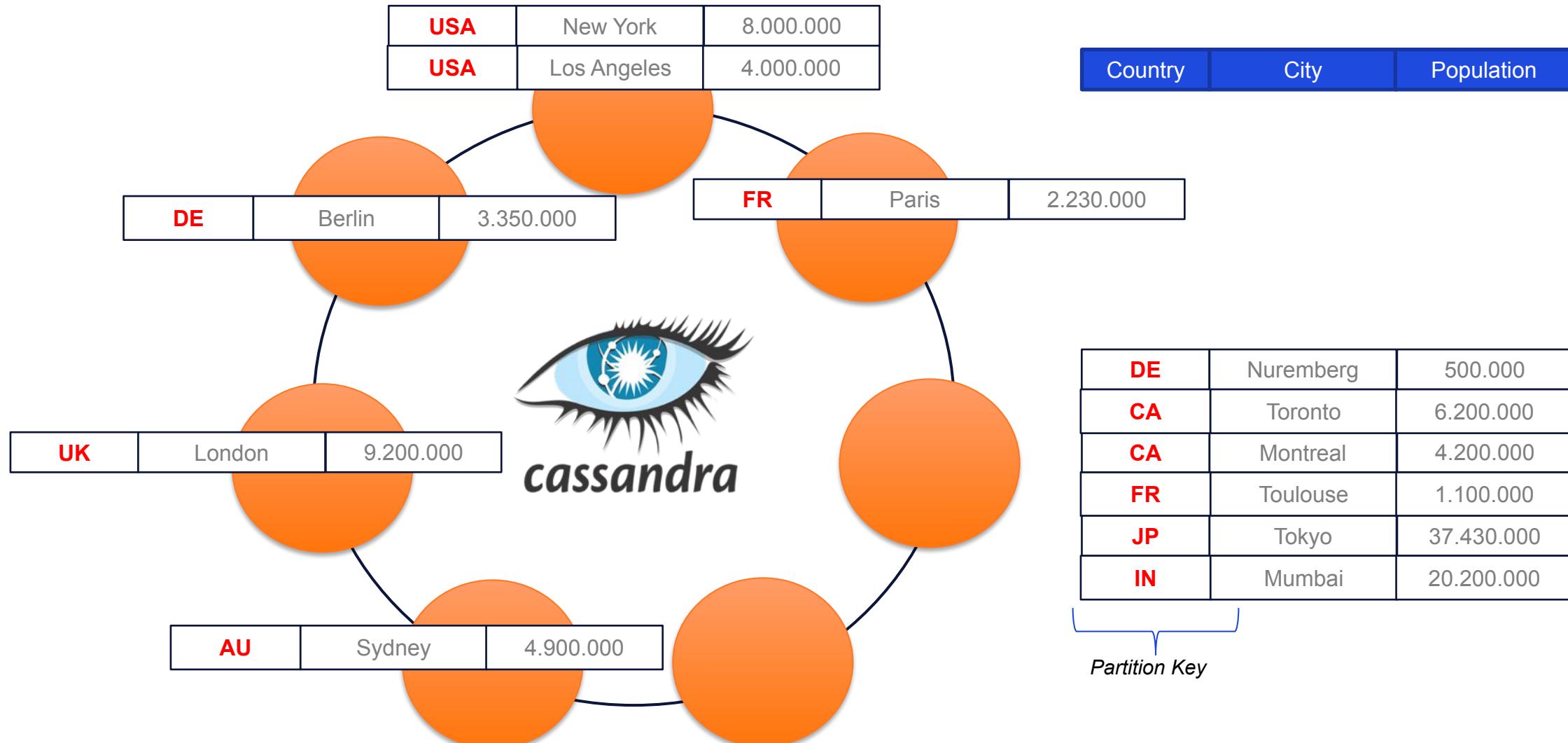


@DataStaxDevs #DataStaxDeveloperDay

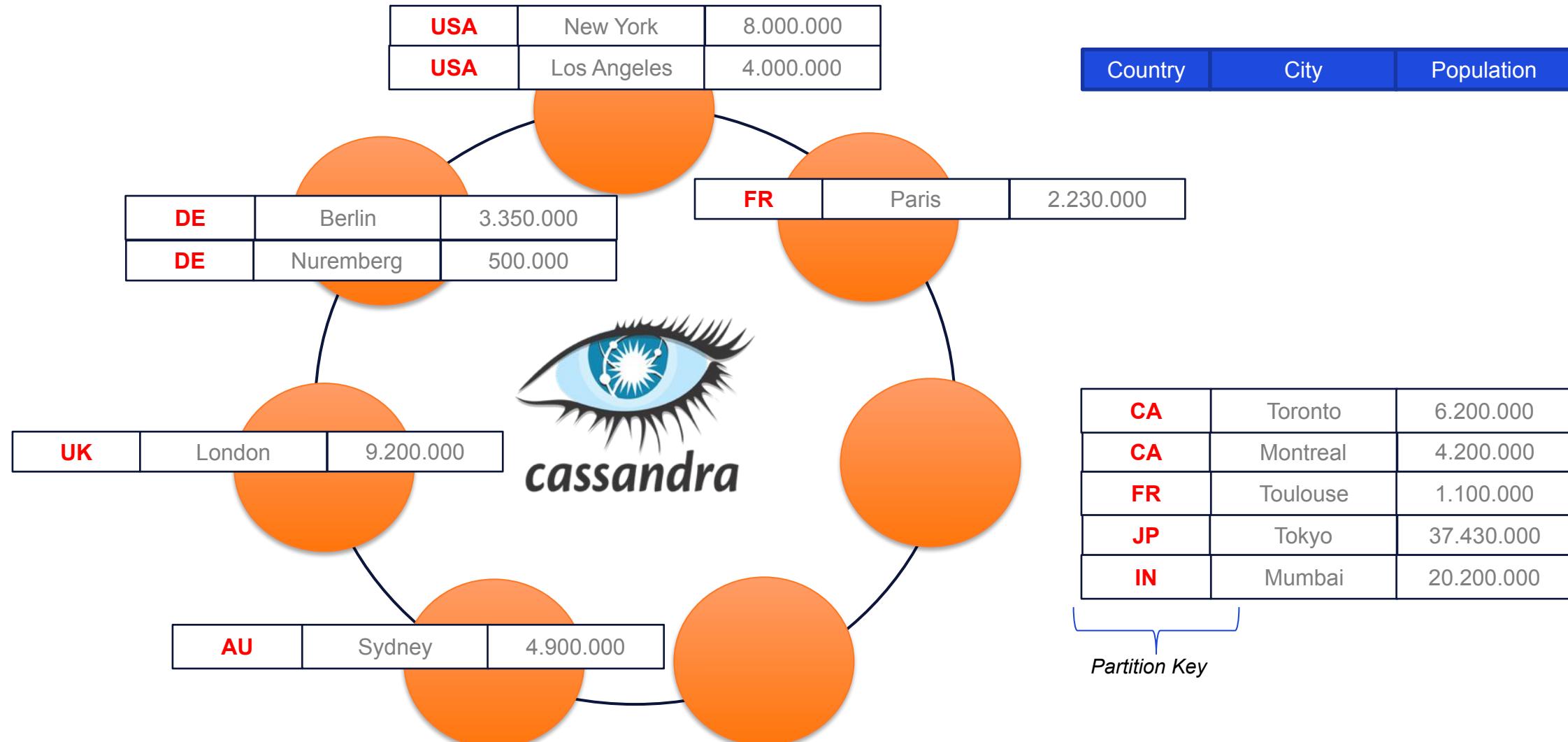
<https://community.datastax.com>



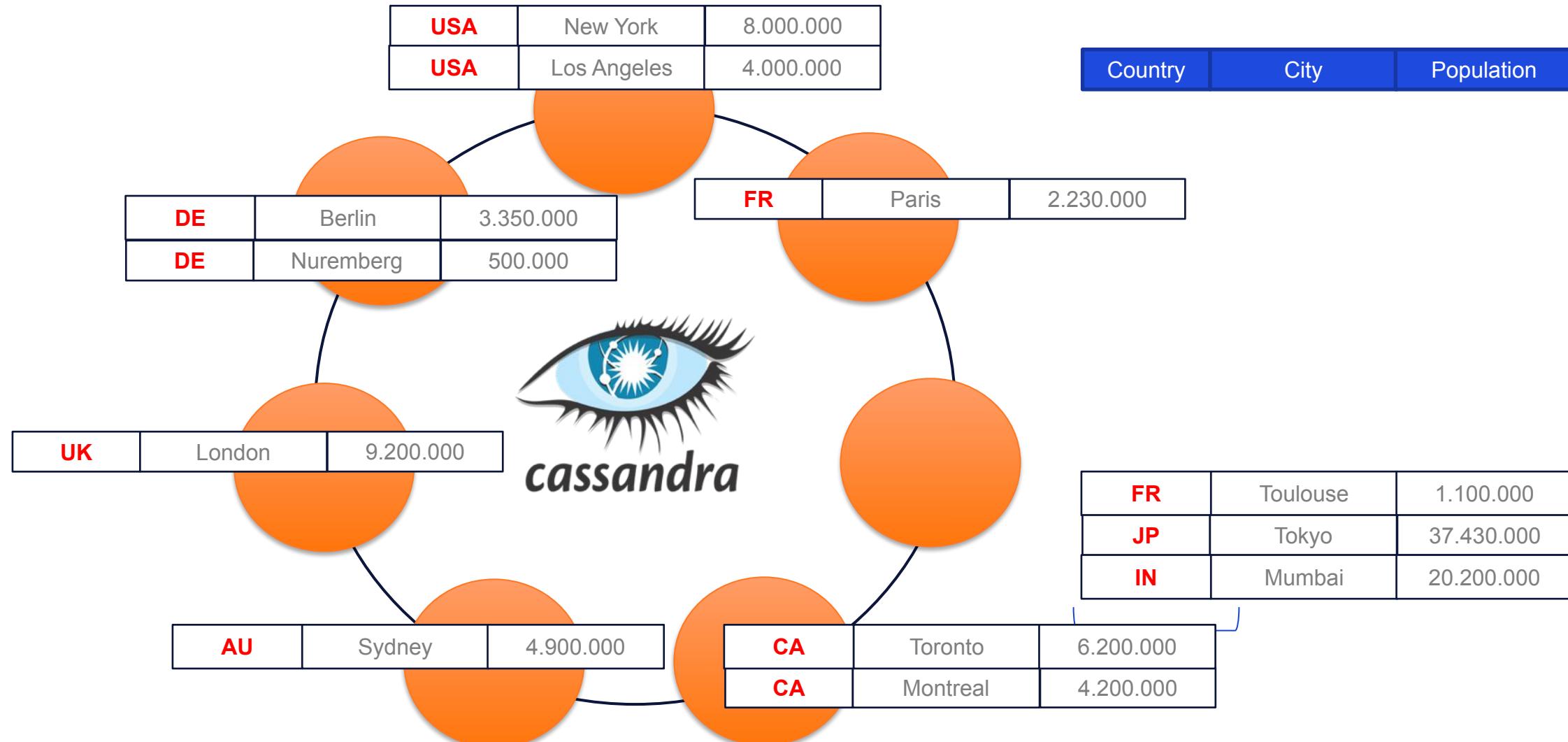
# Data is Distributed



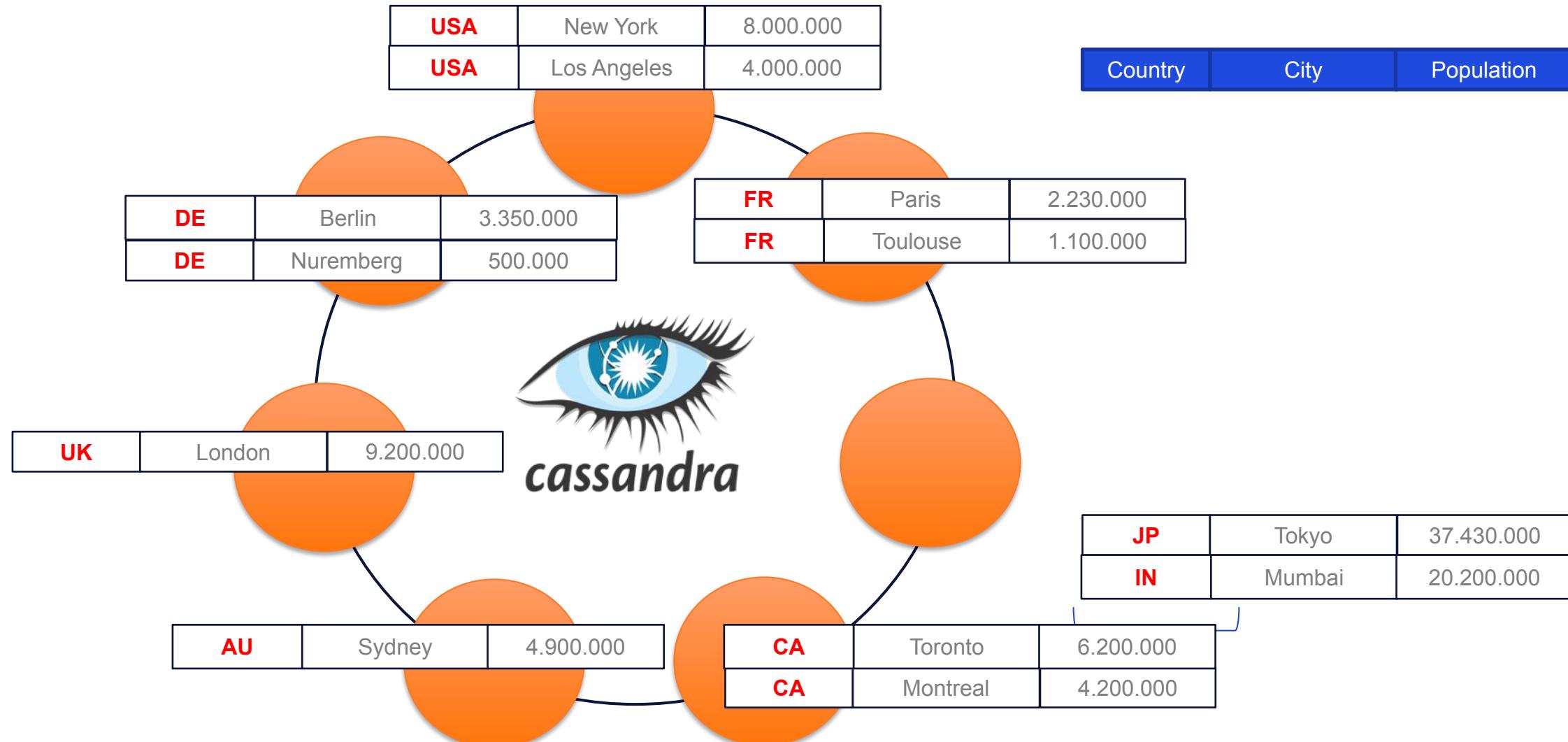
# Data is Distributed



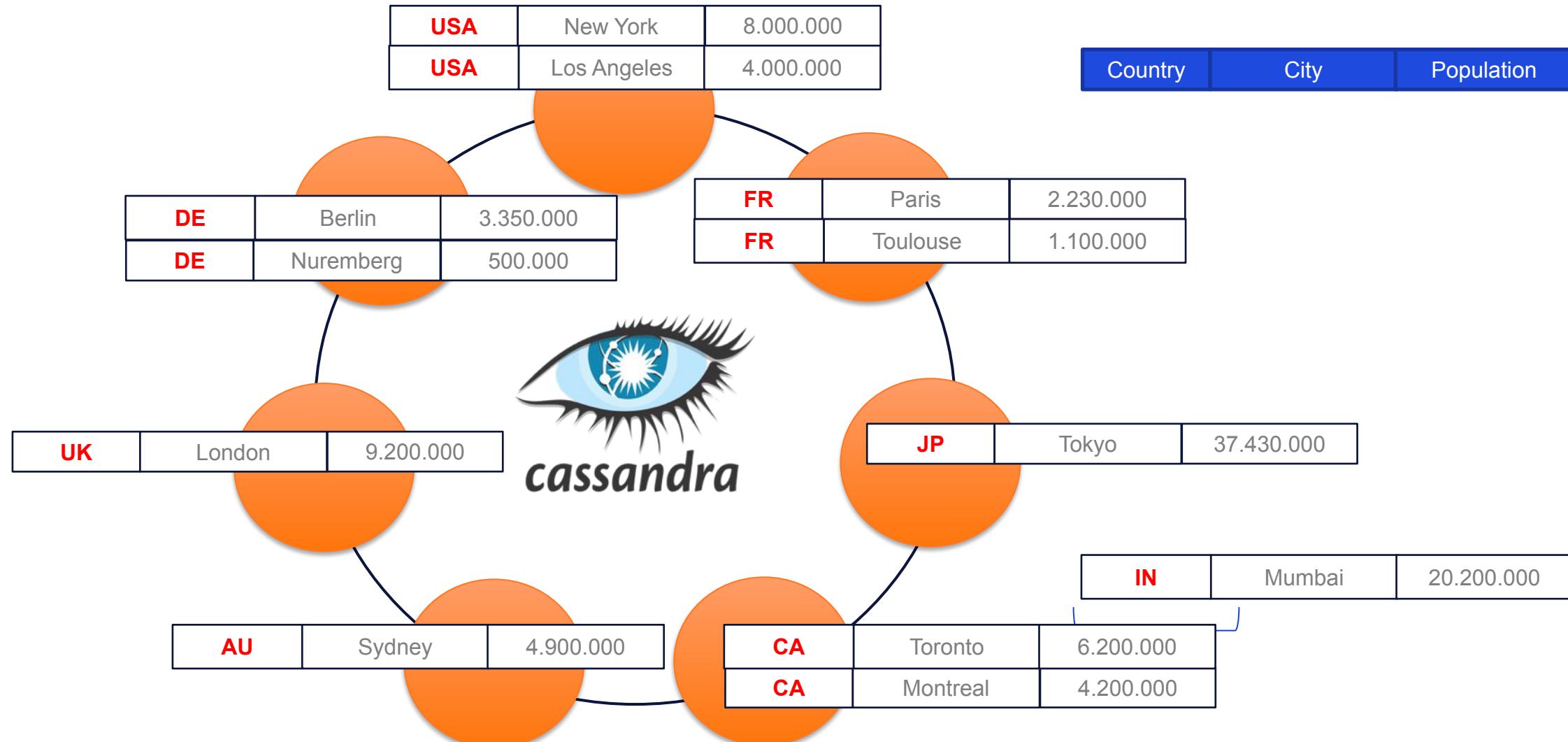
# Data is Distributed



# Data is Distributed



# Data is Distributed

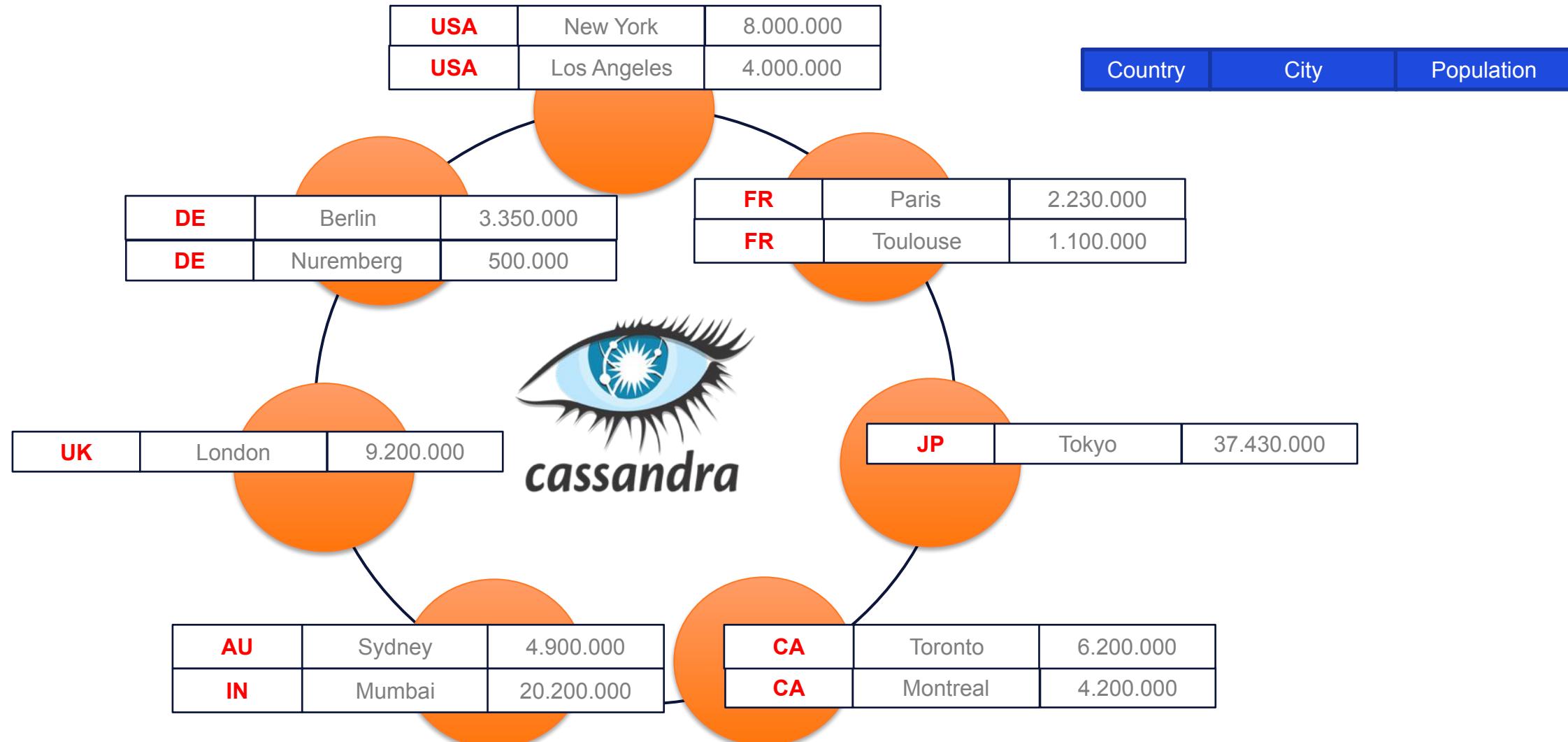


@DataStaxDevs #DataStaxDeveloperDay

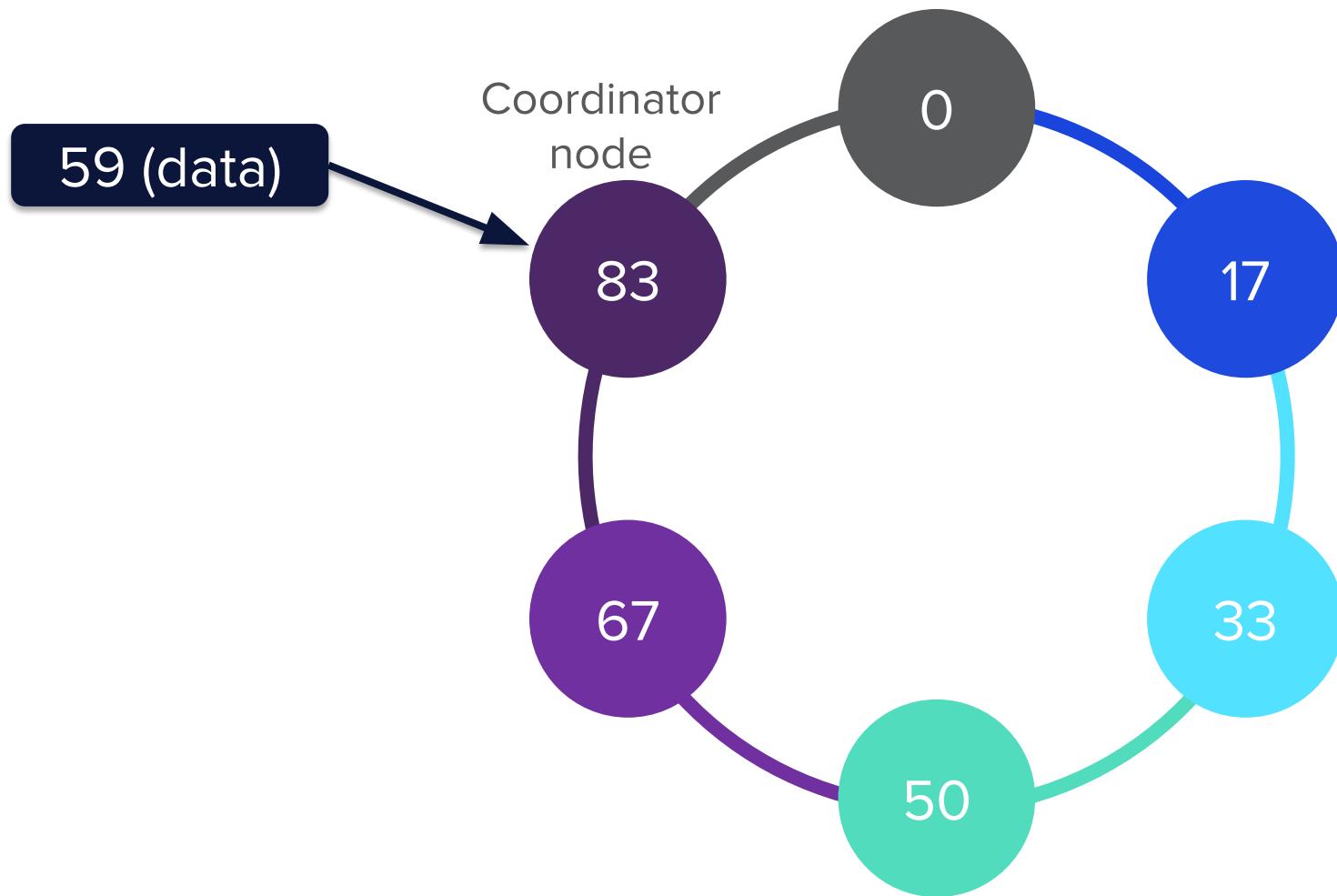
<https://community.datastax.com>



# Data is Distributed



# How the Ring Works

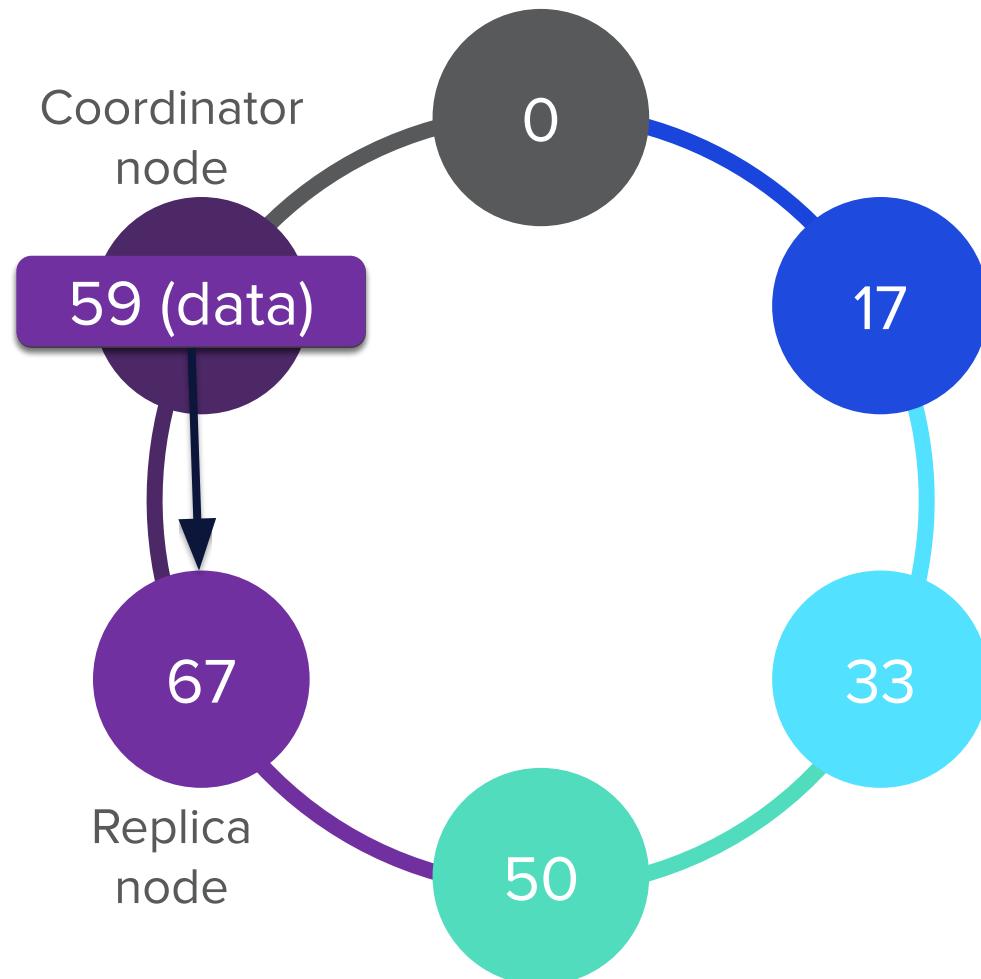


@DataStaxDevs #DataStaxDeveloperDay

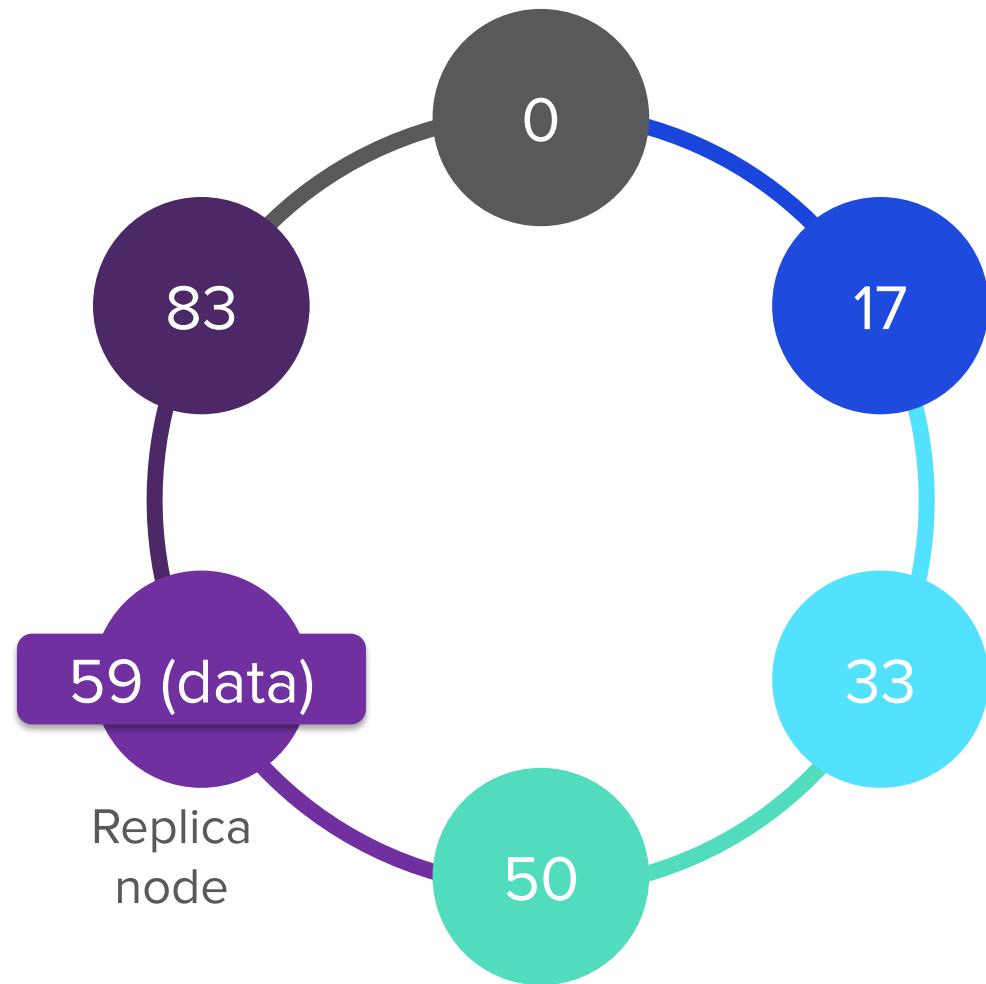
<https://community.datastax.com>



# How the Ring Works



# How the Ring Works



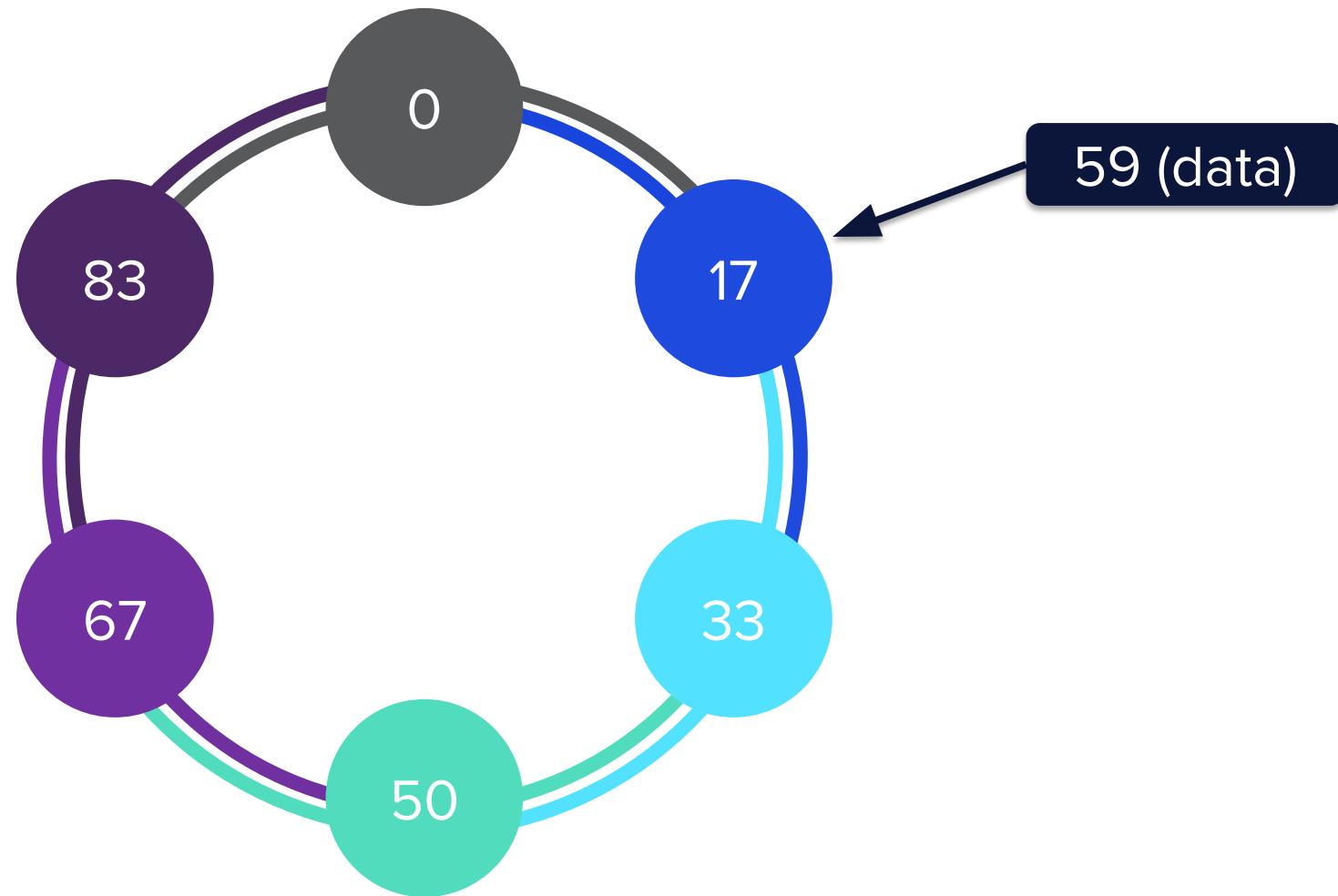
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



# Replication within the Ring

RF = 2



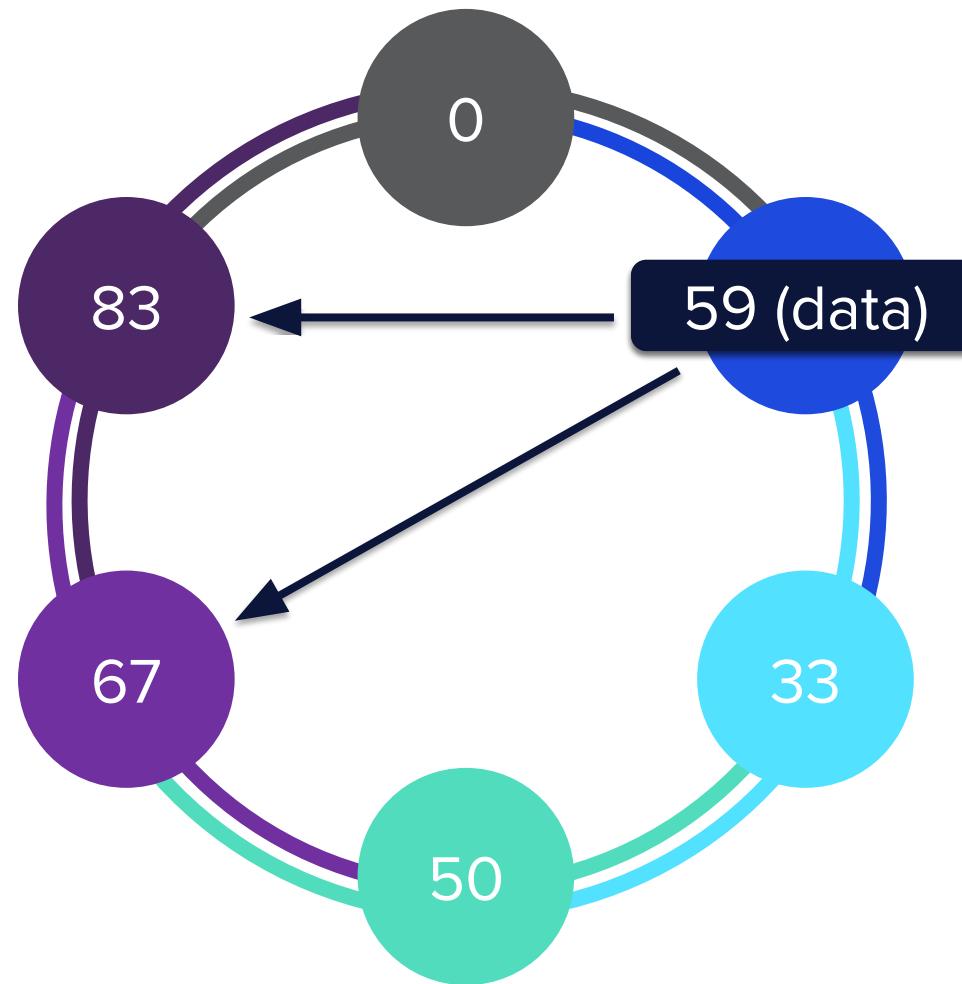
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



# Replication within the Ring

RF = 2



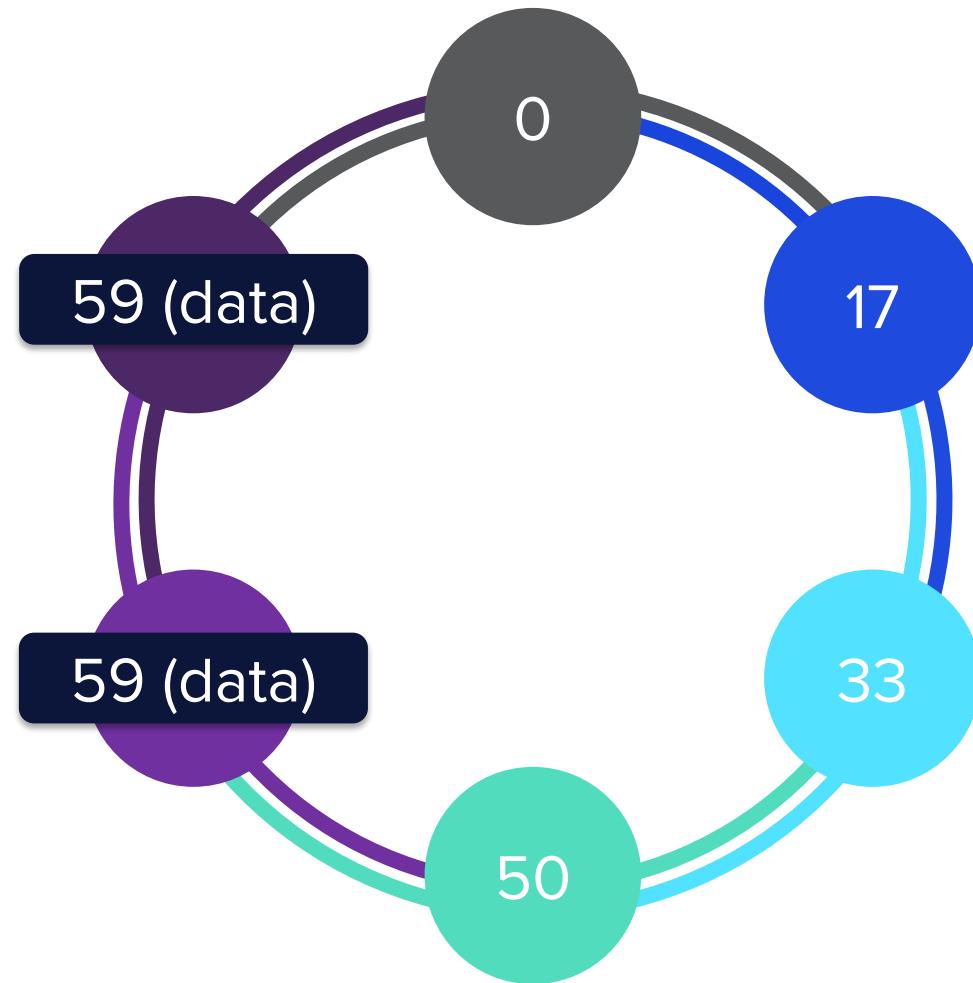
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



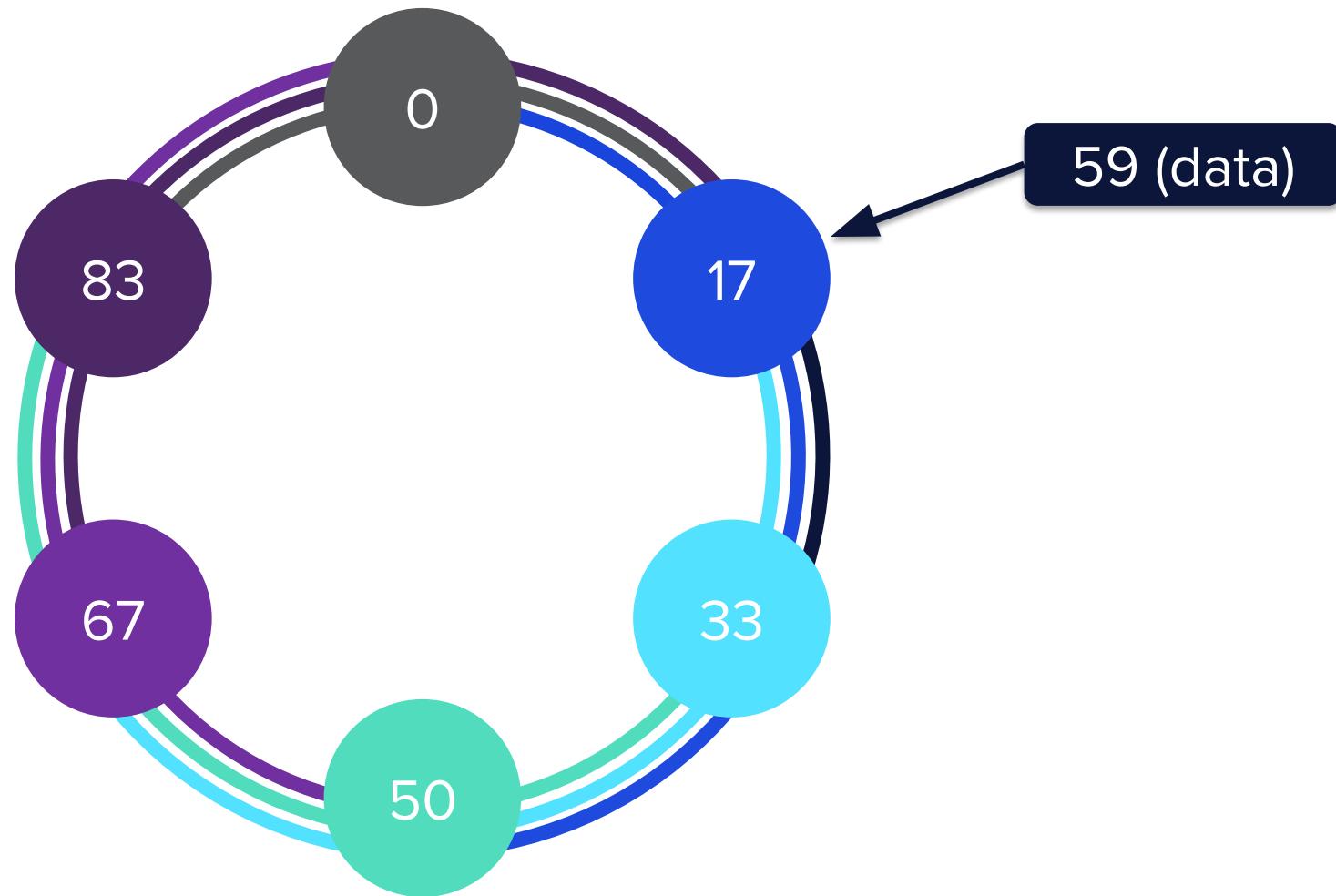
# Replication within the Ring

RF = 2



# Replication within the Ring

RF = 3



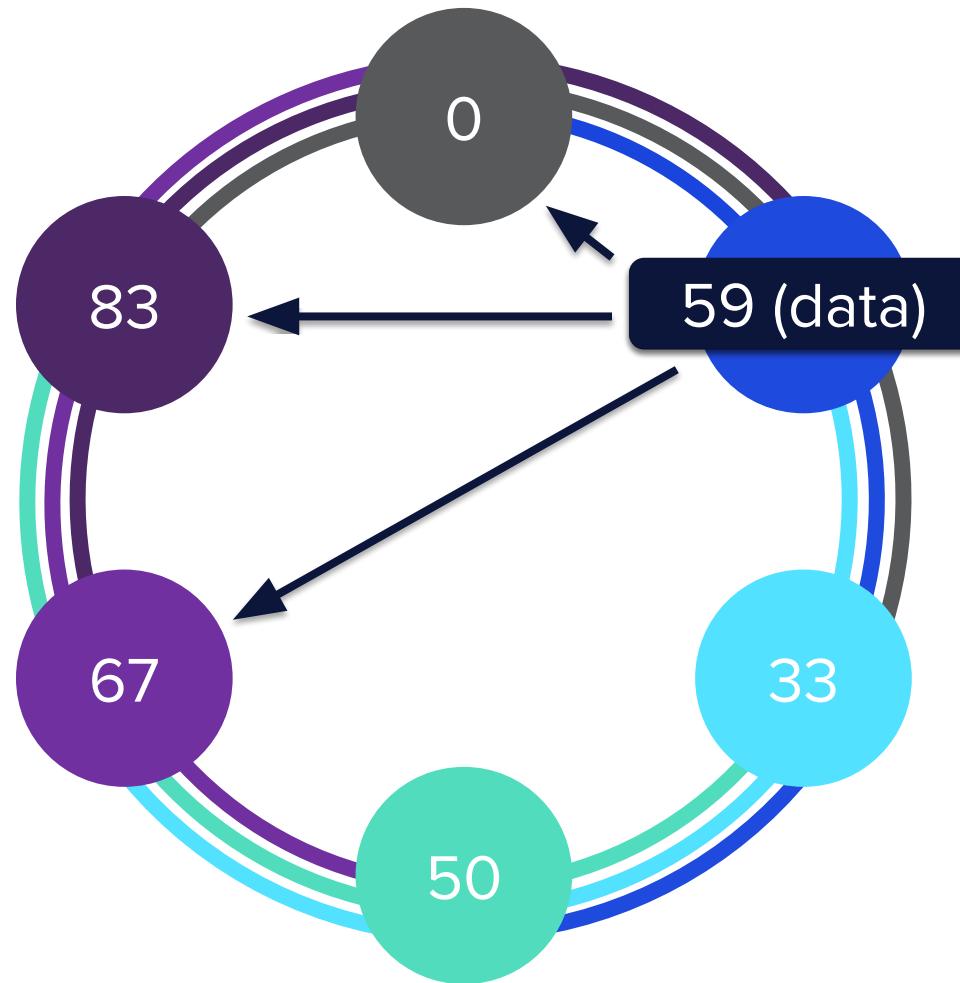
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



# Replication within the Ring

RF = 3



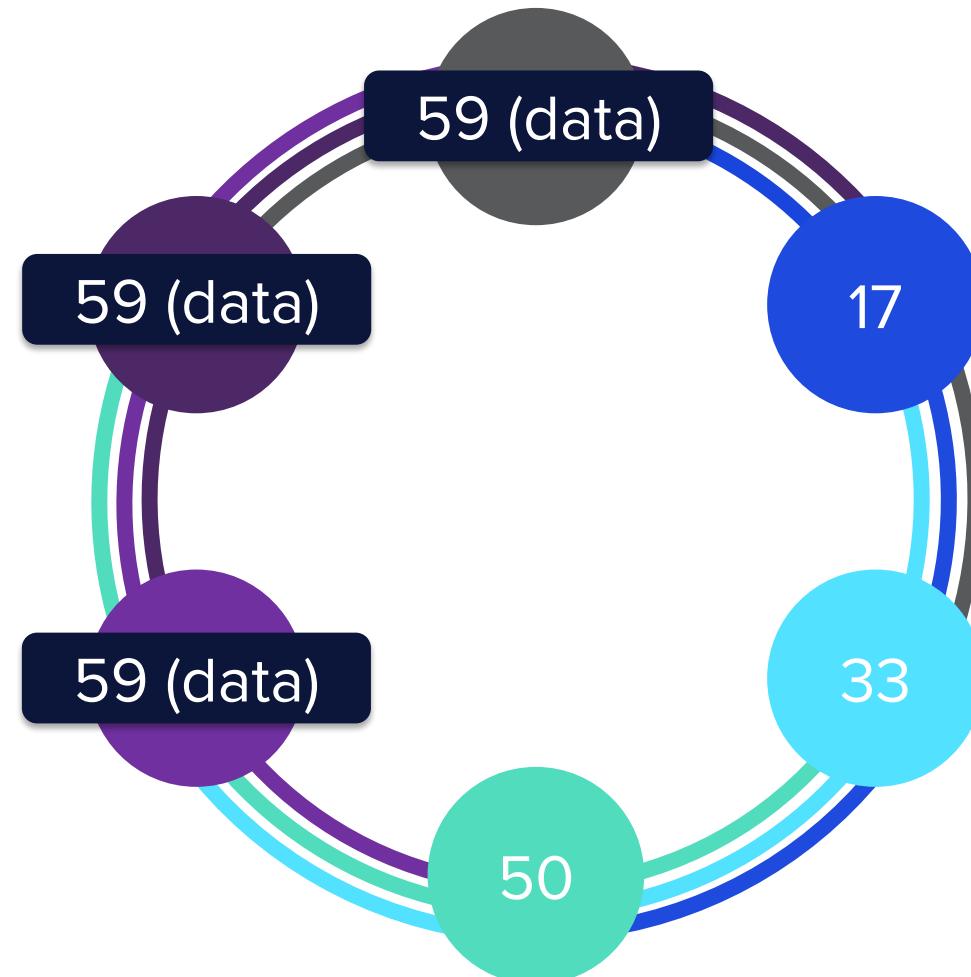
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



# Replication within the Ring

RF = 3



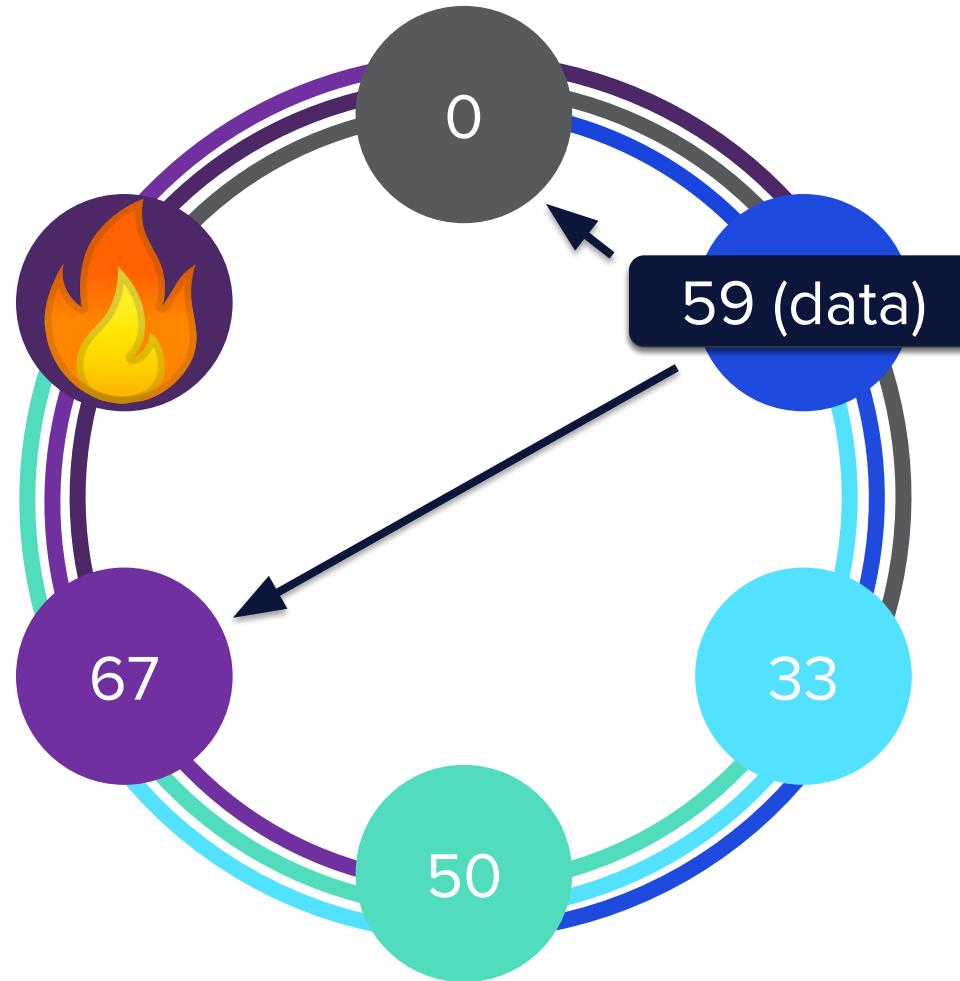
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



# Node Failure

RF = 3



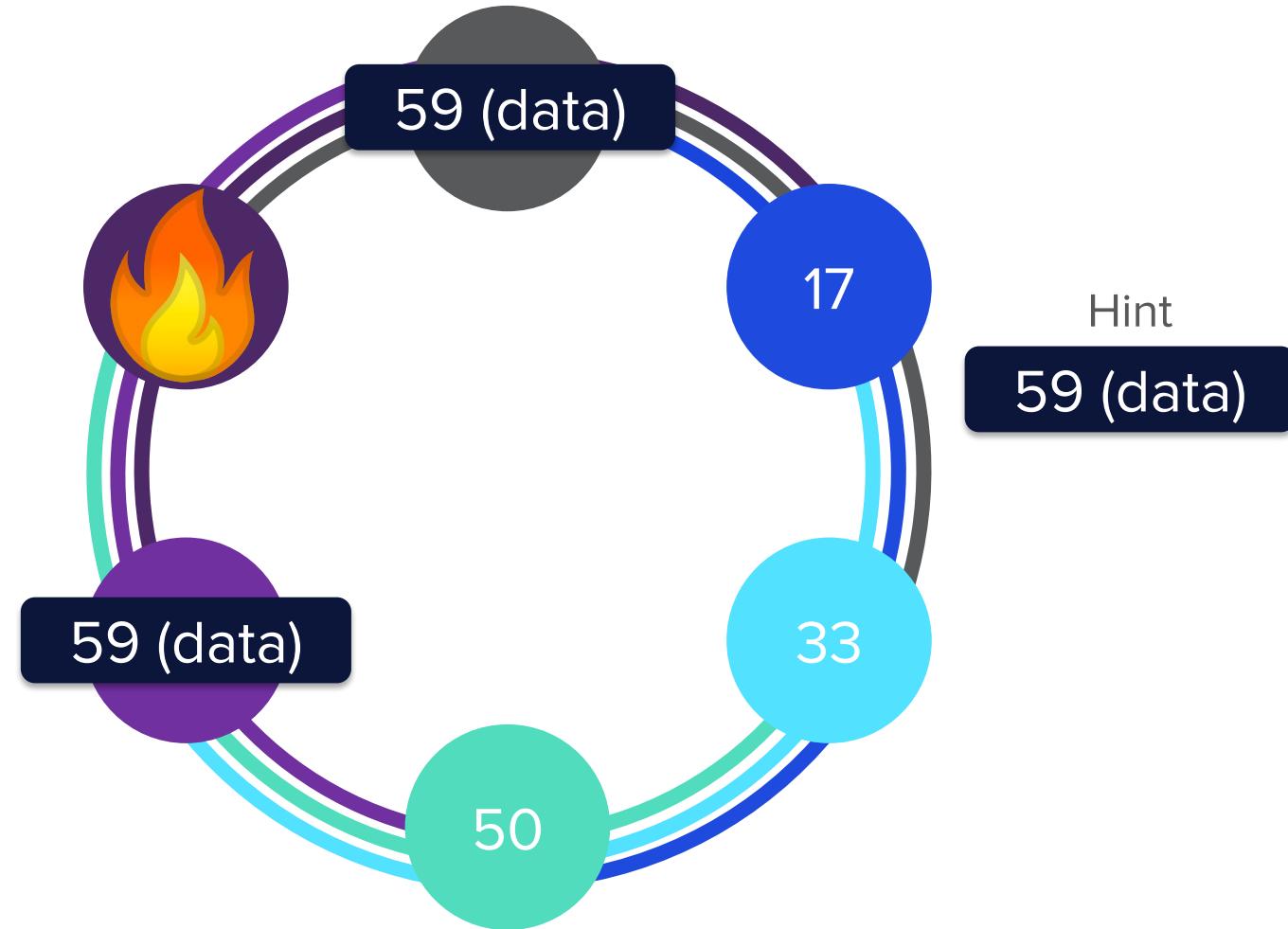
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



# Node Failure

RF = 3



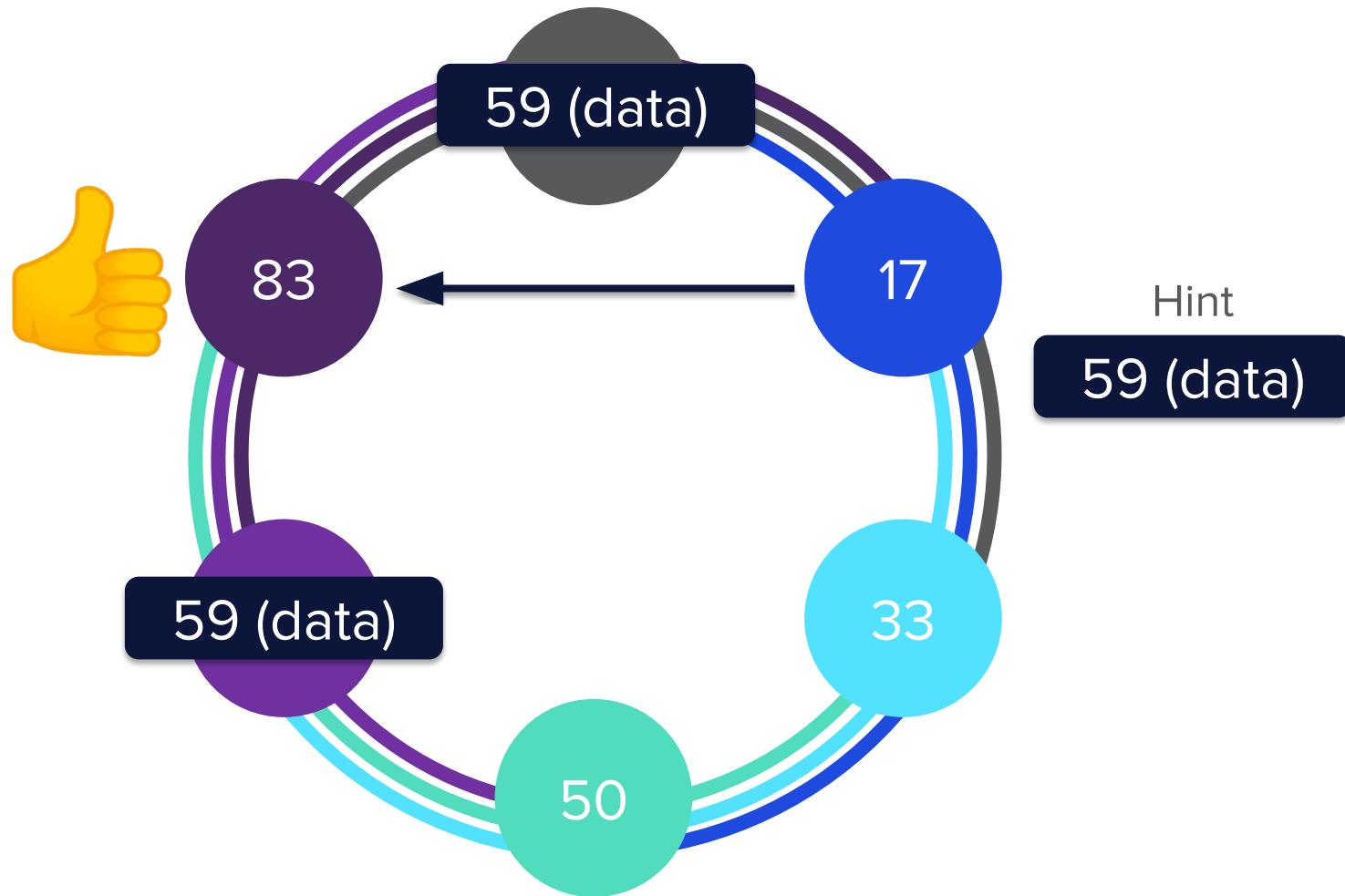
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



# Node Failure

RF = 3



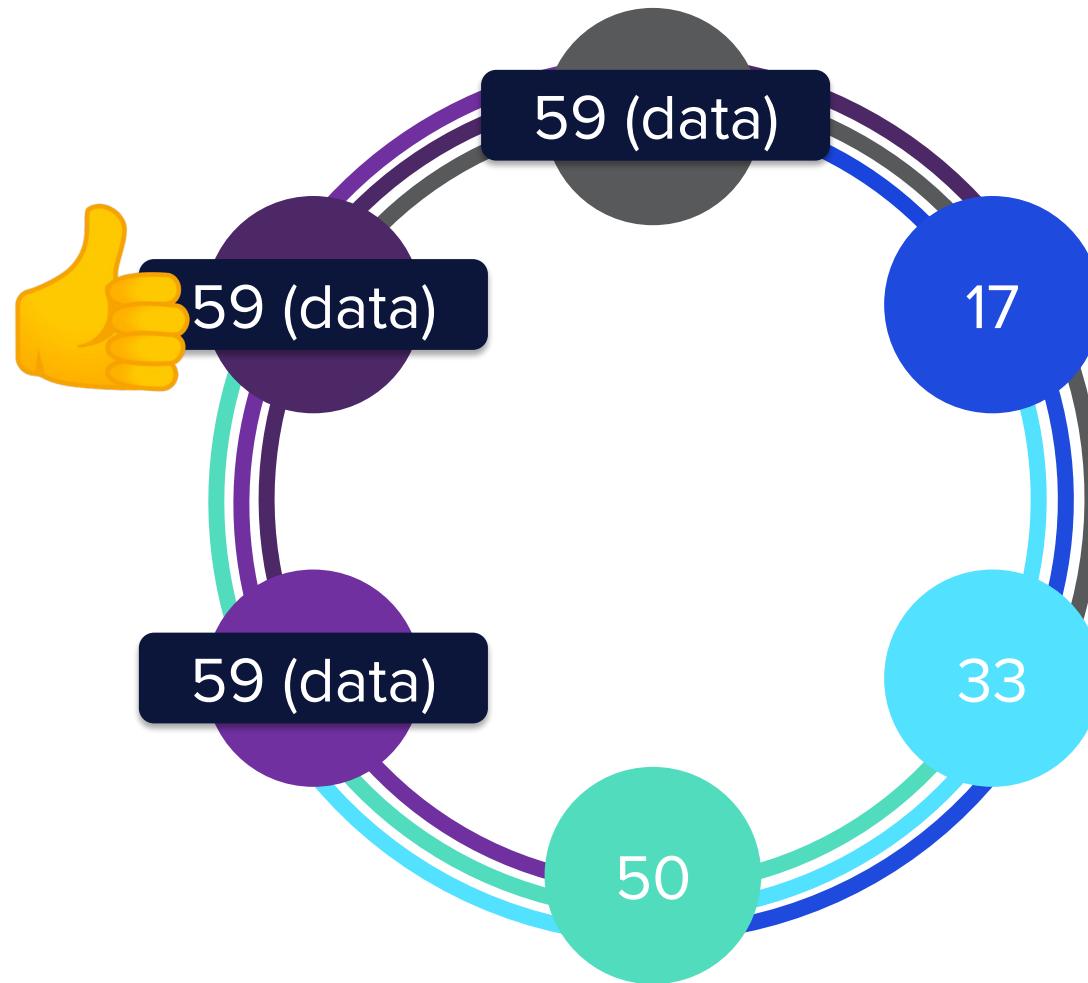
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



# Node Failure – Recovered!

RF = 3



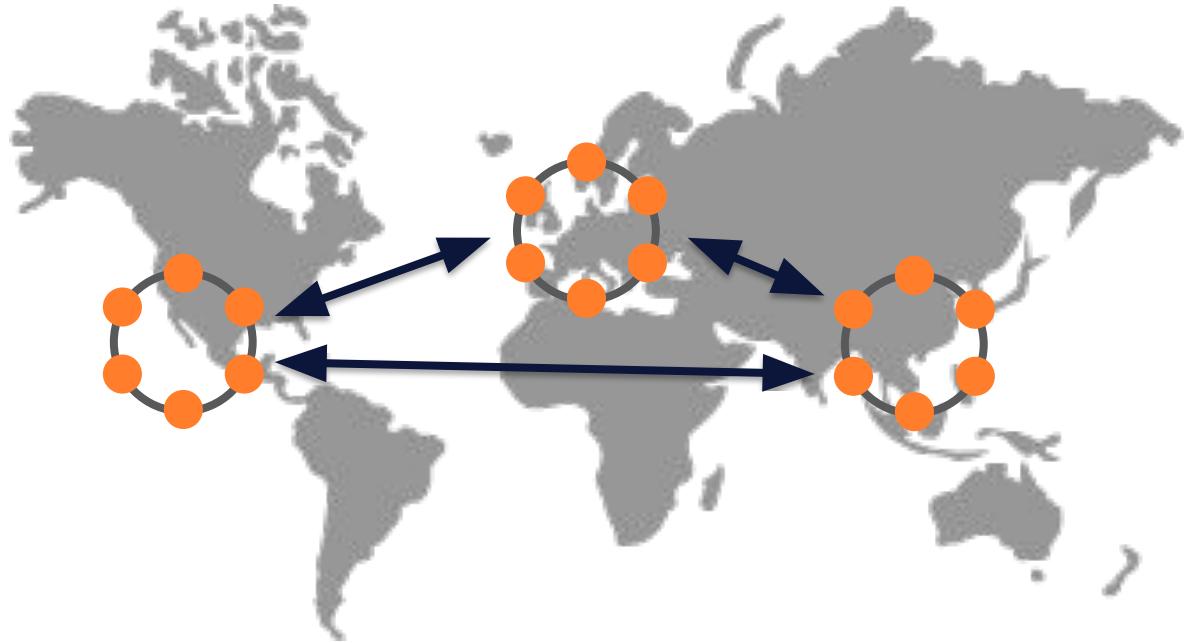
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>

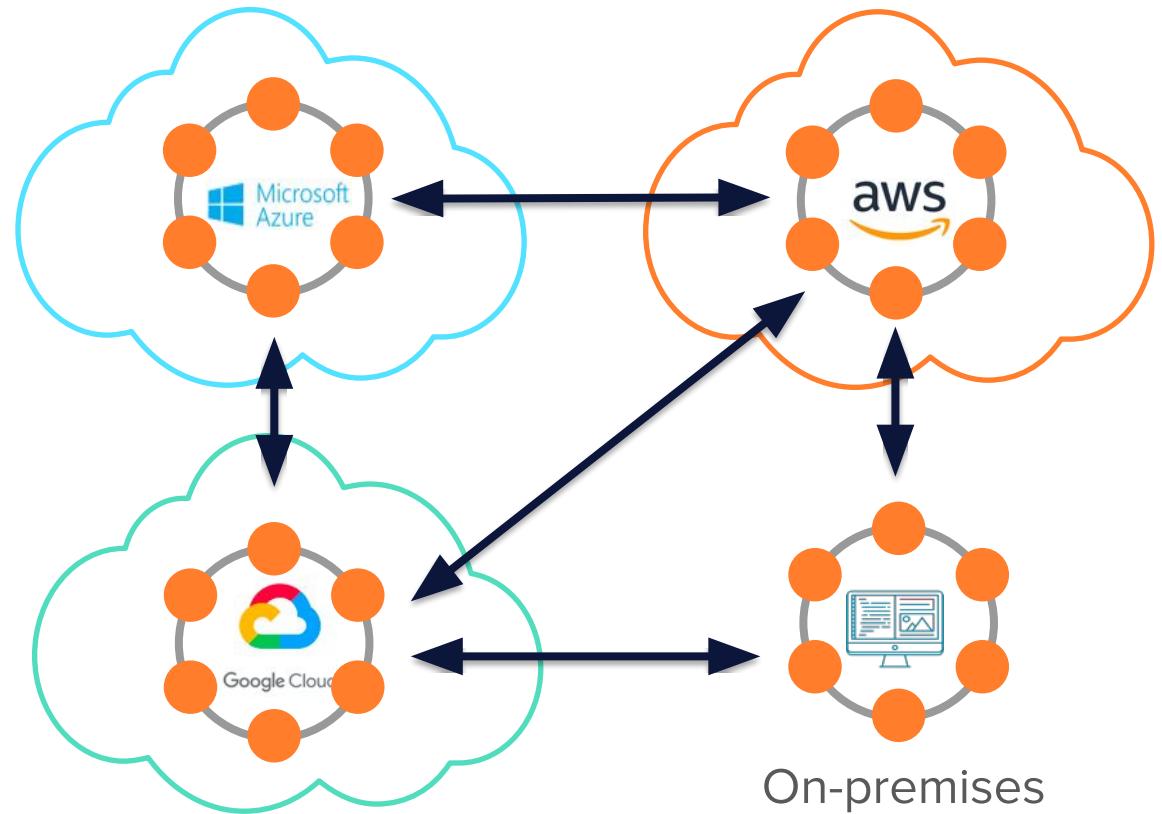


# Data Distributed Everywhere

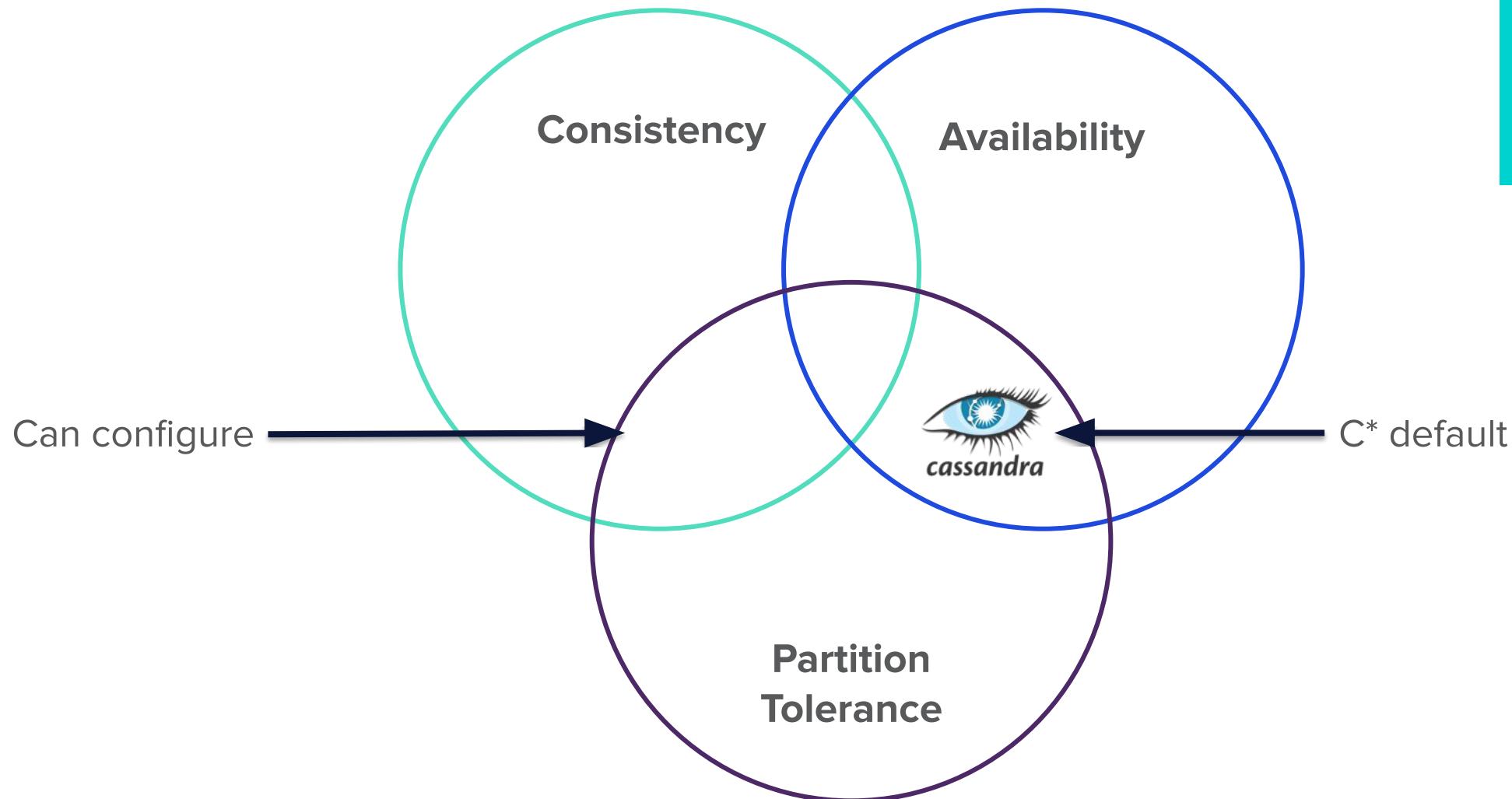
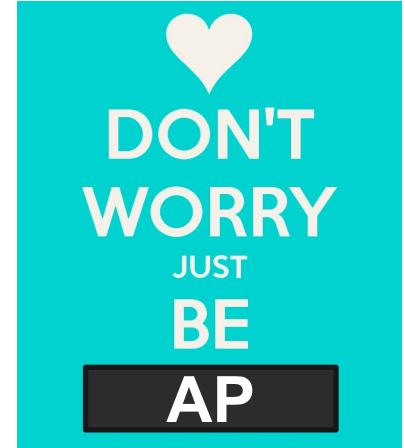
- Geographic Distribution



- Hybrid-Cloud and Multi-Cloud



# Cap Theorem

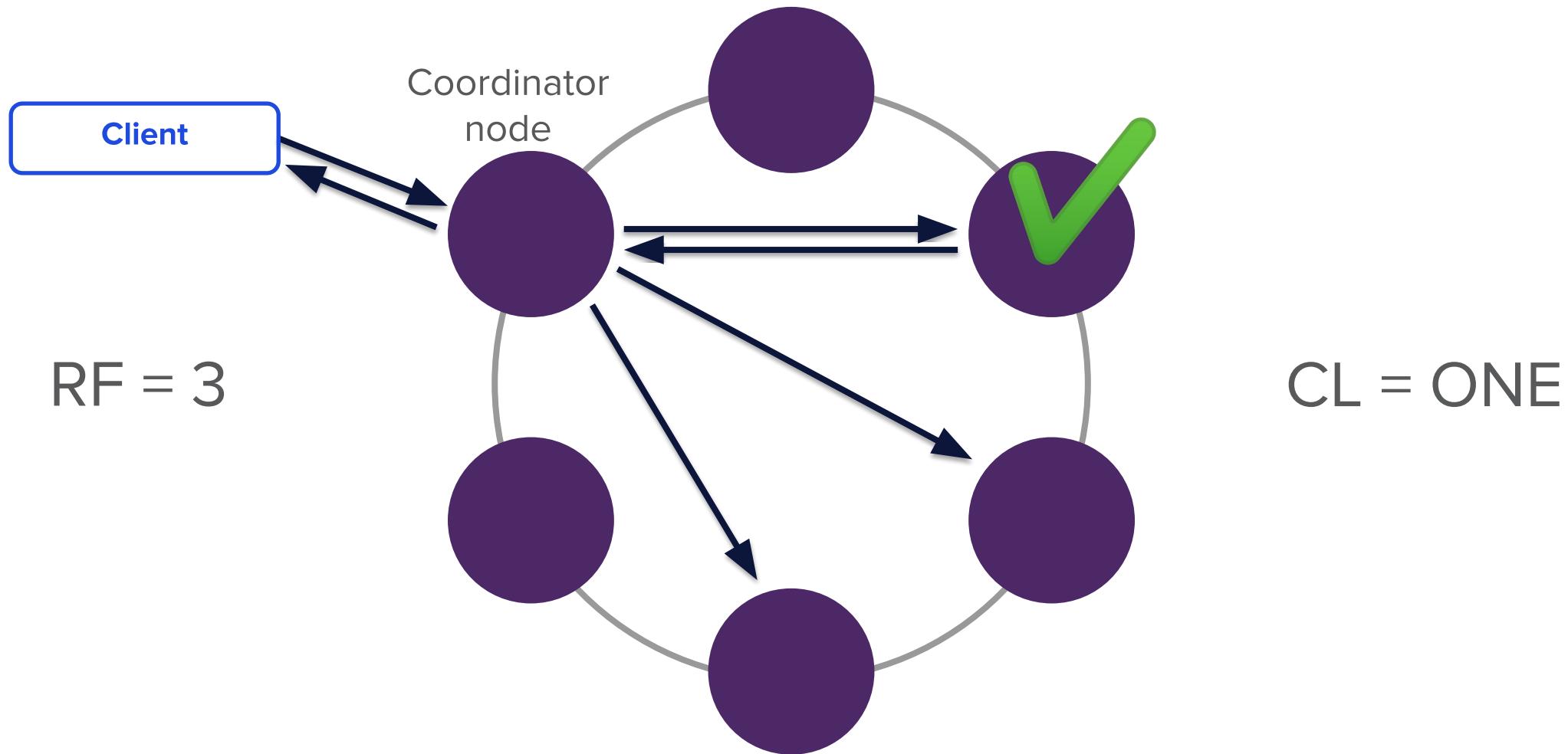


@DataStaxDevs #DataStaxDeveloperDay

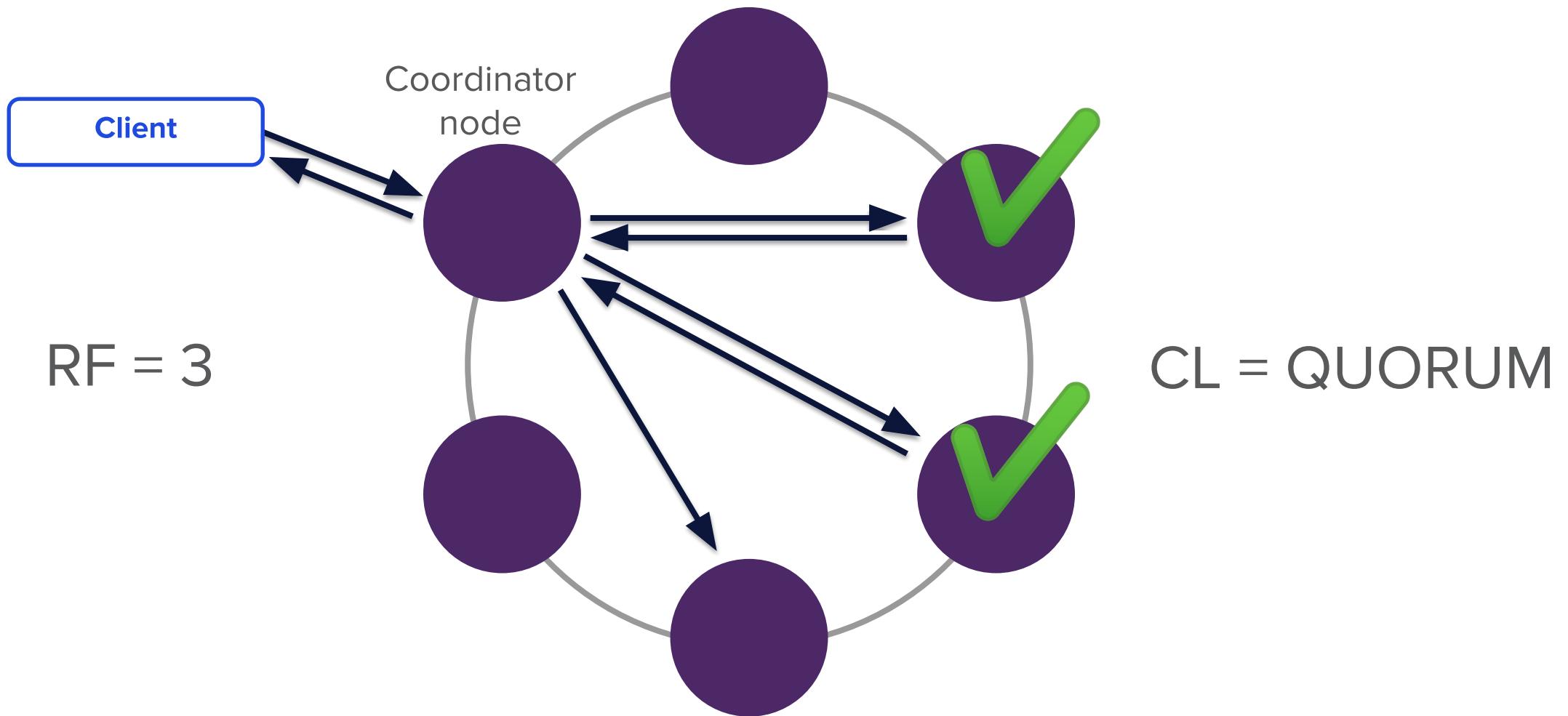
<https://community.datastax.com>



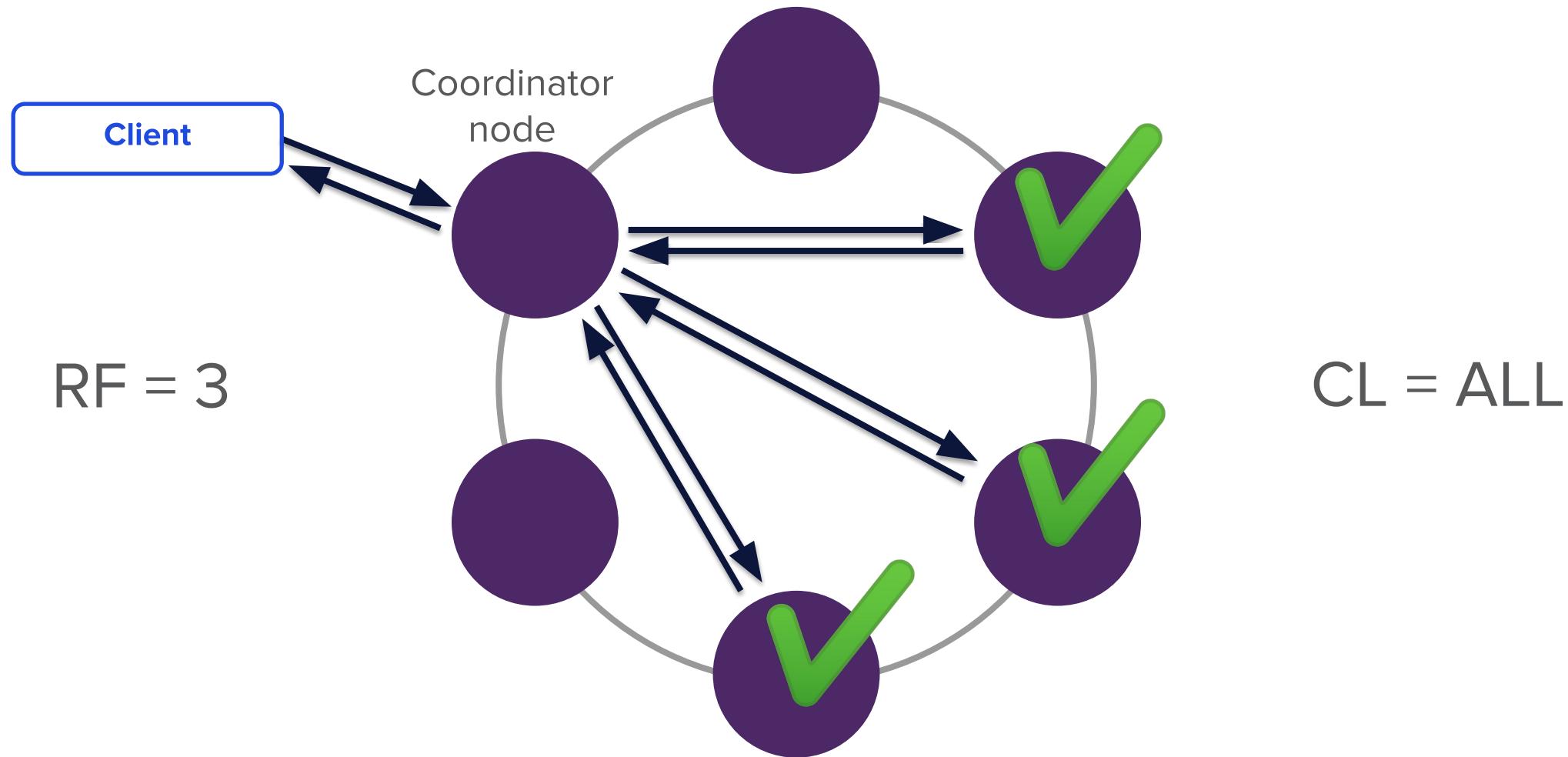
# Consistency Levels



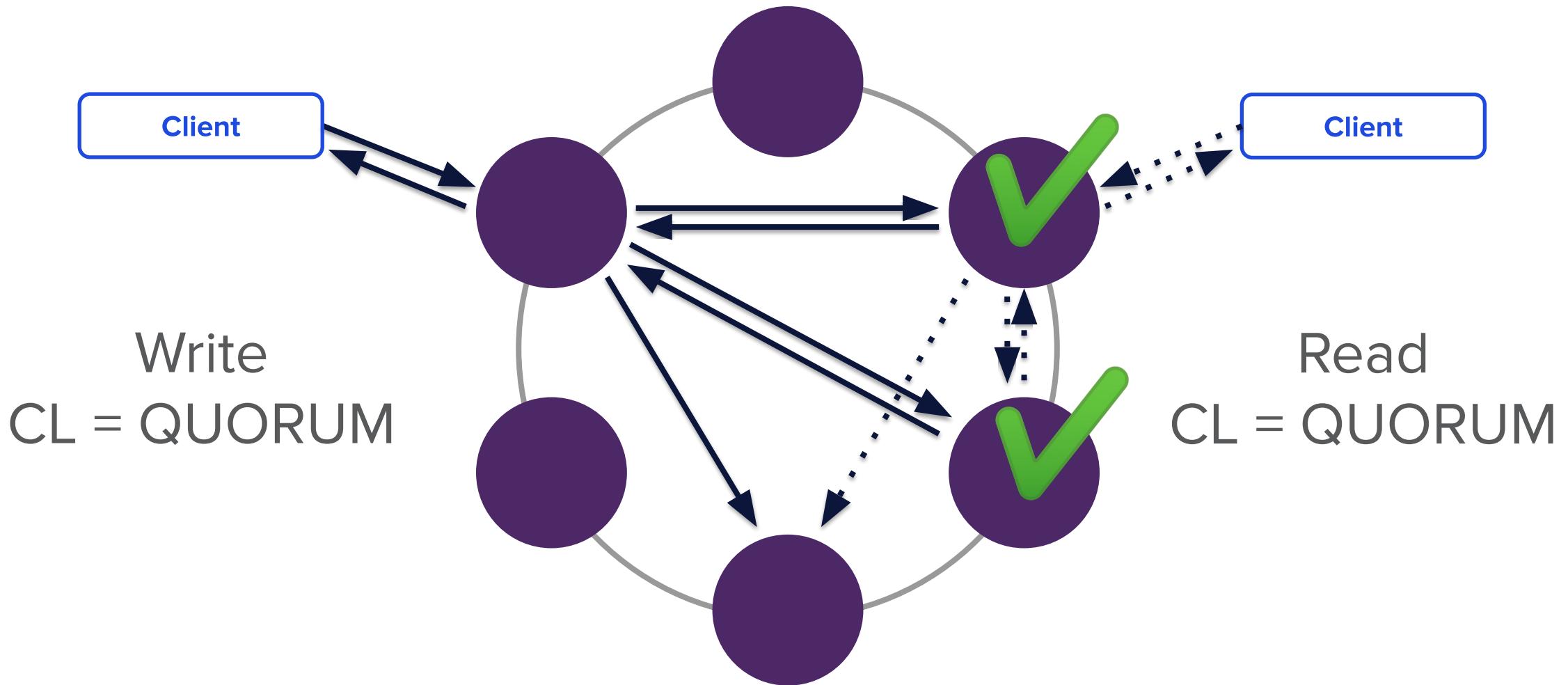
# Consistency Levels



# Consistency Levels



**Immediate Consistency =  $CL_{read} + CL_{write} > RF$**



# 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	

Availability

Mission-Critical
------------------



No Data Loss
Always-on



Caching	
Market Data	Pricing
Inventory	
Inventory	

Distributed

Global Presence
Workload Mobility



Compliance /
GDPR



Banking	
Tracking / Logistics	Retail
Customer Experience	
Customer Experience	

Cloud-native

Modern Cloud Applications
---------------------------



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



@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



# menti.com

# 36 63 61



Available on the iPhone  
**App Store**

GET IT ON  
**Google play**

# Apache Cassandra™ with Kubernetes

1

**Cassandra Why, What, When**

2

**Kubernetes Reminders + Setup**

3

**Cassandra Operator + *Hands-On***

4

**Dashboard UI + *Hands-On***

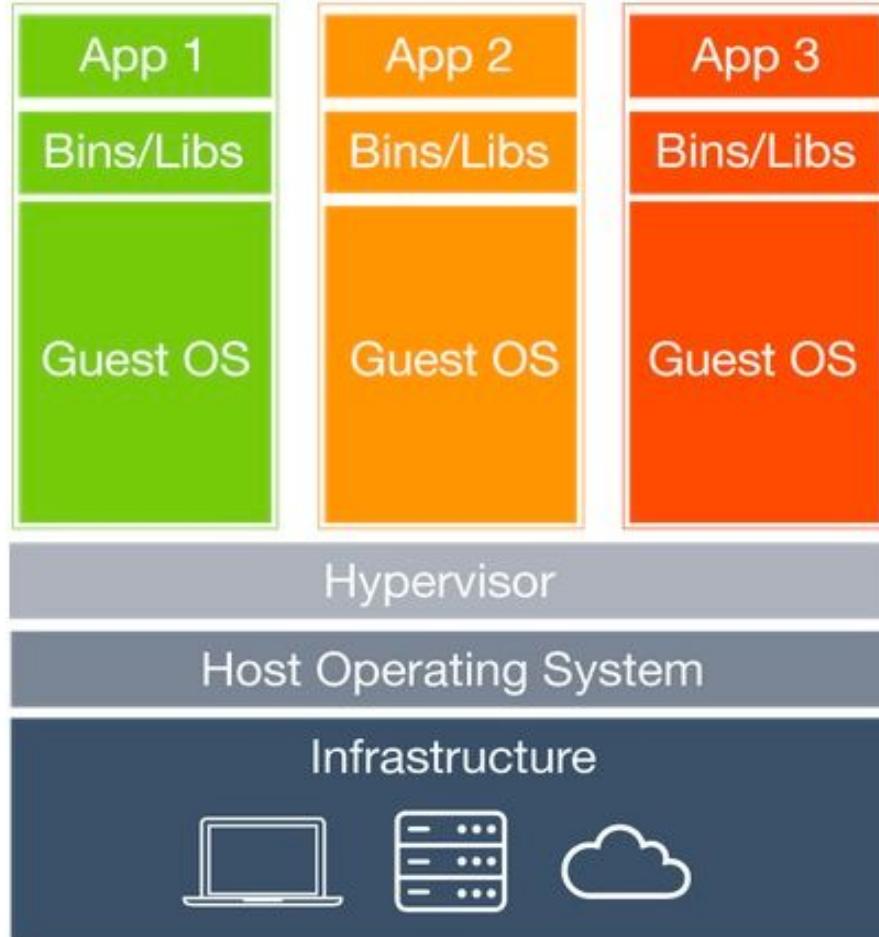
5

**Grafana Prometheus + *Hands-On***

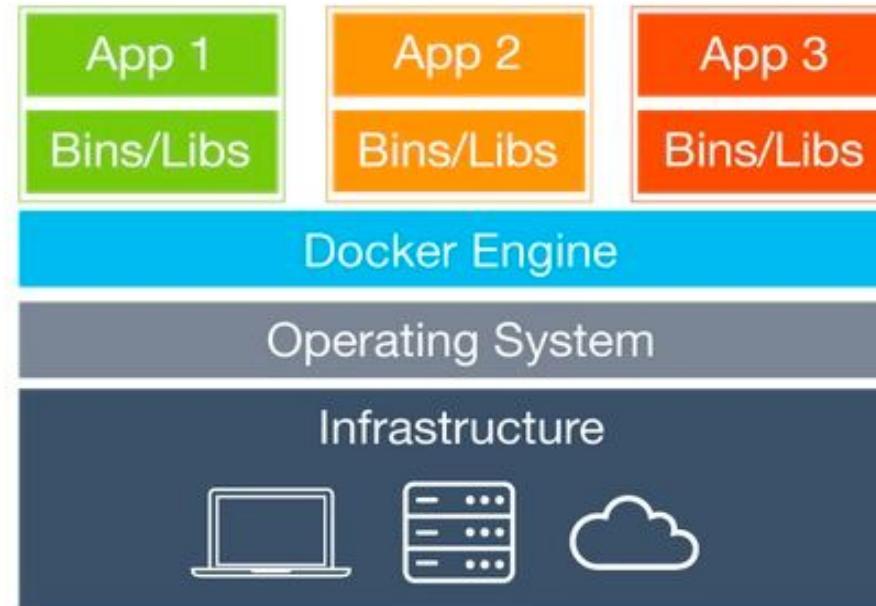
6

**Resources**

# Containers

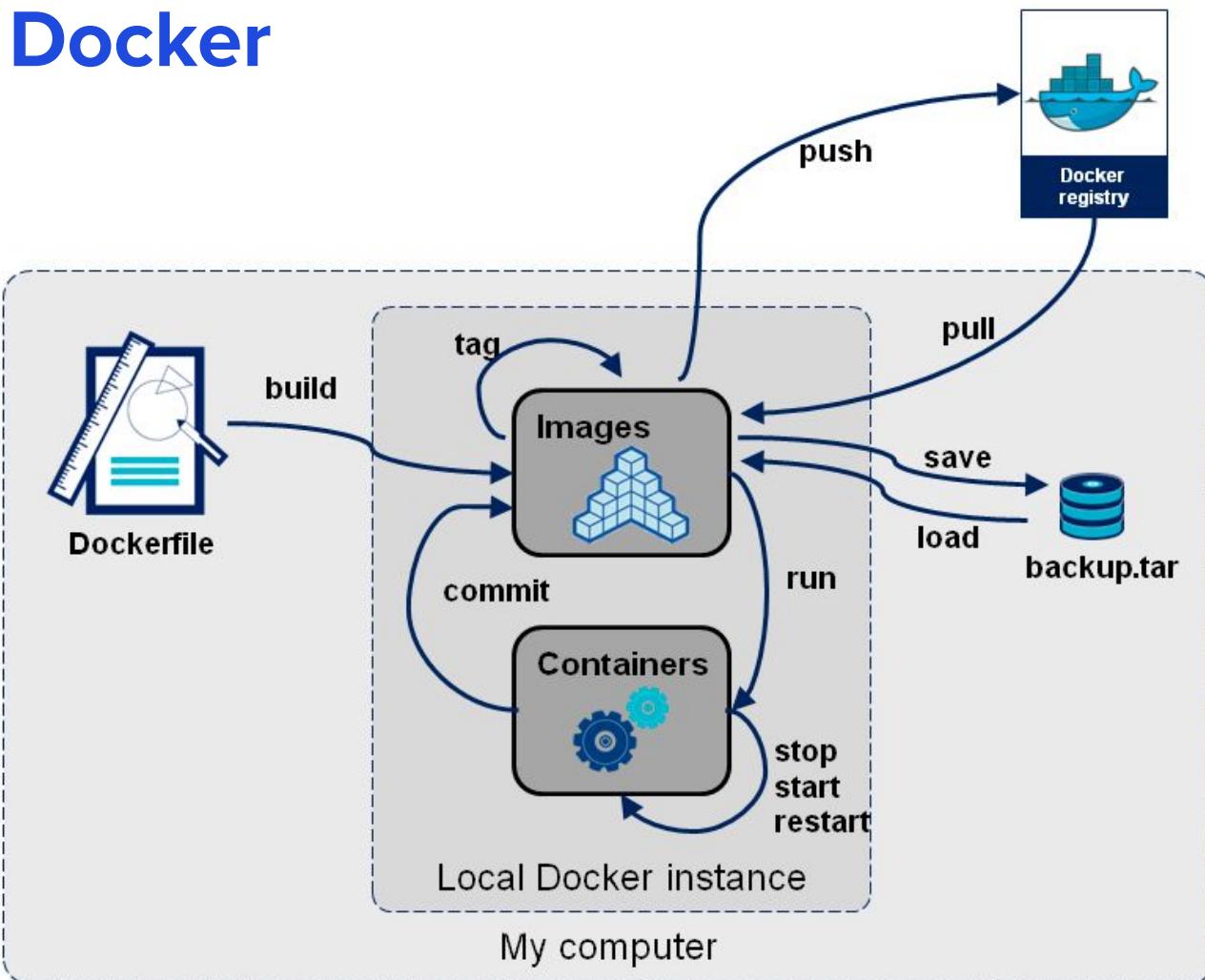


**Virtual Machines**



**Containers**

# Docker



## WHAT

- One container implementation among other : rkt (coreOS), LXC, Linux Vserver, runC, openVZ
- Lifecycle and tooling of containers

## Docker ... (and DB)

- Default behaviour: lost all data at restart
- Persistent storage = Volumes
- Network is explicit (link, docker-network)
- Security is absolute (port mapping)

Tutorial : <https://github.com/clun/my-notebook/blob/master/DOCKER.MD>

# Apache Cassandra™ in Docker

[https://hub.docker.com/\\_/cassandra/#!](https://hub.docker.com/_/cassandra/#!)



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](#)

# Cassandra in docker-compose

```
version: '2'

# 3 node cluster
# If you see exit code 137 (OOM killer) then ensure Docker has access to more resources
services:
  cassandra-seed:
    container_name: cassandra-seed-node
    image: cassandra:3.11.0
    ports:
      - "9042:9042" # Native transport
      - "7199:7199" # JMX
      - "9160:9160" # Thrift clients

  cassandra-node-1:
    image: cassandra:3.11.0
    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"

  # you cannot have multiple nodes join the cluster at the same time when
  # cassandra.consistent.rangemovement is true so we further delay it to give it time to stabilize
  cassandra-node-2:
    image: cassandra:3.11.0
    command: /bin/bash -c "echo 'Waiting for seed node' && sleep 80 && /docker-entrypoint.sh cassandra -f"
    environment:
      - "CASSANDRA_SEEDS=cassandra-seed-node"
    depends_on:
      - "cassandra-seed"
```

```
docker-compose -f docker-compose.yml up
-d --scale node=1
```

<https://github.com/calvinlfer/compose-cassandra-cluster/blob/master/docker-compose.yml>



# kubernetes

# Kubernetes (k8s) Infrastructure



**Cluster:**  
Kubernetes cluster.



**Master:**  
Kubernetes Control Plane.



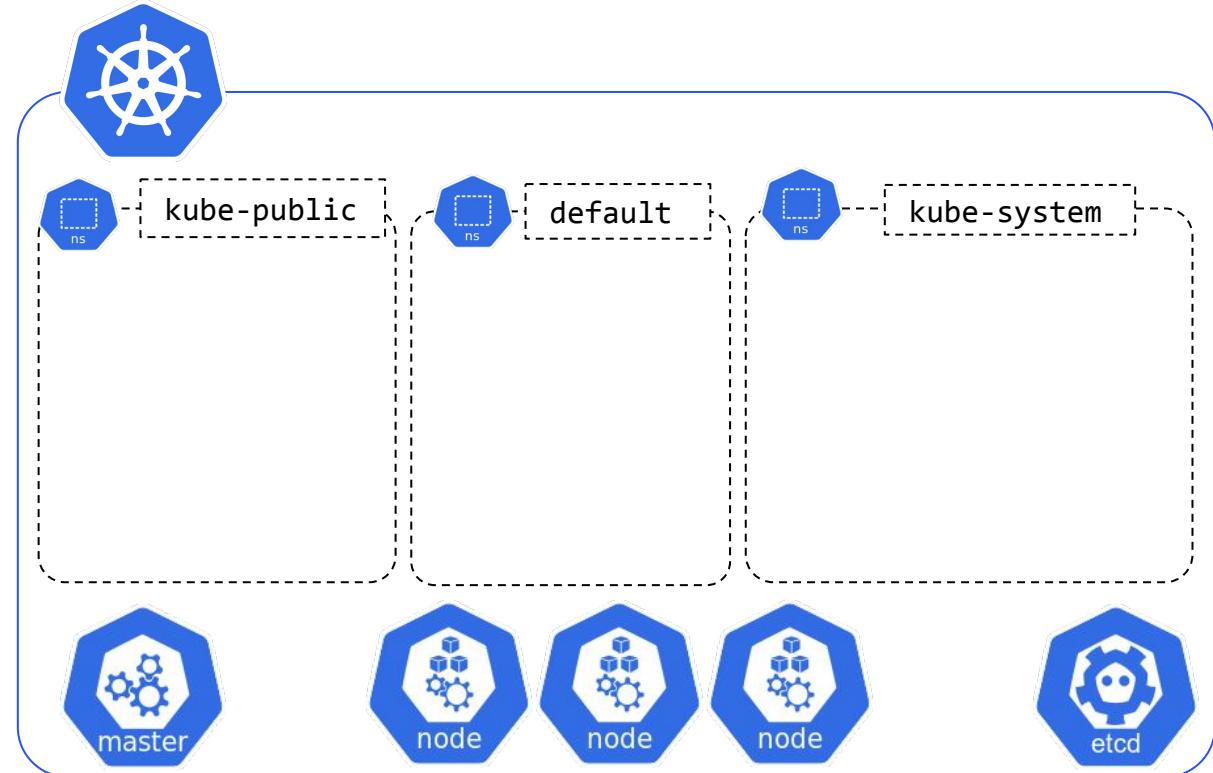
**Node:**  
Worker machine in Kubernetes cluster.



**ETCD**  
Kubernetes's backing store.



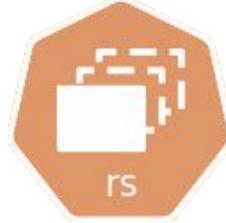
**Namespace:** Namespace provides a scope for Names. Use of multiple namespaces is optional.



# K8s Primitives : Compute



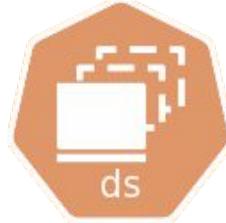
**POD** is a collection of containers that can run on a host. This resource is created by clients and scheduled onto hosts.



**ReplicaSet** ensures that a specified number of pod replicas are running at any given time.



**DaemonSet:** DaemonSet represents the configuration of a daemon set.



**Deployment:** Deployment enables declarative updates for Pods and ReplicaSet.



**Job:** Job represents the configuration of a single job.



**CronJob:** A CronJob manages time based Job, namely: once at a specified point in time repeatedly at a specified point in time

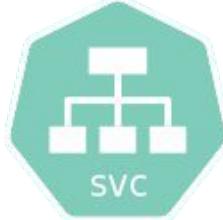


**StatefulSet:** StatefulSet represents a set of pods with consistent identities. Identities are defined as: network, storage.

# K8s Primitives : Network



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



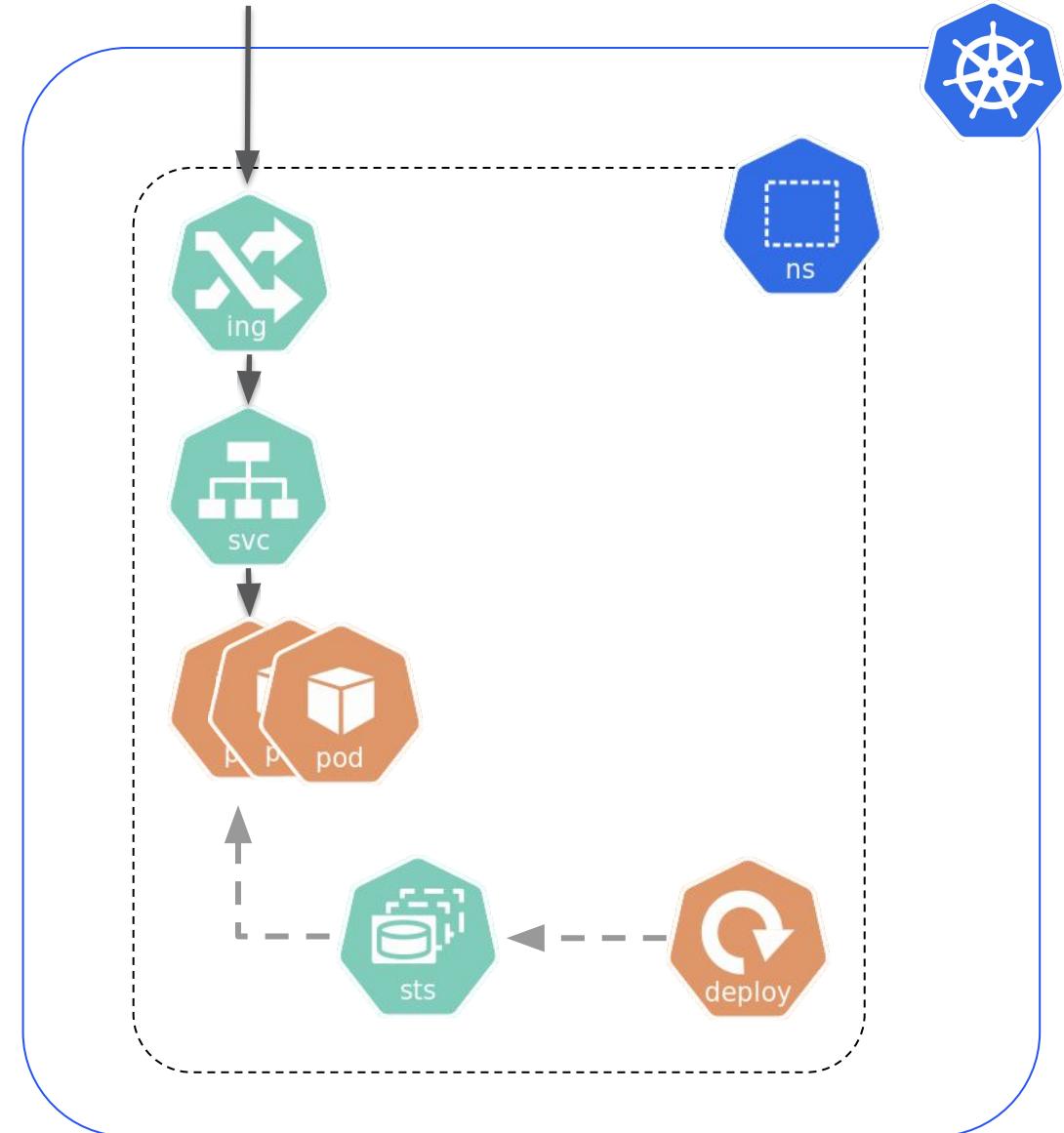
**Service** is a named abstraction of software service with ports to listen on and selector to determine which pods will answer requests.



**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 : Storage



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

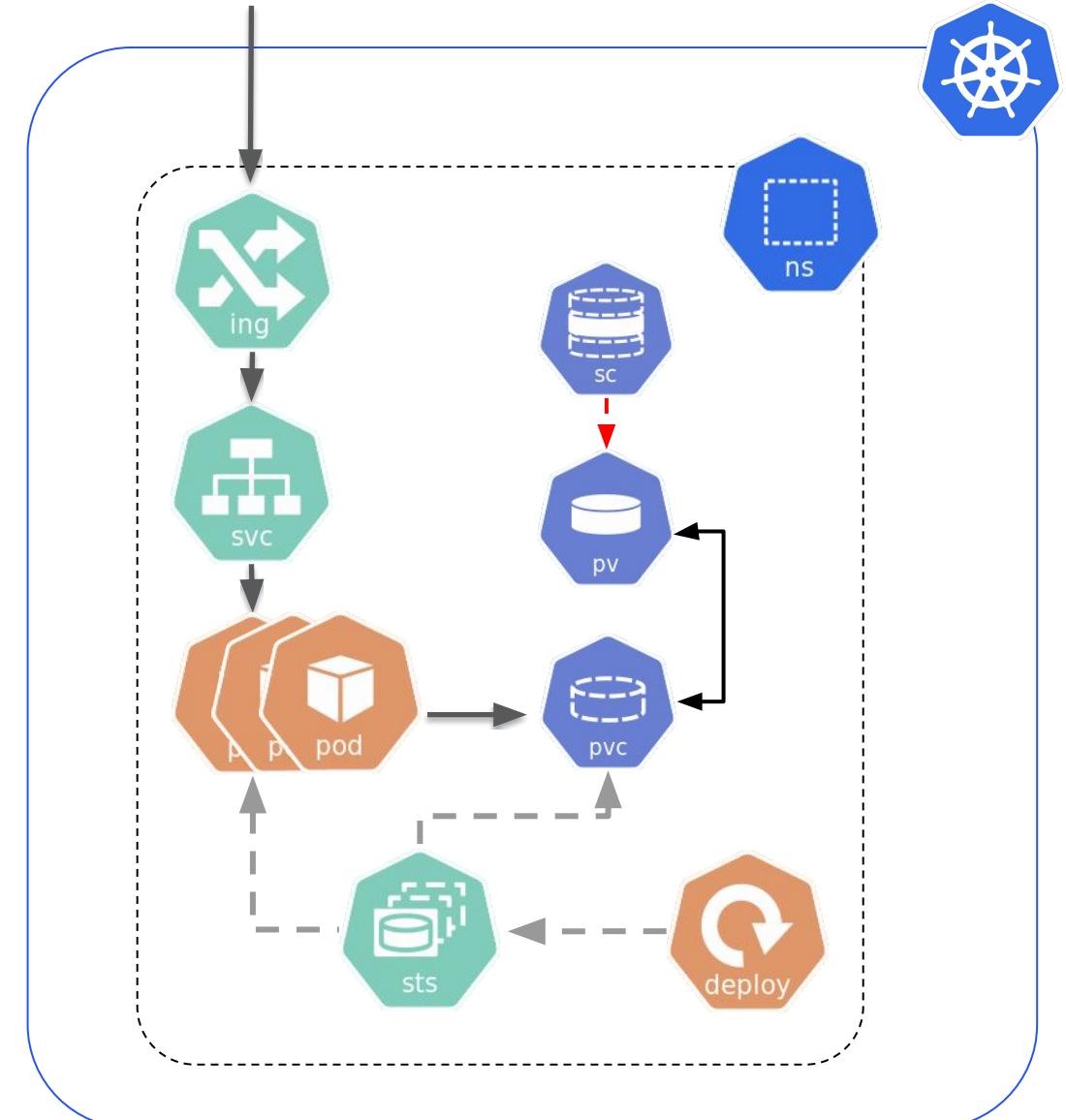


**PersistentVolumeClaim:**

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.



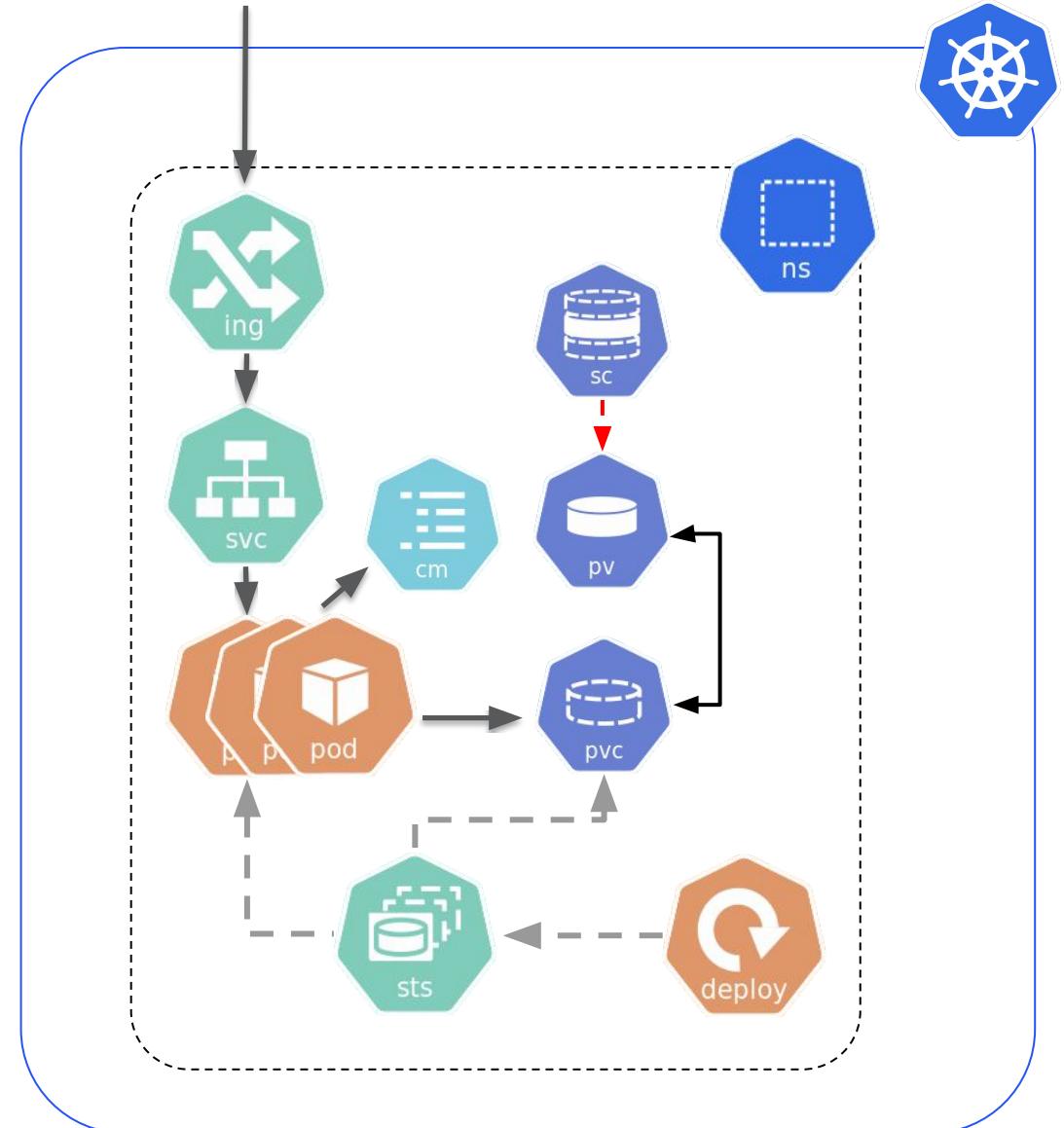
# K8s Primitives : Configuration



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



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



# Environment and Services



Azure Kubernetes Service (AKS)



Amazon EKS



Google Kubernetes Engine (GKE)



## Cloud



**Red Hat**  
CodeReady  
Containers



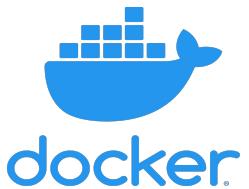
DATASTAX®

## Local Environment



# YOUR LAPTOP

Your Laptop



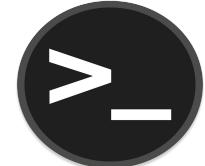
docker



kind



kubernetes



# CLOUD INSTANCE



Cloud Env



docker



kubernetes



kind



CentOS



# Online Workshops



## Section #1 : Setup Your Cluster

<https://github.com/DataStax-Academy/kubernetes-workshop-online/blob/master/0-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 from now all command will be the same on each platform, as 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.

# Apache Cassandra™ with Kubernetes

1

**Cassandra Why, What, When**

2

**Kubernetes Reminders + Setup**

3

**Cassandra Operator + *Hands-On***

4

**Dashboard UI + *Hands-On***

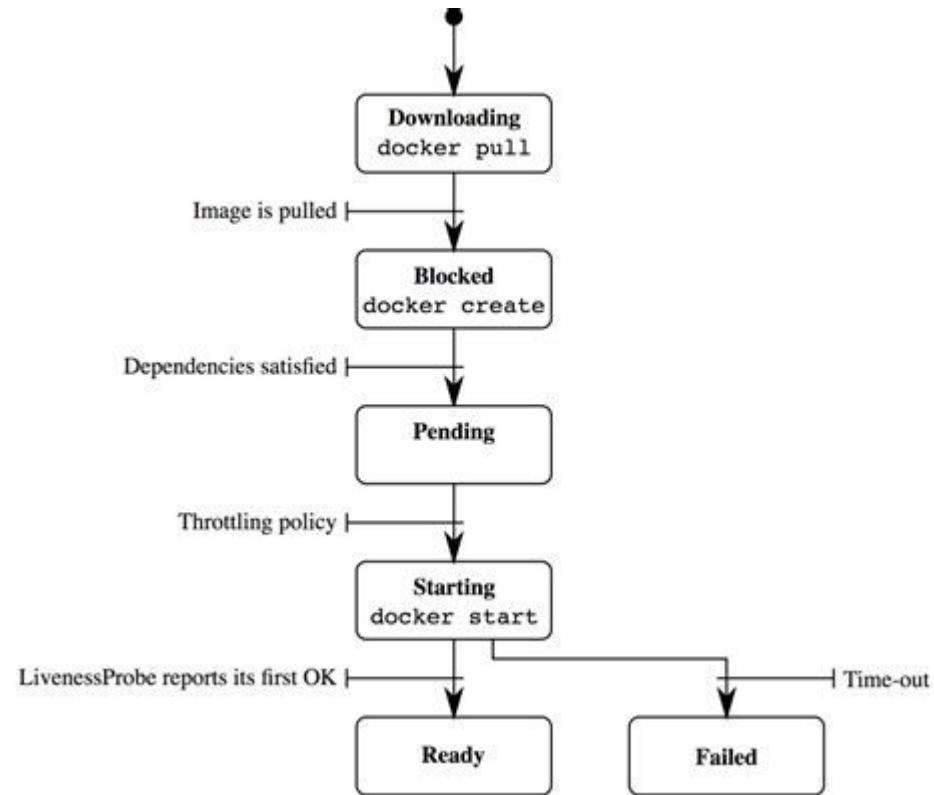
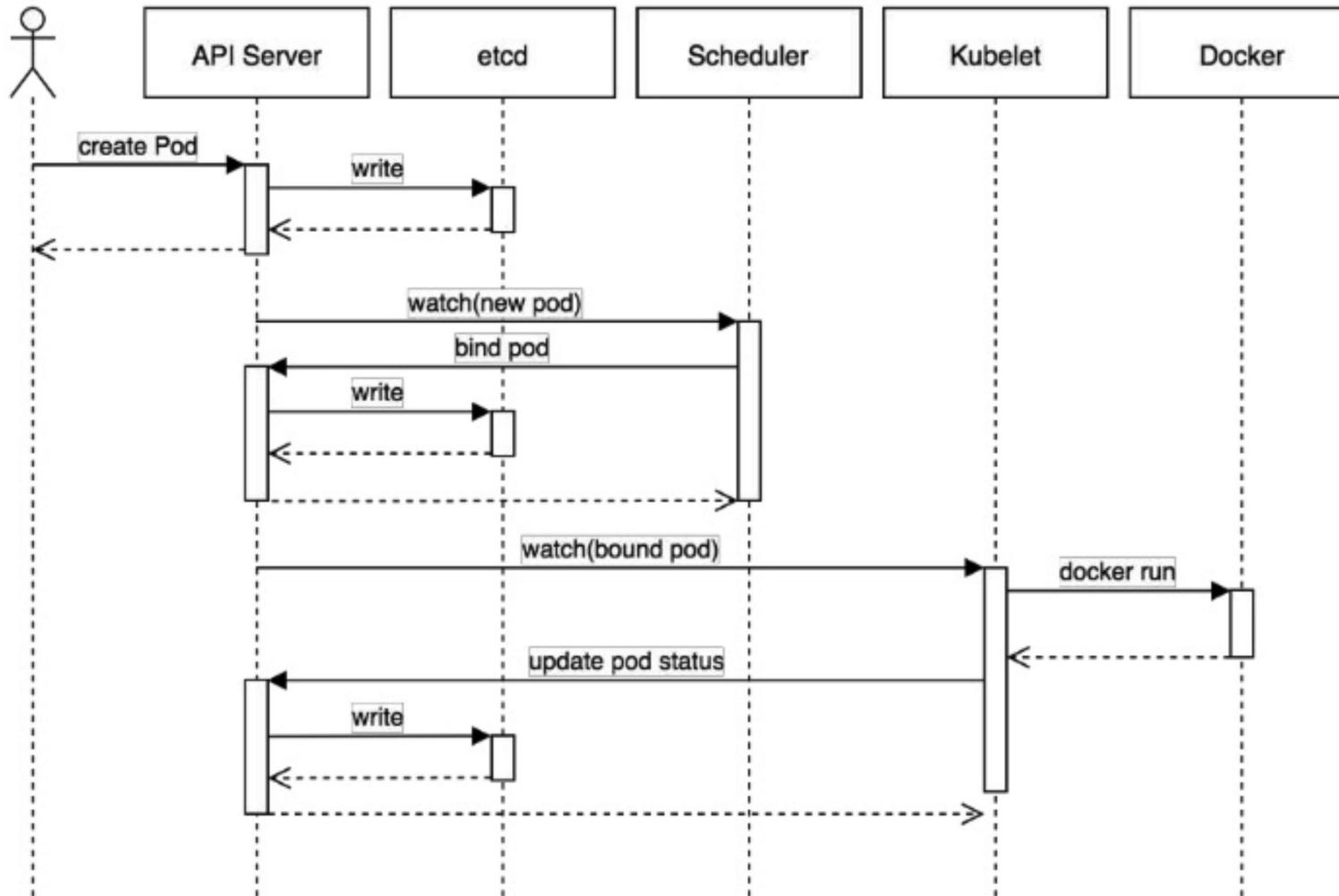
5

**Grafana Prometheus + *Hands-On***

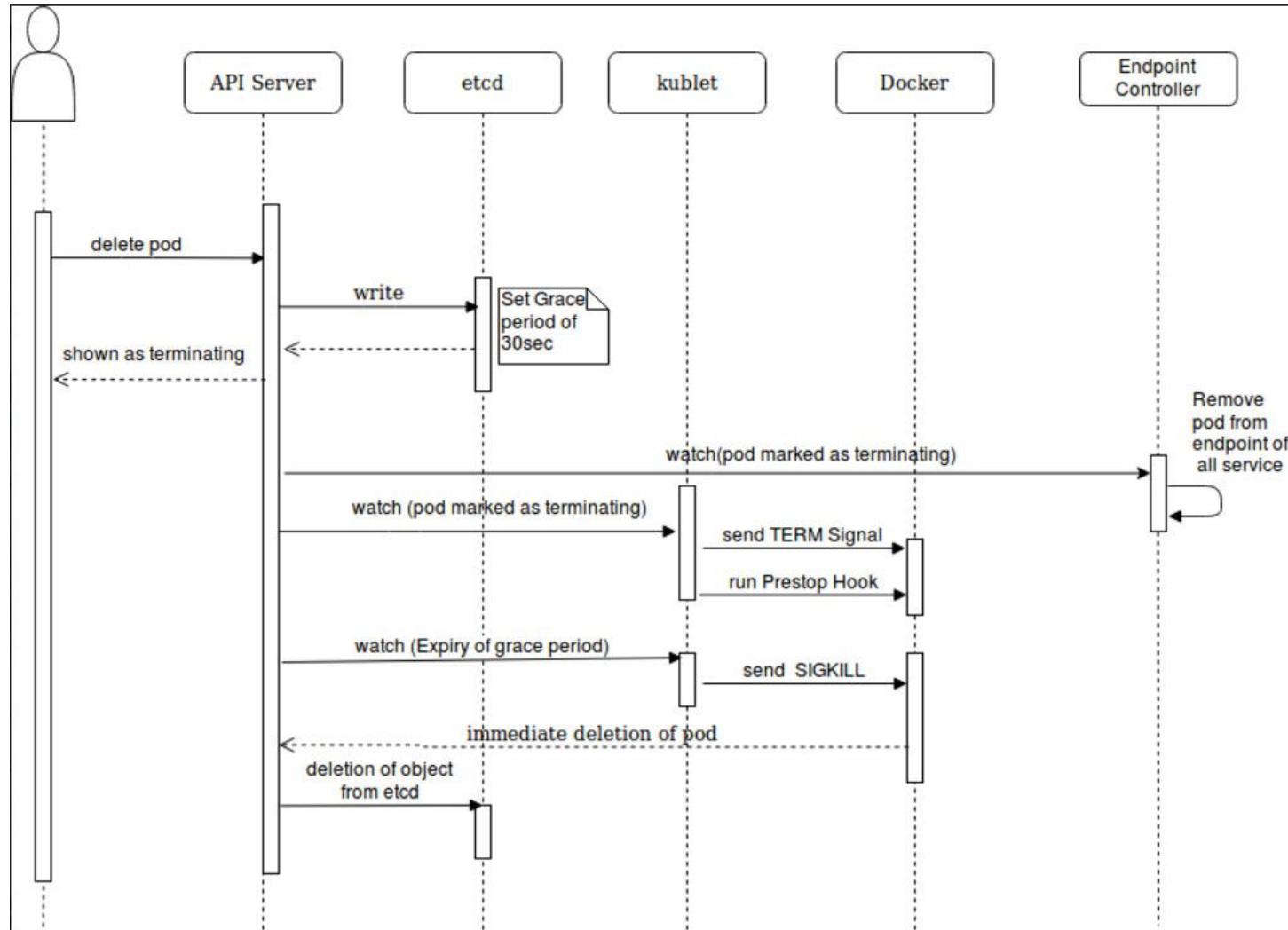
6

**Resources**

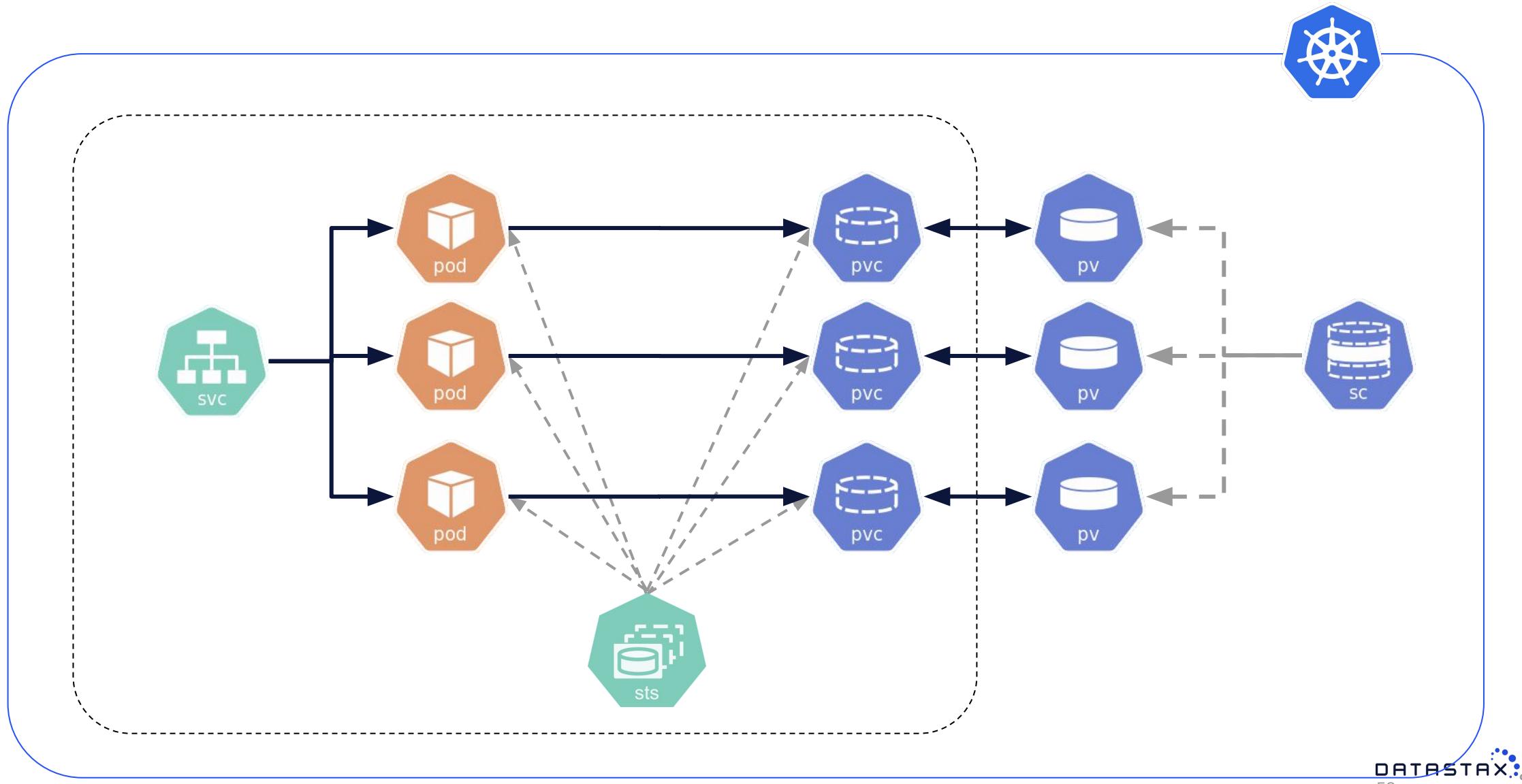
# Lifecycle of a pod (1 / 2)



# Lifecycle of a pod ( 2 / 2)



# Database deployment are stateful

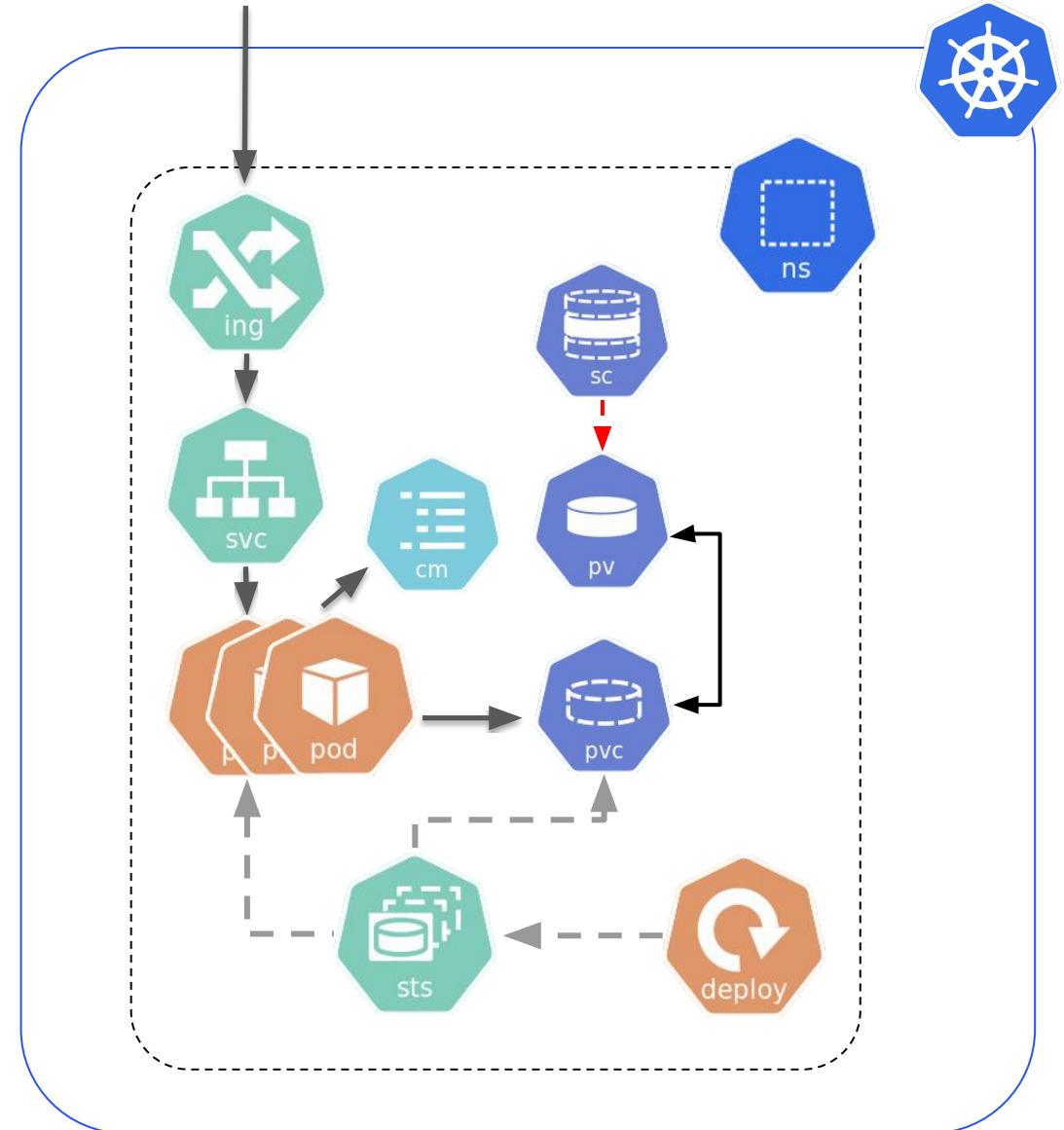


# Kubernetes: Custom Resources



`CustomResourceDefinition` :  
Extensions of the Kubernetes API.

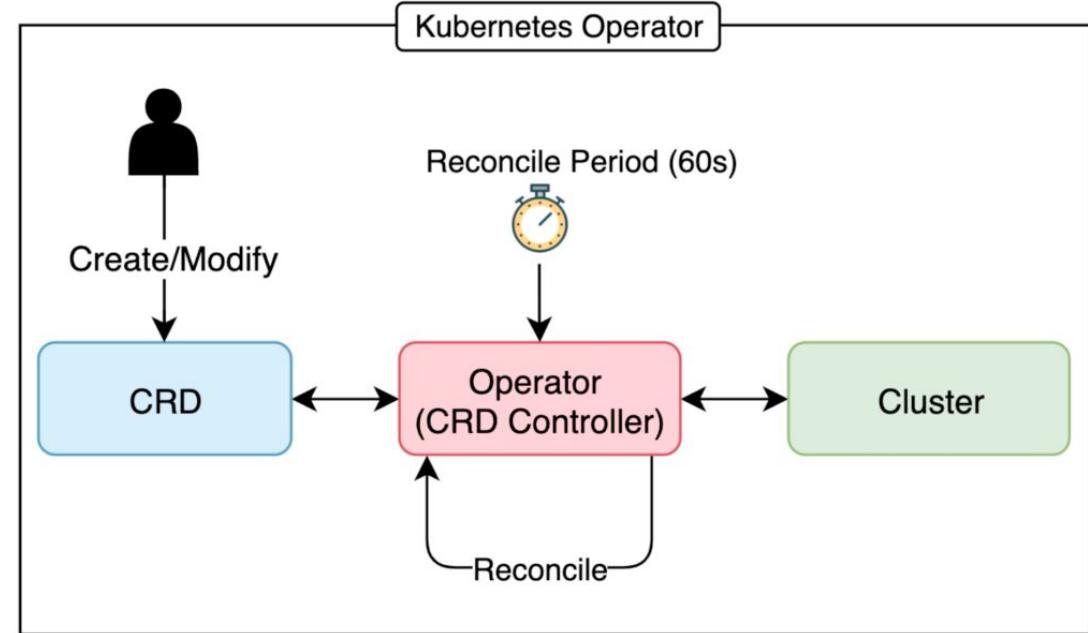
- It represents a customization of a particular Kubernetes installation.
- many core Kubernetes functions are now built using custom resources, making Kubernetes more modular.



# What is an operator ?

**Objectives :**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*





# 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)

# 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

The screenshot shows the Management API for Apache Cassandra's Swagger UI interface. At the top, it displays the title "Management API for Apache Cassandra" with a version of 0.1 and OAS3 support. Below the title, there is a brief description: "This is a Restful service for operating Apache Cassandra. You can find out more about the Management API on [Github](#)". The interface includes a "Apache 2.0" license link and a "default" section. The "default" section lists several endpoints with their methods and descriptions:

- 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)

# menti.com

# 36 63 61



Kubernetes Quiz Time! - 6 questions



Available on the iPhone  
App Store

GET IT ON  
Google play

# Online Workshops



## Section #2 : 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

# Apache Cassandra™ with Kubernetes

1

**Cassandra Why, What, When**

2

**Kubernetes Reminders + Setup**

3

**Cassandra Operator + *Hands-On***

4

**Dashboard UI + *Hands-On***

5

**Grafana Prometheus + *Hands-On***

6

**Resources**

# Online Workshops



## Section #3 : Dashboard

<https://github.com/DataStax-Academy/kubernetes-workshop-online/blob/master/2-dashboard/README.MD>

The screenshot shows a Kubernetes dashboard interface. On the left, a sidebar lists various cluster components: Cluster, Cluster Roles, Namespaces, Nodes, Persistent Volumes, Storage Classes, and several sections under Workloads (Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets). Below these are sections for Discovery and Load Balancing (Ingresses, Services), Config and Storage (Config Maps, Persistent Volume Claims, Secrets), and Custom Resource Definitions. The main area is titled "Overview". It features a "Workloads" section with four green circular icons labeled "Deployments", "Pods", "Replica Sets", and "Stateful Sets". Below this is a "Deployments" table with one entry for "cassandra-operator". The "Pods" section contains three entries for "cluster1-dct1-default-sts-0", "cluster1-dct1-default-sts-1", and "cluster1-dct1-default-sts-2", all of which are running on "kind-cassandra-worker3", "kind-cassandra-worker4", and "kind-cassandra-worker2" respectively, with ages of 32 minutes, 33 minutes, and 14 minutes.

Name	Labels	Pods	Age	Images
cassandra-operator	-	1 / 1	an hour	datastax/cassandra:1.1.0

Name	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Age
cluster1-dct1-default-sts-0	app.kubernetes.io/managed-by: cassandra-operator	kind-cassandra-worker3	Running	0	-	-	32 minutes
cluster1-dct1-default-sts-1	app.kubernetes.io/managed-by: cassandra-operator	kind-cassandra-worker4	Running	0	-	-	33 minutes
cluster1-dct1-default-sts-2	app.kubernetes.io/managed-by: cassandra-operator	kind-cassandra-worker2	Running	0	-	-	14 minutes

# Apache Cassandra™ with Kubernetes

1

**Cassandra Why, What, When**

2

**Kubernetes Reminders + Setup**

3

**Cassandra Operator + *Hands-On***

4

**Dashboard UI + *Hands-On***

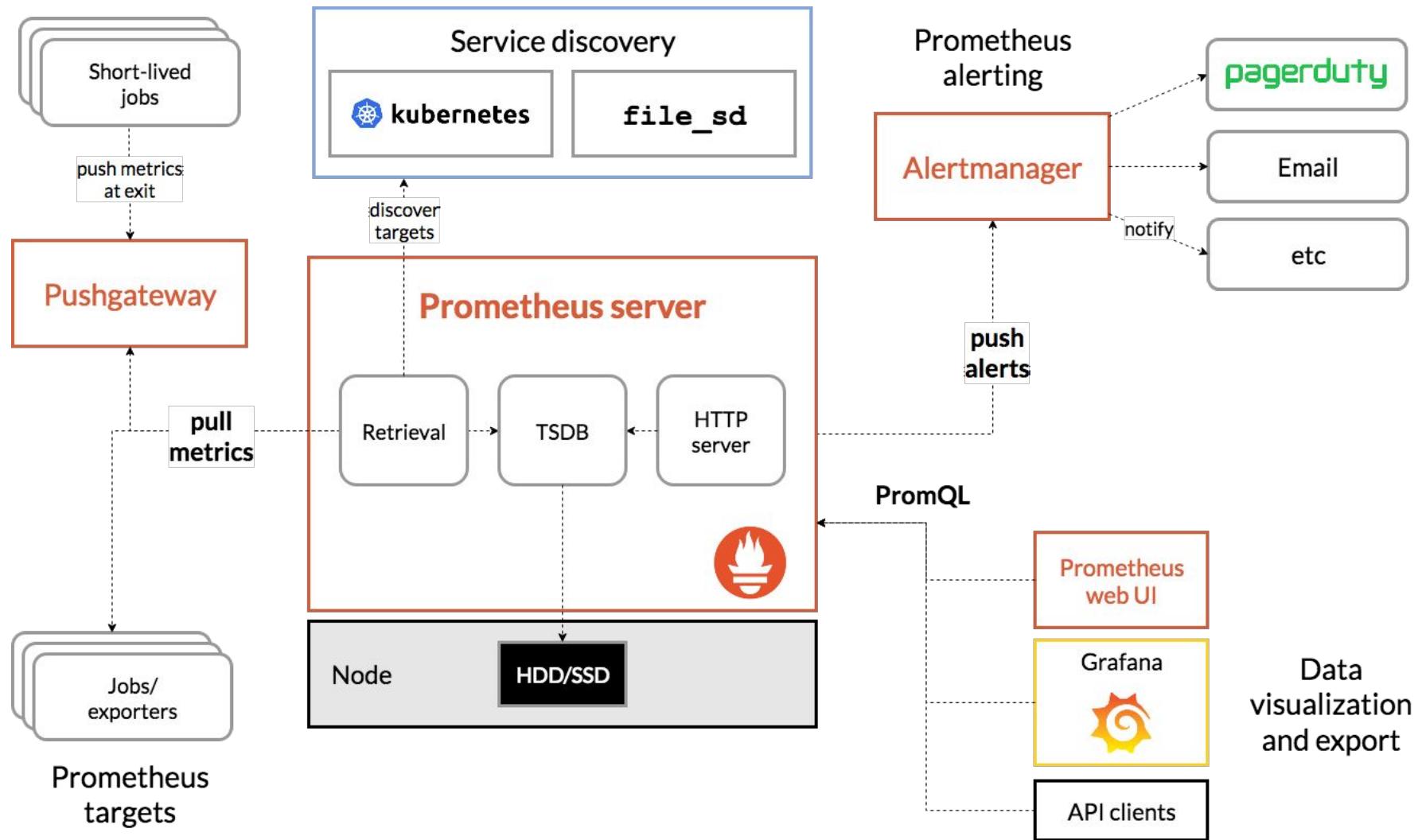
5

**Grafana Prometheus + *Hands-On***

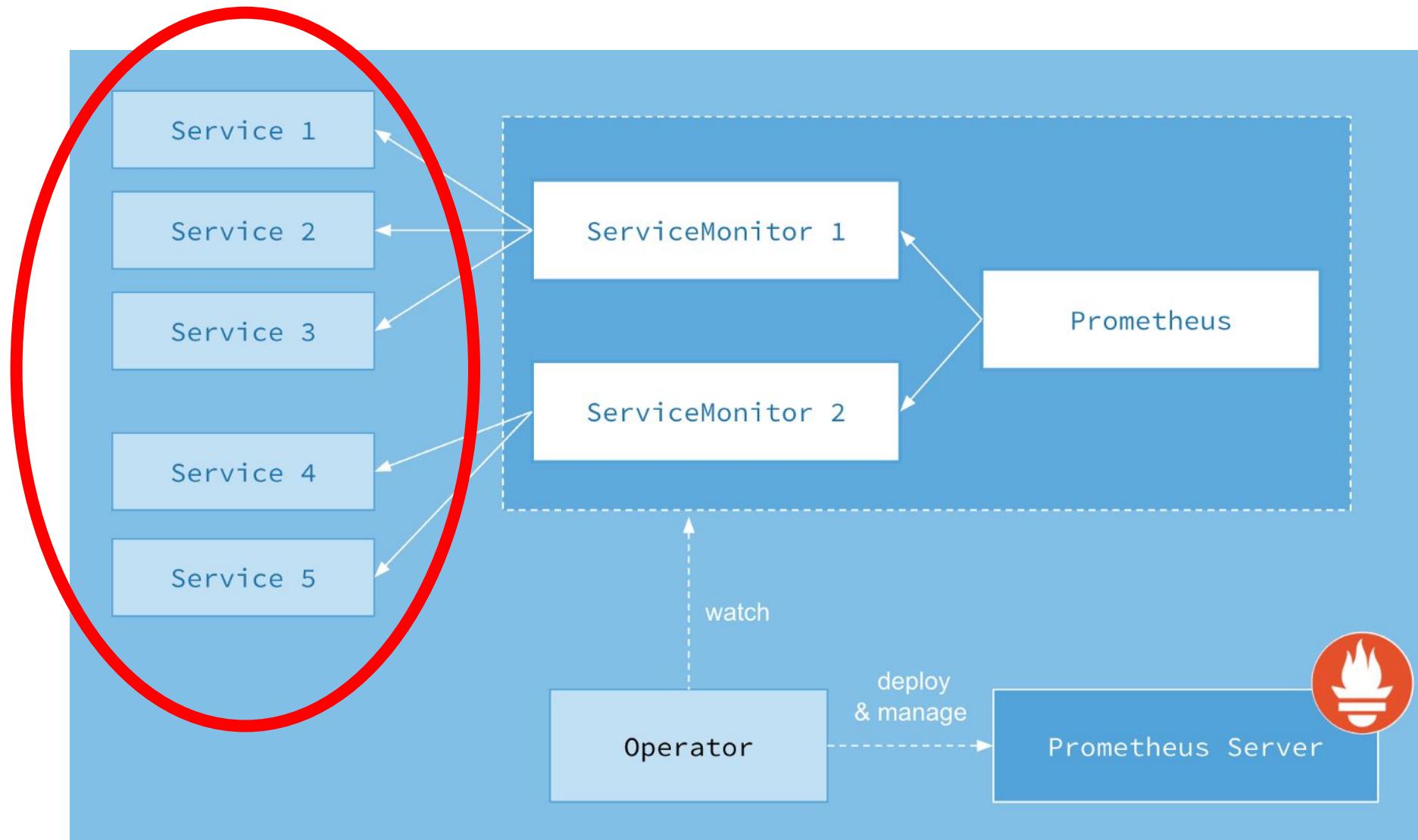
6

**Resources**

# Prometheus + Grafana



# Prometheus metrics collection



# Dse Metrics Exporter

## Reference Documentation

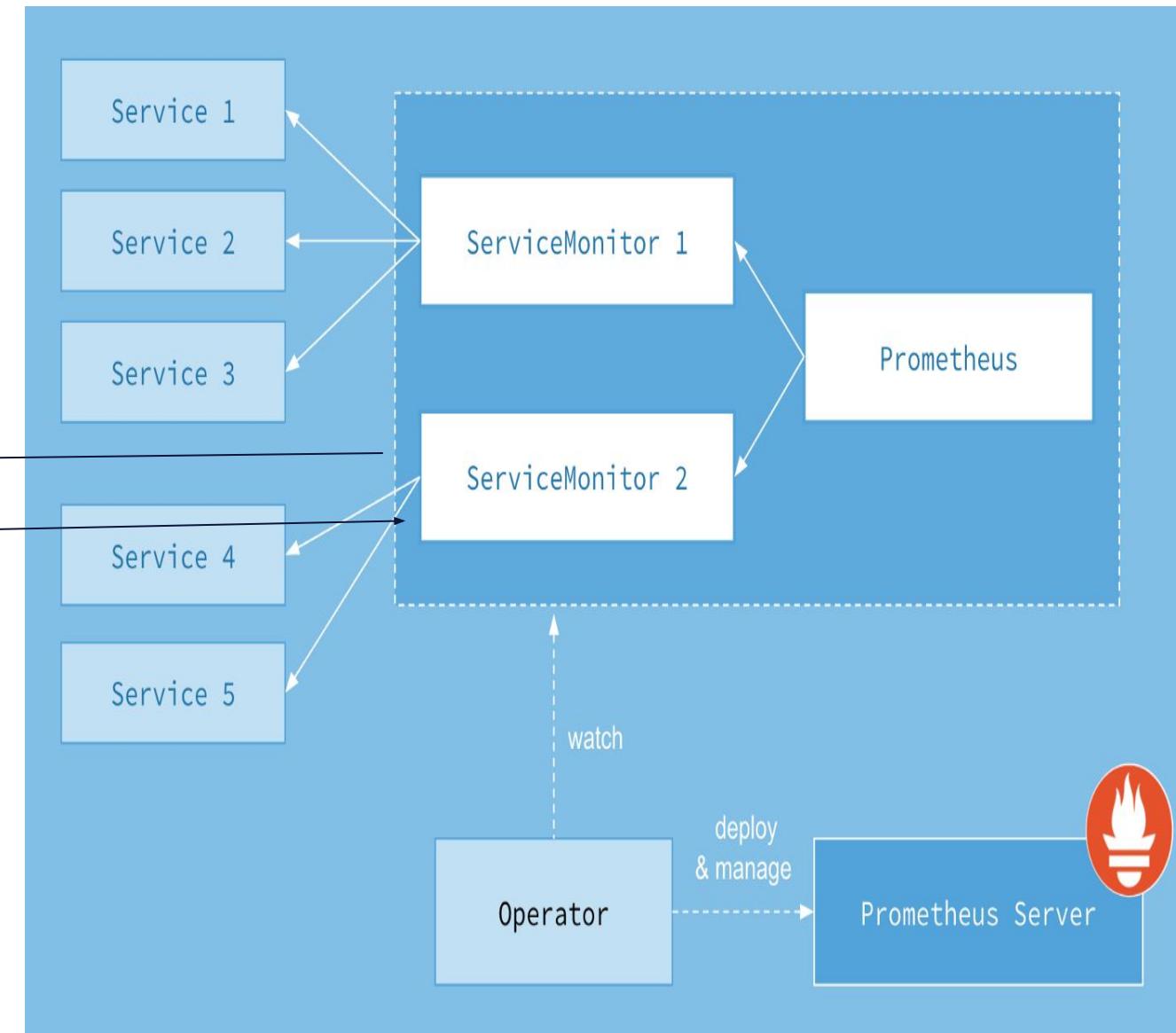
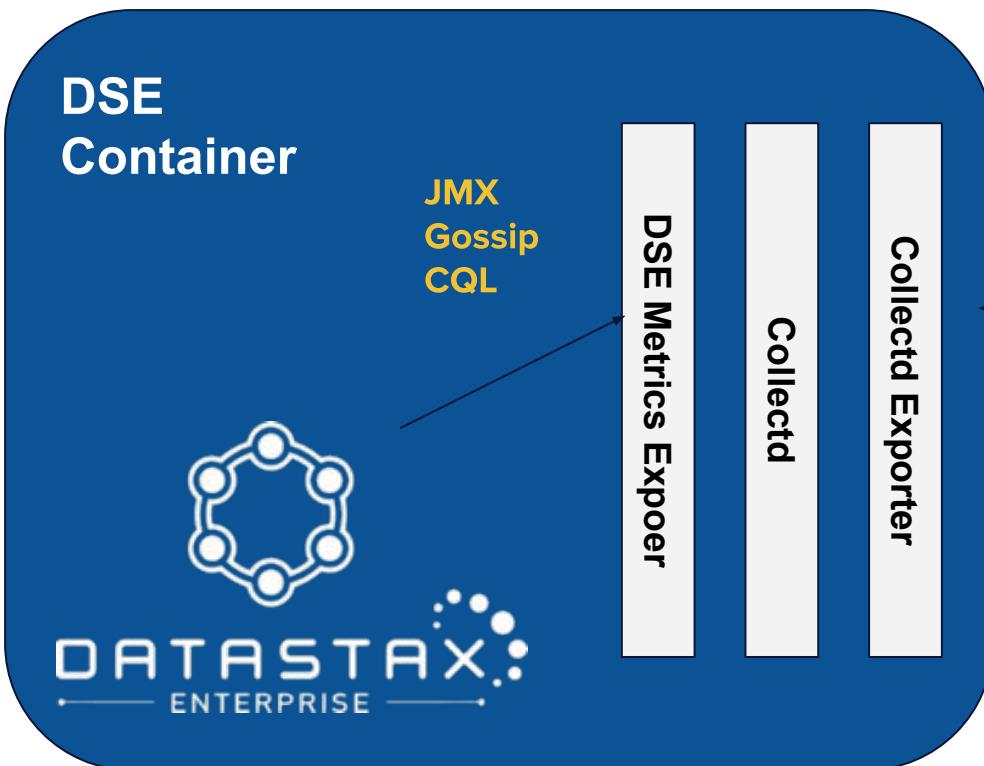
[https://docs.datastax.com/en/dse/6.7/dse-dev/datastax\\_enterprise/tools/metricsCollector/mcIntroduction.html](https://docs.datastax.com/en/dse/6.7/dse-dev/datastax_enterprise/tools/metricsCollector/mcIntroduction.html)

- 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



# Online Workshops



## Section #4 : Grafana Prometheus

[https://github.com/DataStax-Academy/kubernetes-workshop-online/blob/master/3-prometheus\\_grafana/README.MD](https://github.com/DataStax-Academy/kubernetes-workshop-online/blob/master/3-prometheus_grafana/README.MD)



# Apache Cassandra™ with Kubernetes

1

**Cassandra Why, What, When**

2

**Kubernetes Reminders + Setup**

3

**Cassandra Operator + *Hands-On***

4

**Dashboard UI + *Hands-On***

5

**Grafana Prometheus + *Hands-On***

6

**Resources**

# Developer Resources

**LEARN**

Join [academy.datastax.com](https://academy.datastax.com)

<https://academy.datastax.com/resources/cassandra-developer-workshop>

Free online courses - Cassandra certifications

**ASK/SHARE**

Join [community.datastax.com](https://community.datastax.com)

Ask/answer community user questions - share your expertise

**CONNECT**

Follow us @DataStaxDevs

We are on Twitter - Twitch!

**REVIEW**

Slides and code for this course are available at

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

[www.datastax.com/keepcalm](http://www.datastax.com/keepcalm)

SRE office hours for Apache Cassandra

Cassandra cluster health checks

@ no charge

Send an email to [keepcalm@dastastax.com](mailto:keepcalm@dastastax.com)



# Training Courses at DataStax Academy

- Free self-paced DSE 6 courses
  - [DS201: DataStax Enterprise 6 Foundations of Apache Cassandra™](#)
  - [DS210: DataStax Enterprise 6 Operations with Apache Cassandra™](#)
  - [DS220: DataStax Enterprise 6 Practical Application Data Modeling with Apache Cassandra™](#)
  - [DS330: DataStax Enterprise 6 Graph](#)
  - [DS332: DataStax Enterprise 6 Graph Analytics \(NEW\)](#)



# Cassandra The Definitive Guide

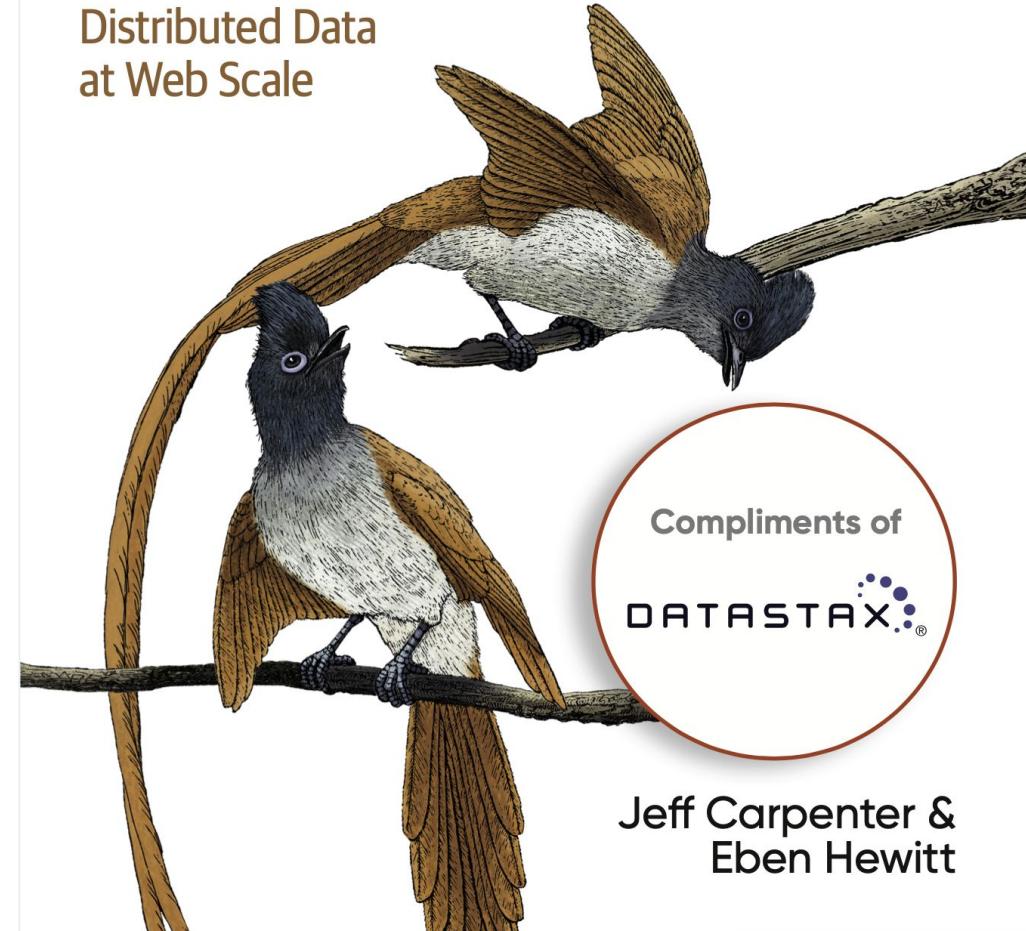
<https://www.datastax.com/resources/ebook/oreilly-cassandra-definitive-guide>

O'REILLY®

Third  
Edition

# Cassandra The Definitive Guide

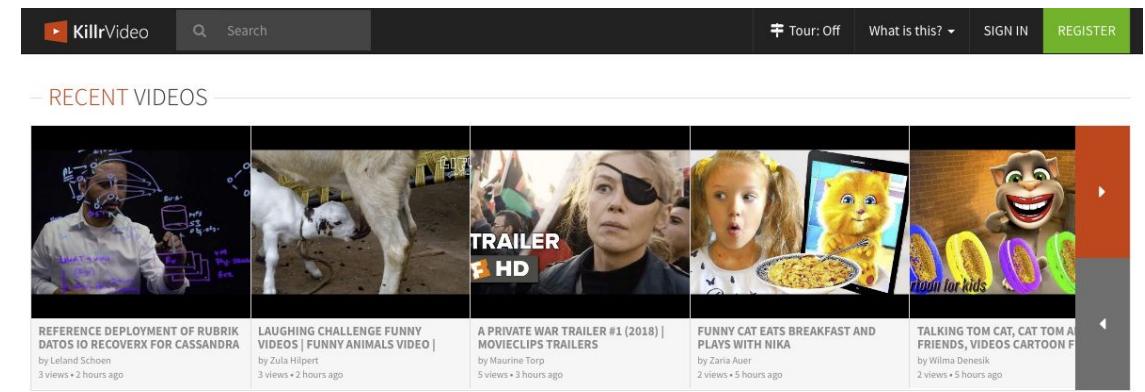
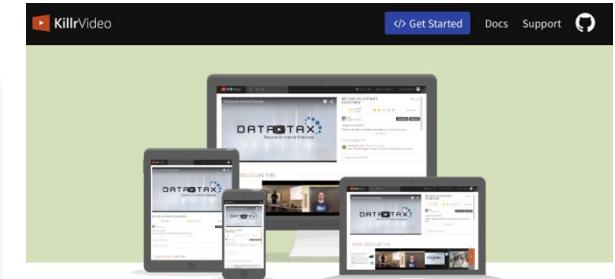
Distributed Data  
at Web Scale



Jeff Carpenter &  
Eben Hewitt

# KillrVideo Reference Application

- Reference application for learning how to use Apache Cassandra and DataStax Enterprise
  - DataStax Drivers
  - Docker images
- Source code freely available
  - <https://github.com/killrvideo>
- Live version
  - <http://killrvideo.com>
- Download, test, modify, contribute!



# menti.com

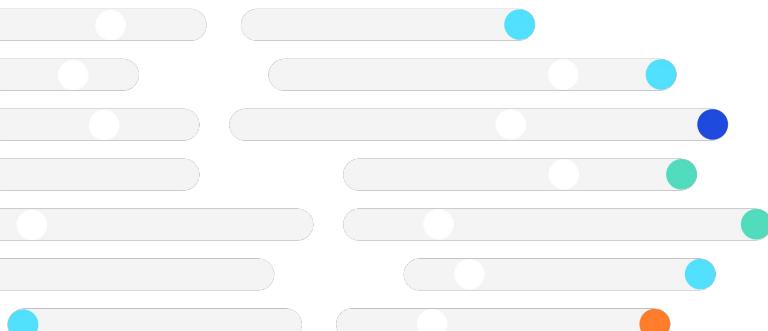
# 36 63 61

TAKE THE SURVEY



Available on the iPhone  
**App Store**

GET IT ON  
**Google play**



**Thank you**

**SURVEY**