



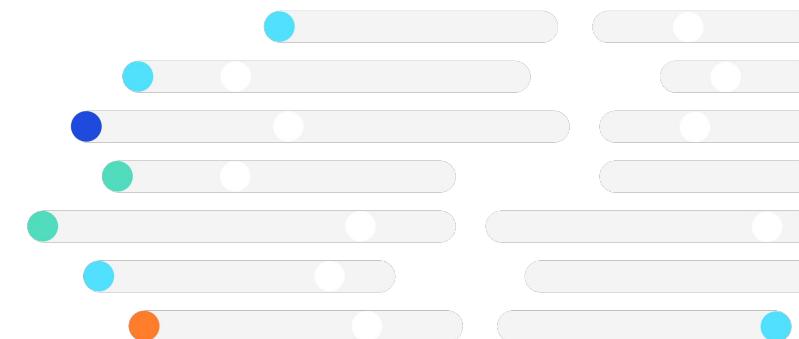
kubernetes



CLOUD NATIVE
COMPUTING FOUNDATION

KubeCon 2020 Workshop

**Putting Cassandra on Automatic
with Kubernetes**



Apache Cassandra™ with Kubernetes

- 1 **Housekeeping and Quiz**
- 2 **Cassandra Basics**
- 3 **Setting Up and Monitoring Cassandra**
- 4 **Working with Data**
- 5 **Scaling Up and Down**
- 6 **Running Repairs**
- 7 **Backing Up and Restoring Data**
- 8 **Resources**



Disclaimer

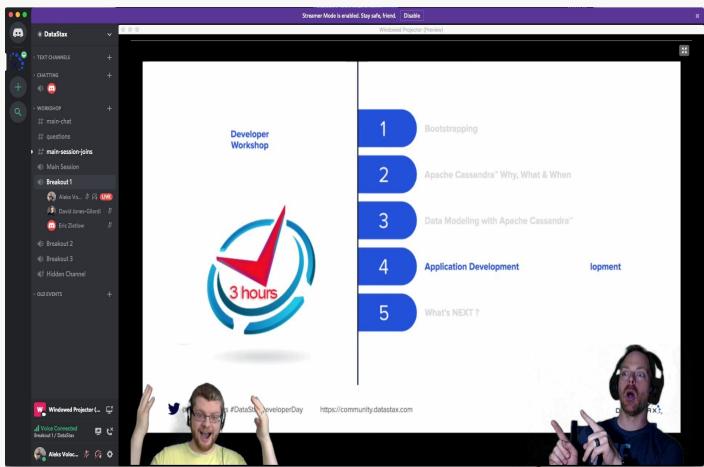
Today,

we won't introduce the basics of Docker and Kubernetes

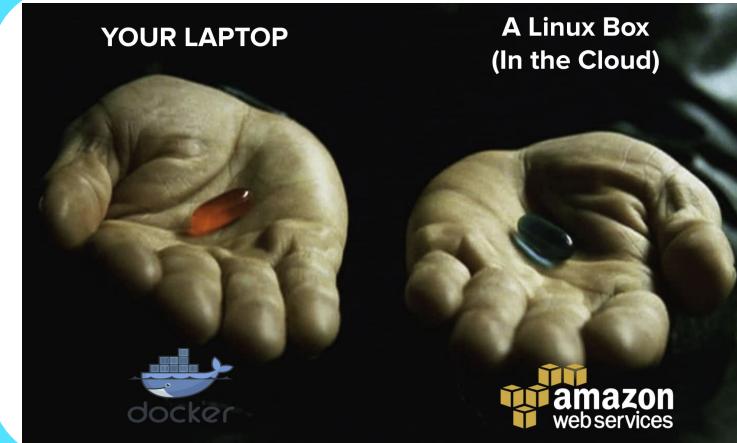
It's a KubeCon after all!



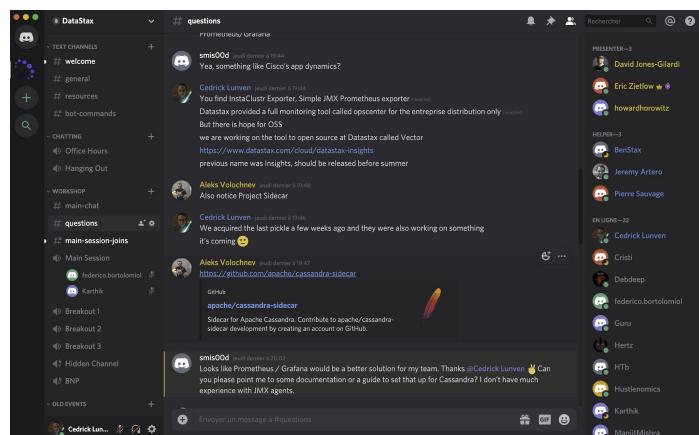
STREAMS



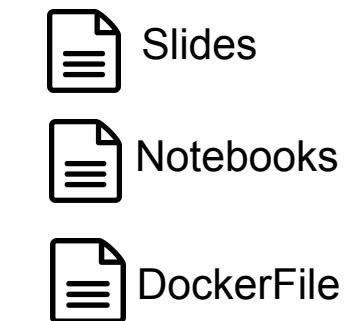
RUNTIME



QUESTIONS



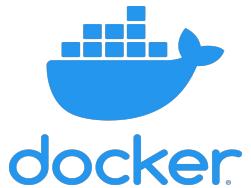
MATERIALS





YOUR LAPTOP

Your Laptop



docker



kind



kubernetes



CLOUD INSTANCE



<http://kubecon2020.datastaxtraining.com/>

Cloud Env



kubernetes



menti.com



Available on the iPhone
App Store

GET IT ON
Google play

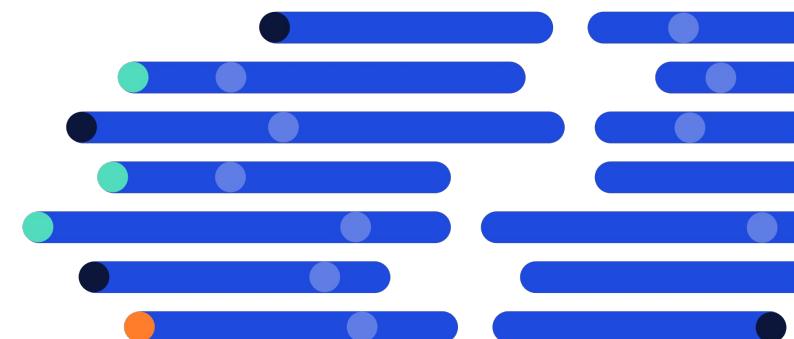
Online Workshops



Section #0 : Requesting Cloud Instances

<https://kubecon2020.datastaxtraining.com/>

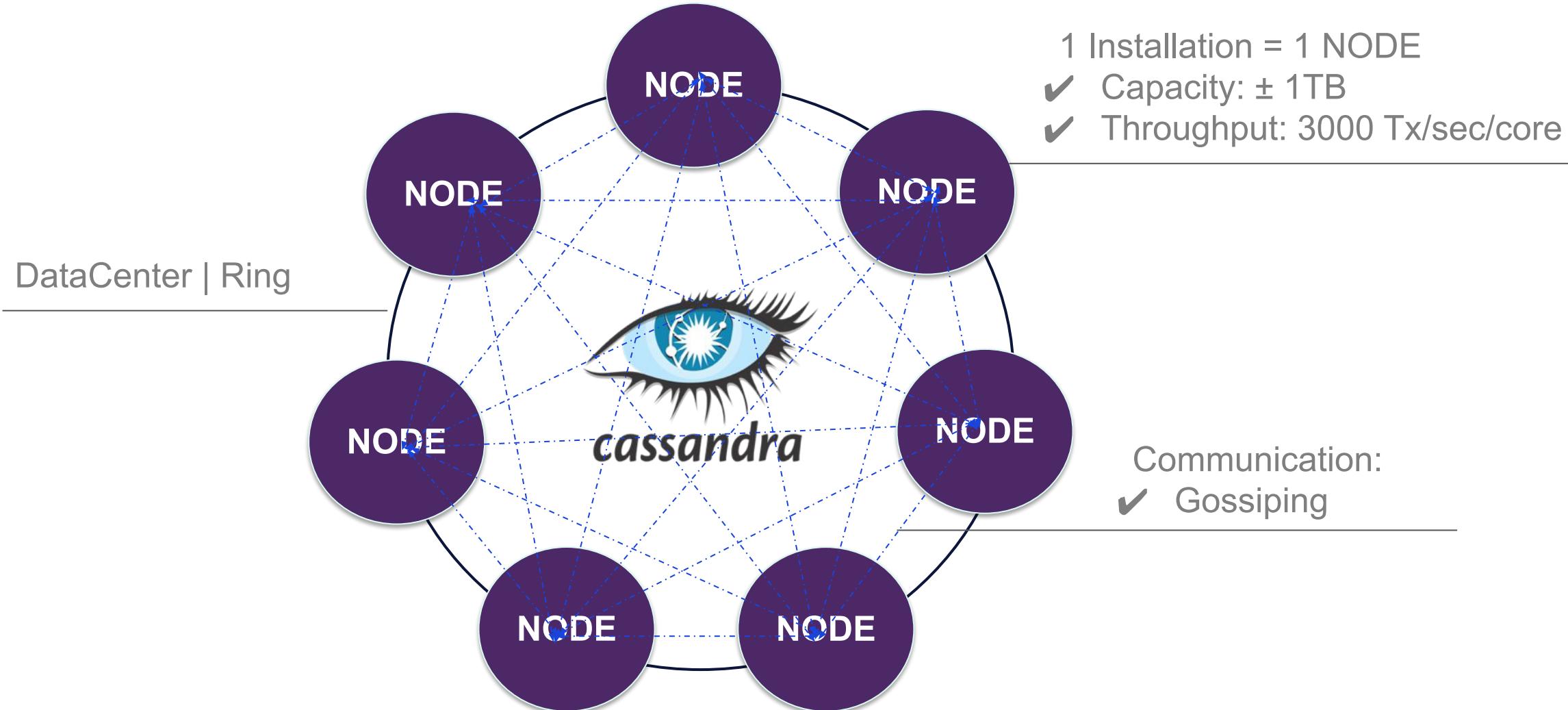
- Get Our Cloud Instance



Apache Cassandra™ with Kubernetes

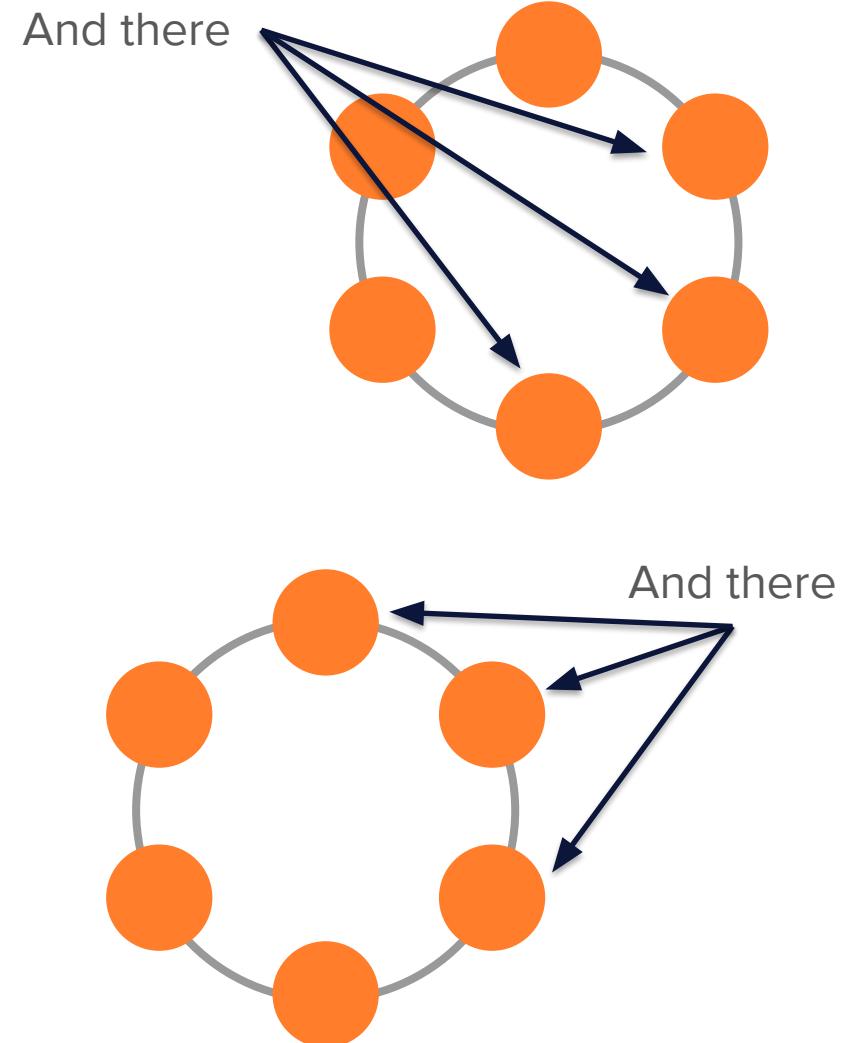
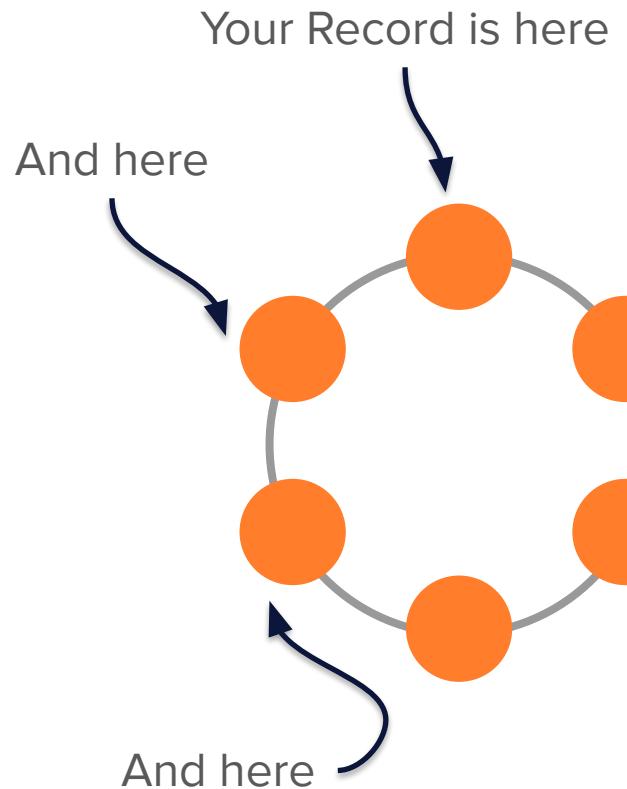
- 1 Housekeeping and Quiz
- 2 Cassandra Basics
- 3 Setting Up and Monitoring Cassandra
- 4 Working with Data
- 5 Scaling Up and Down
- 6 Running Repairs
- 7 Backing Up and Restoring Data
- 8 Resources

Apache Cassandra™ = NoSQL Distributed Database



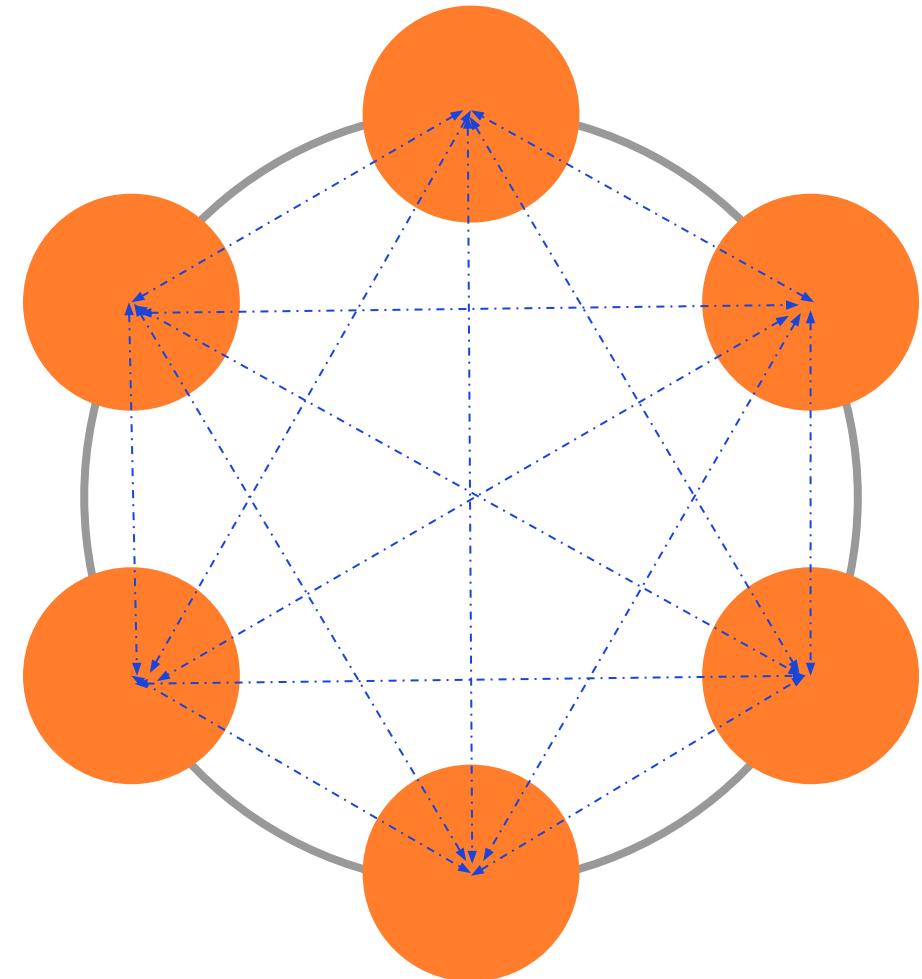
Cassandra Features

- Distributed
 - Responsive
 - Scalable
- Replicated
 - Available
- No Primary/Secondary Nodes
 - No SPoF



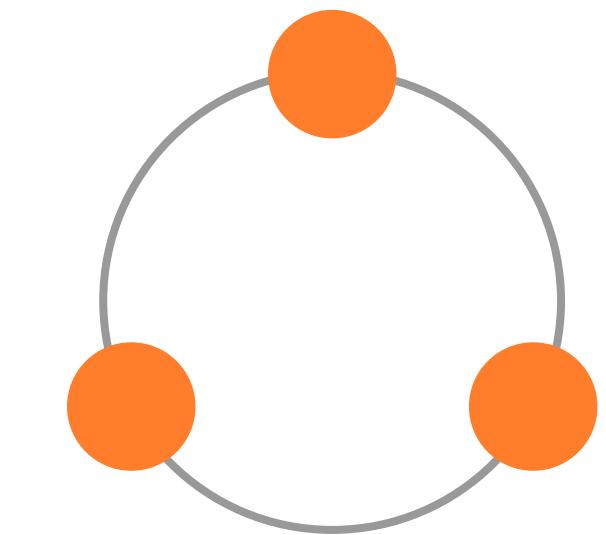
Cassandra is Distributed and Highly Available

- Nodes organized in a group
- Nodes monitor each other's health status using a gossip protocol
- Cassandra routes requests away from nodes that are down or slow

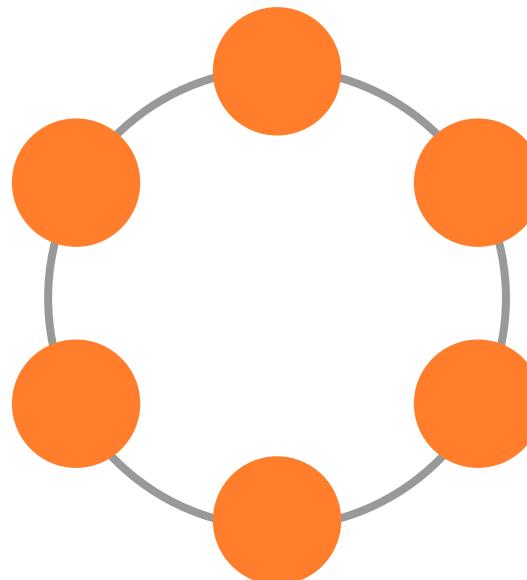


Horizontal vs. Vertical Scaling

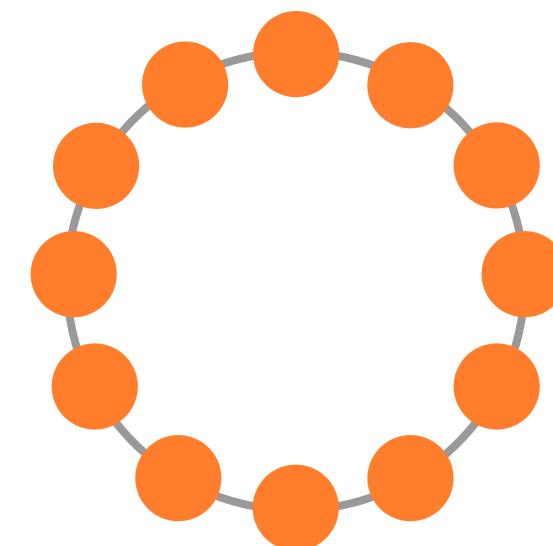
- Vertical scaling requires one large expensive machine
- Horizontal scaling requires multiple less-expensive commodity hardware
- Cassandra scales horizontally *and* linearly!



100,000 transactions/second



200,000 transactions/second



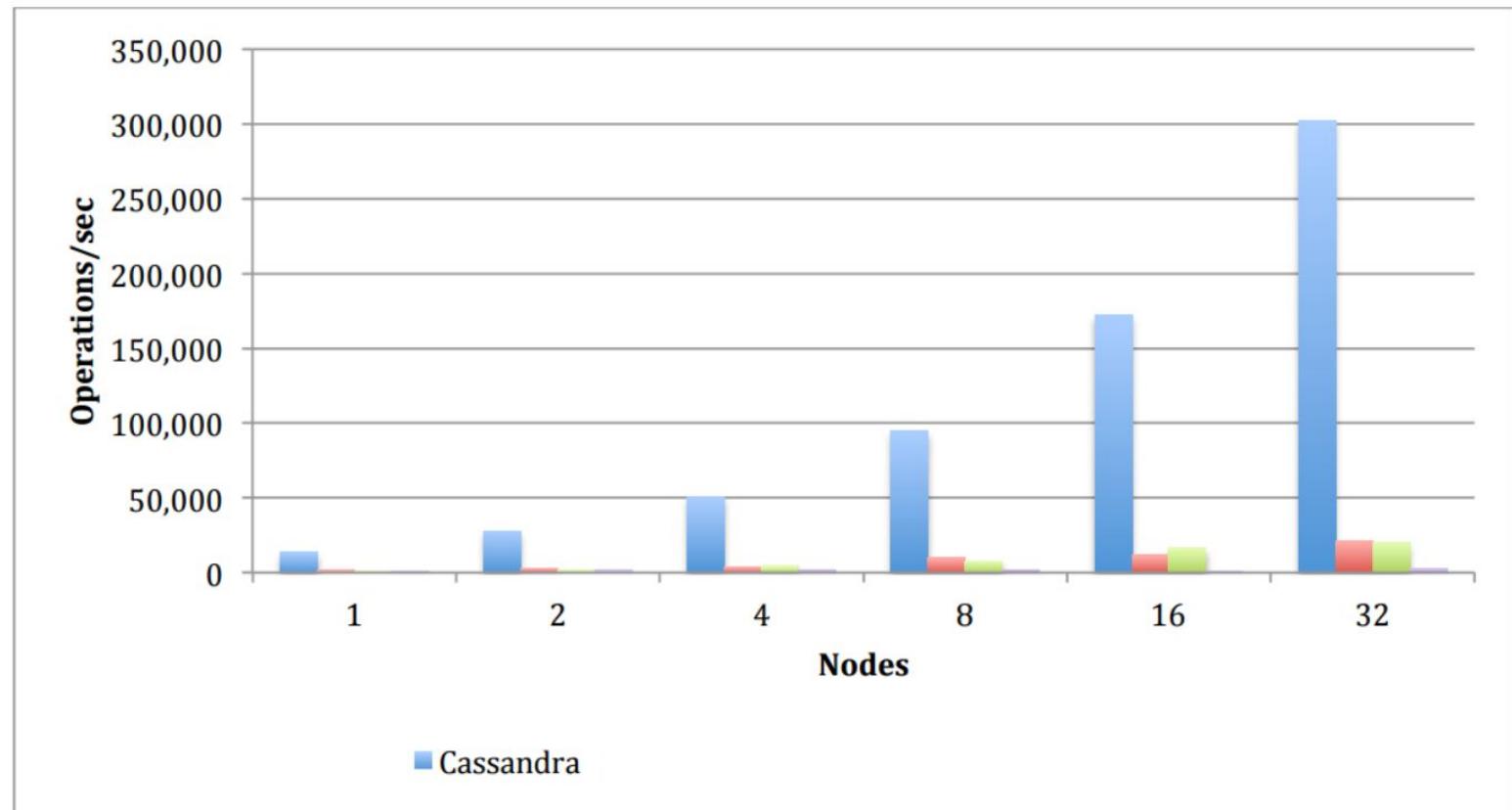
400,000 transactions/second



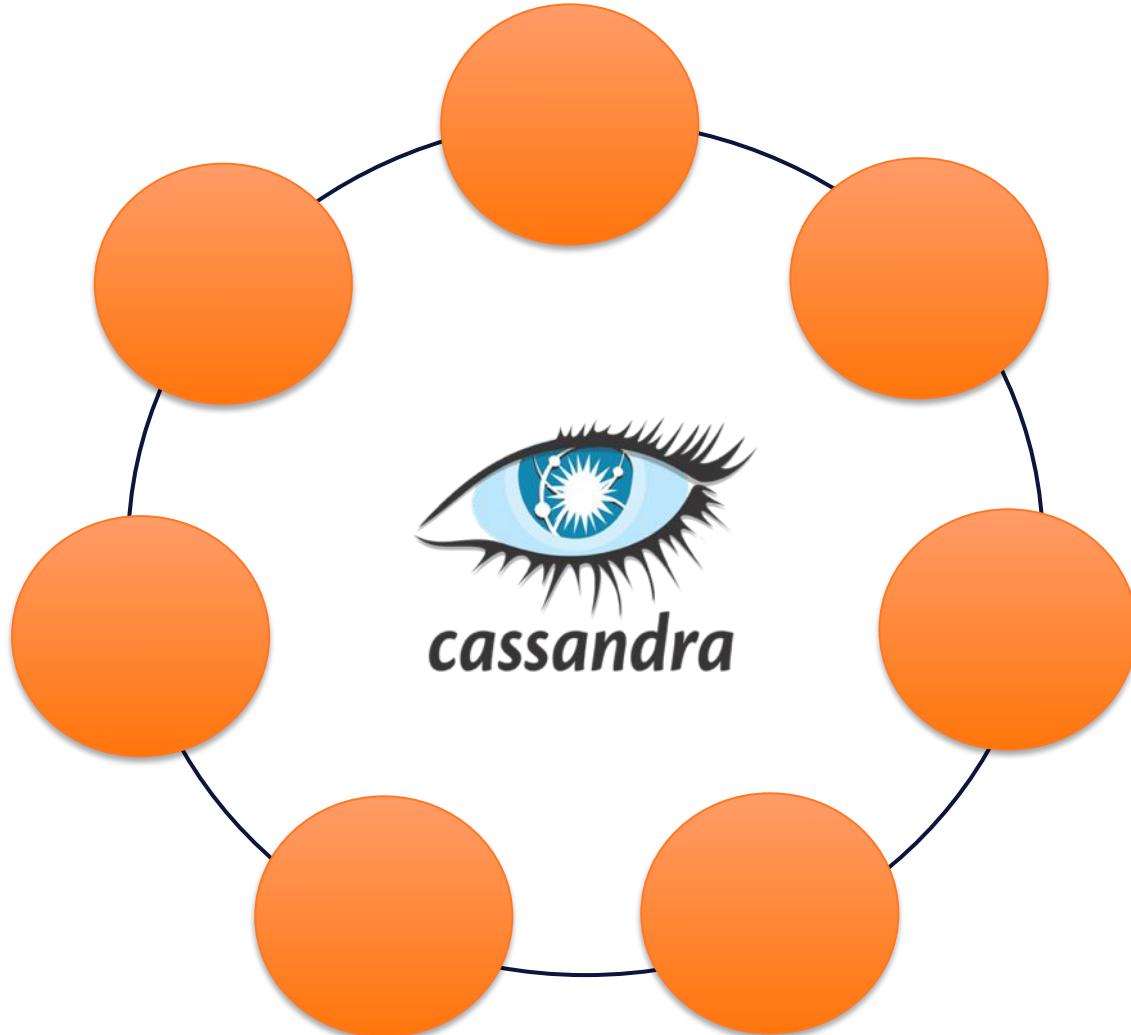
Scales Linearly

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

Balanced Read/Write Mix



Data is Distributed



City	Name	Phone
Moscow	Ivan	8.000.000
Moscow	Anna	4.000.000
SPb	Aleksandr	2.230.000
Berlin	Felix	3.350.000
London	John	9.200.000
Sydney	Abigail	4.900.000
Berlin	Sarah	500.000
Toronto	Andrew	6.200.000
Toronto	Peter	4.200.000
Paris	Cedrick	1.100.000
Paris	Evelyn	37.430.000
Mumbai	Prabhakaran	20.200.000

Partition Key

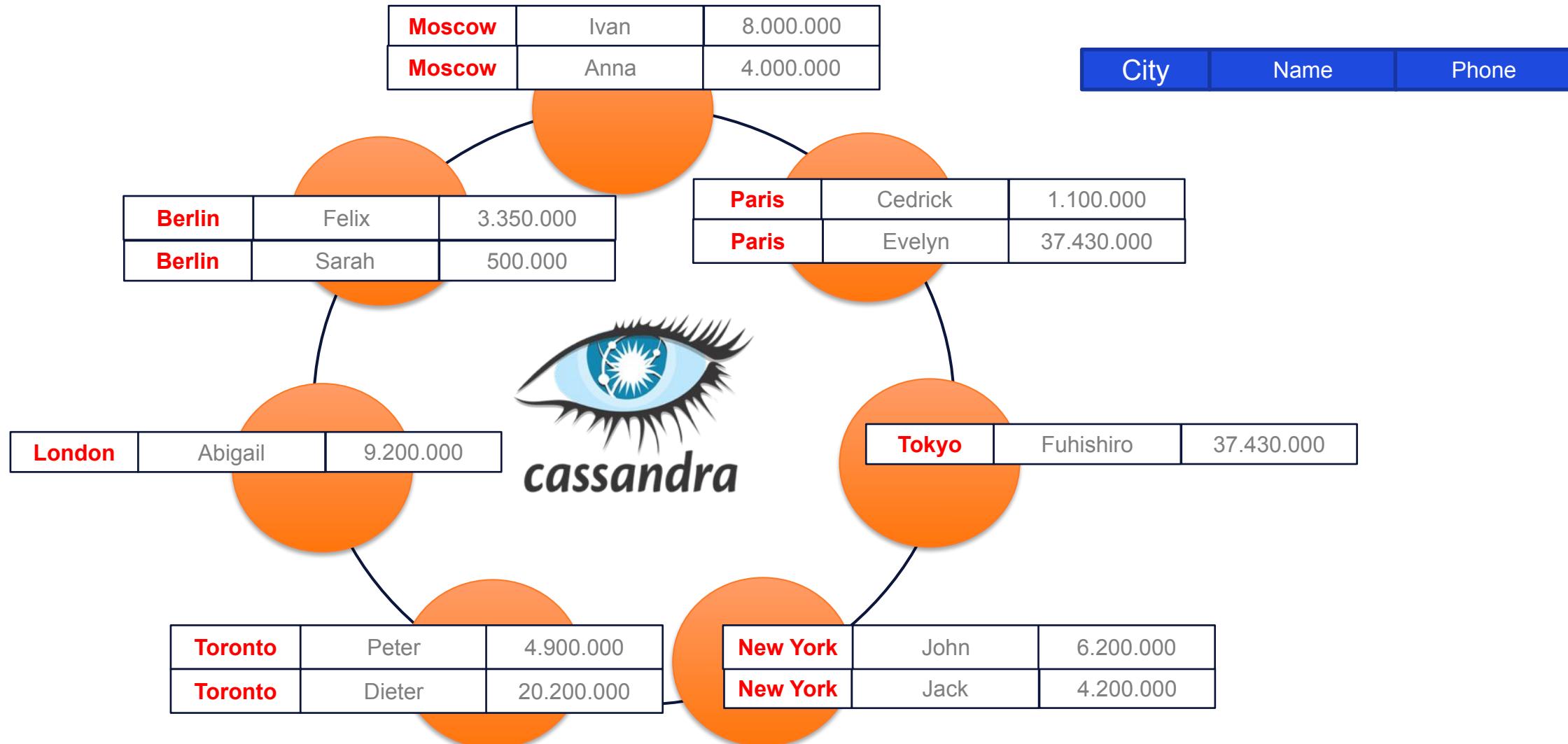


@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



Data is Distributed

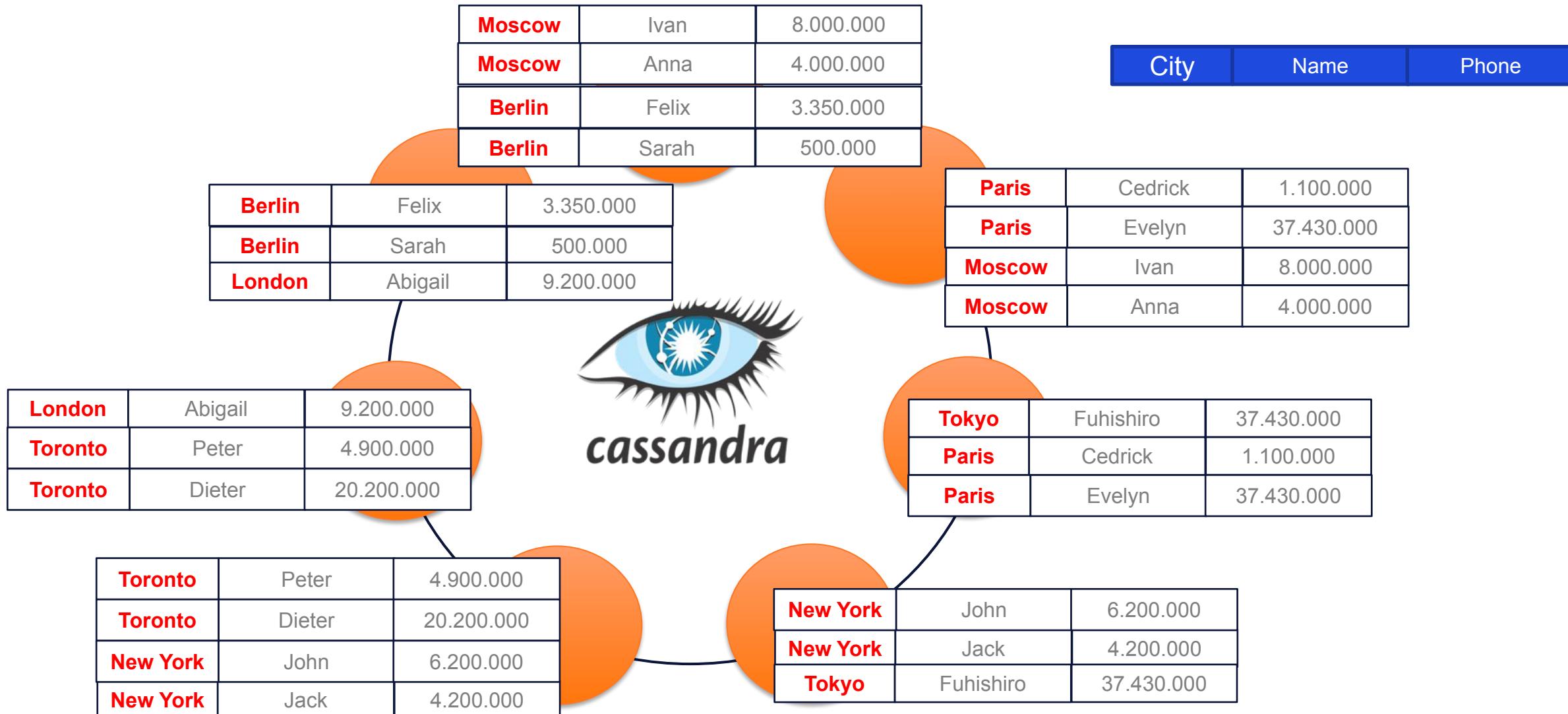


@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>

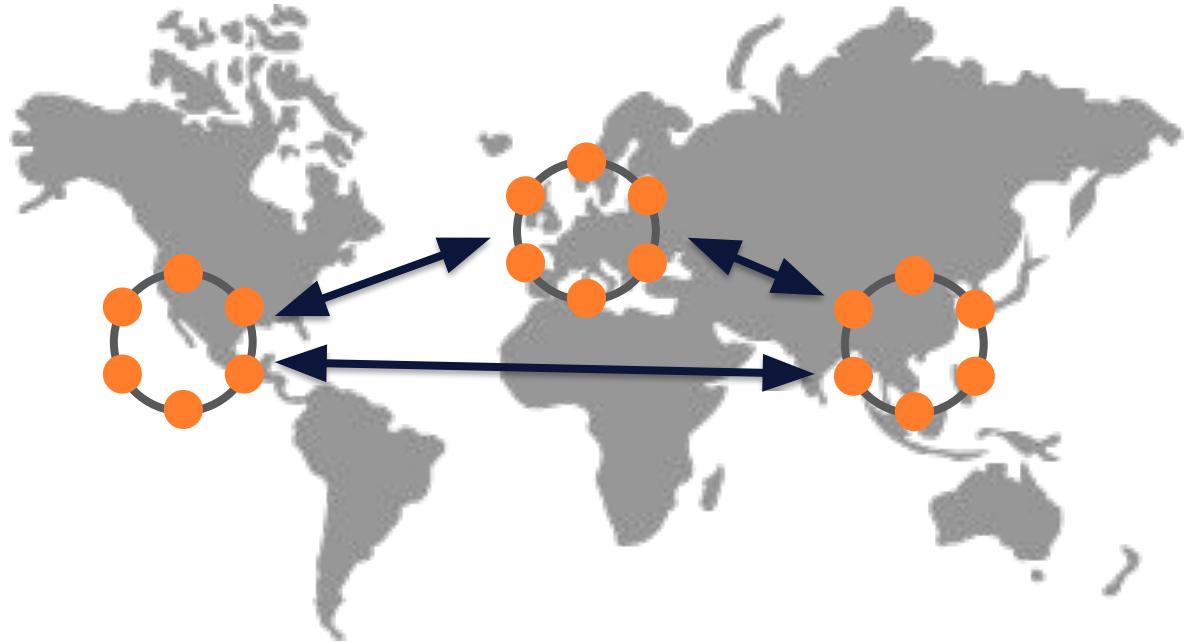


Data is Replicated

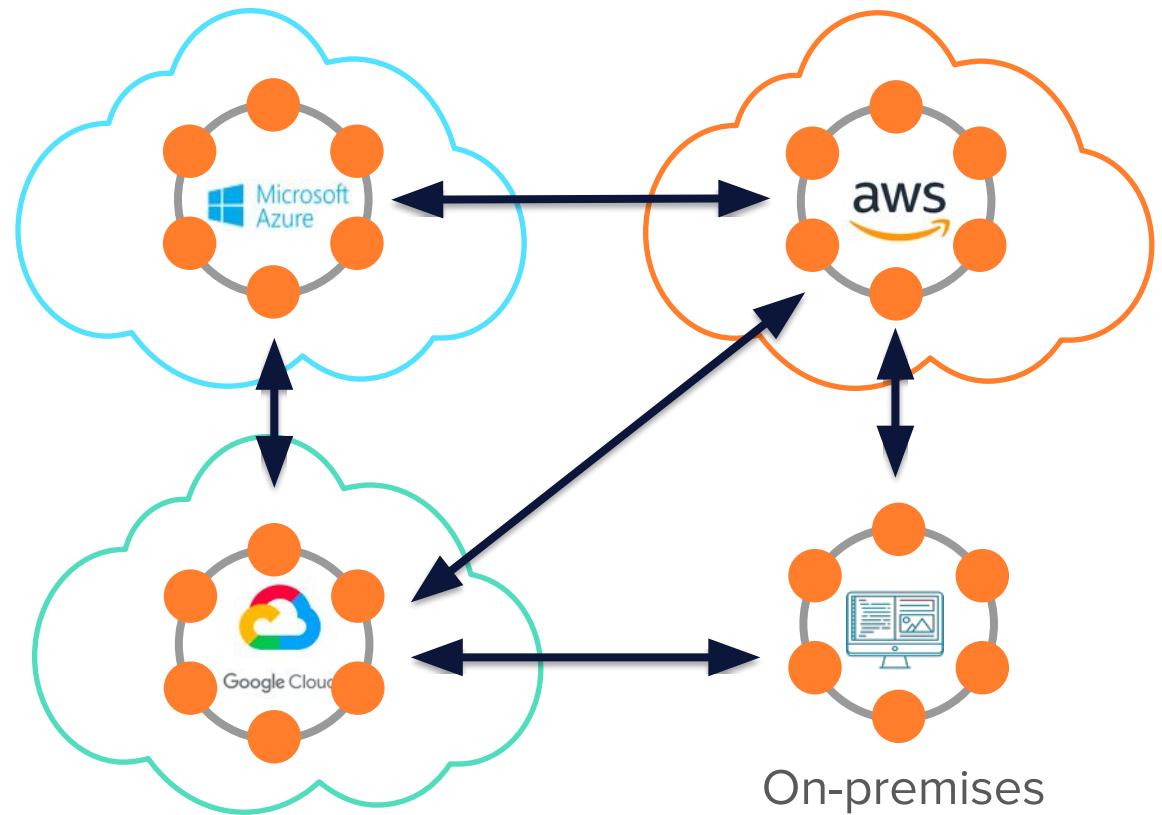


Data Distributed Everywhere

- Geographic Distribution



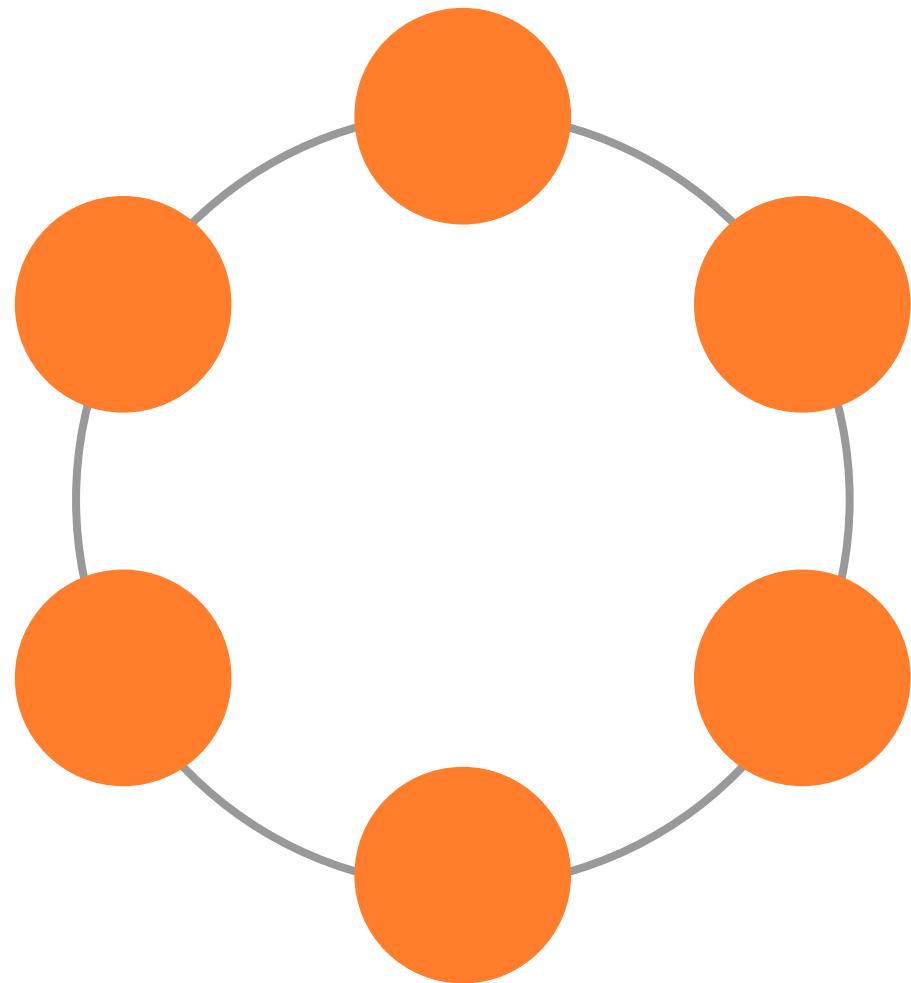
- Hybrid-Cloud and Multi-Cloud



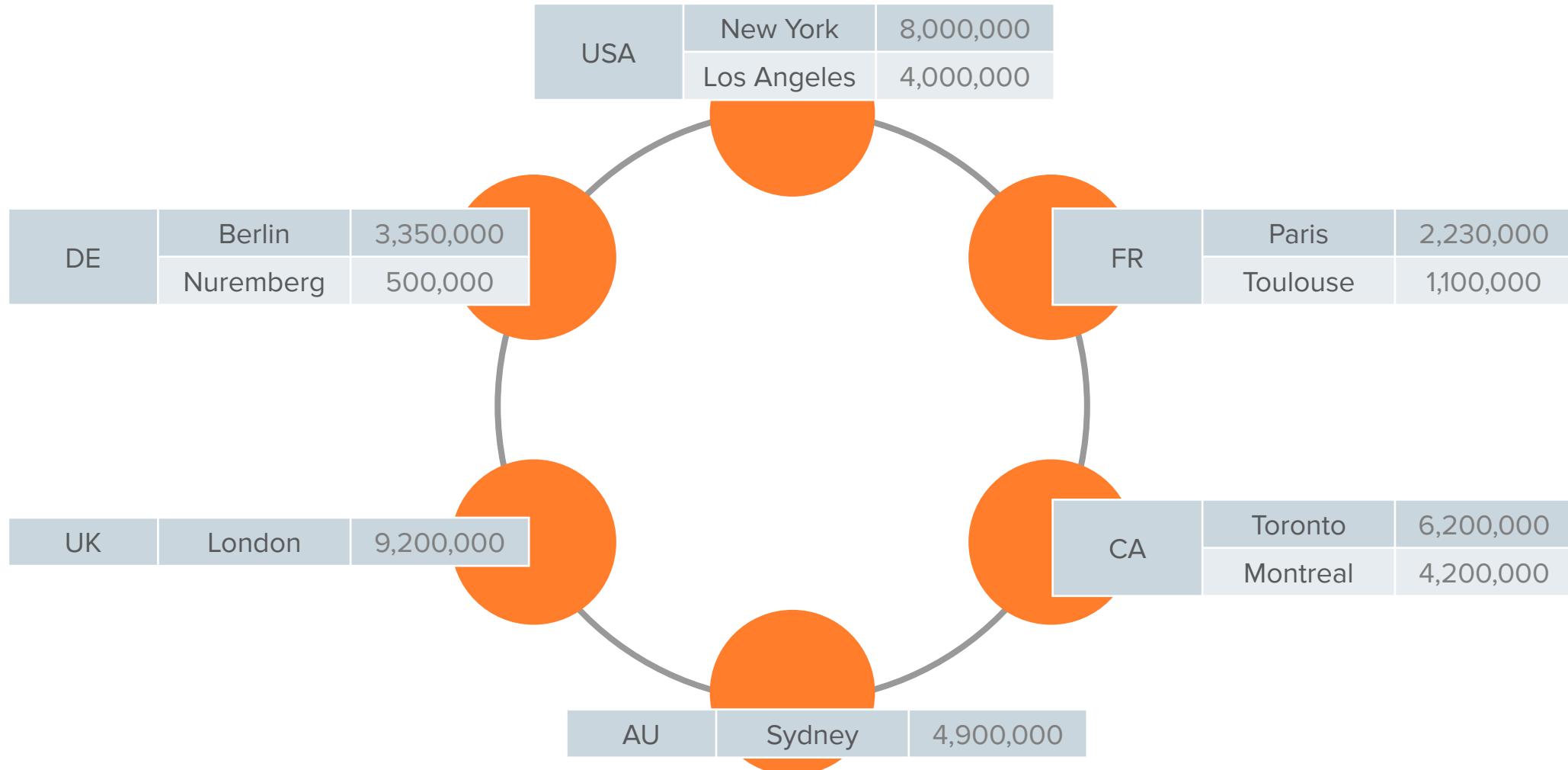
How the Ring Works

- How is the data distributed?

Country	City	Population
USA	New York	8,000,000
	Los Angeles	4,000,000
DE	Berlin	3,350,000
	Nuremberg	500,000
FR	Paris	2,230,000
	Toulouse	1,100,000
CA	Toronto	6,200,000
	Montreal	4,200,000
UK	London	9,200,000
AU	Sydney	4,900,000



How the Ring Works

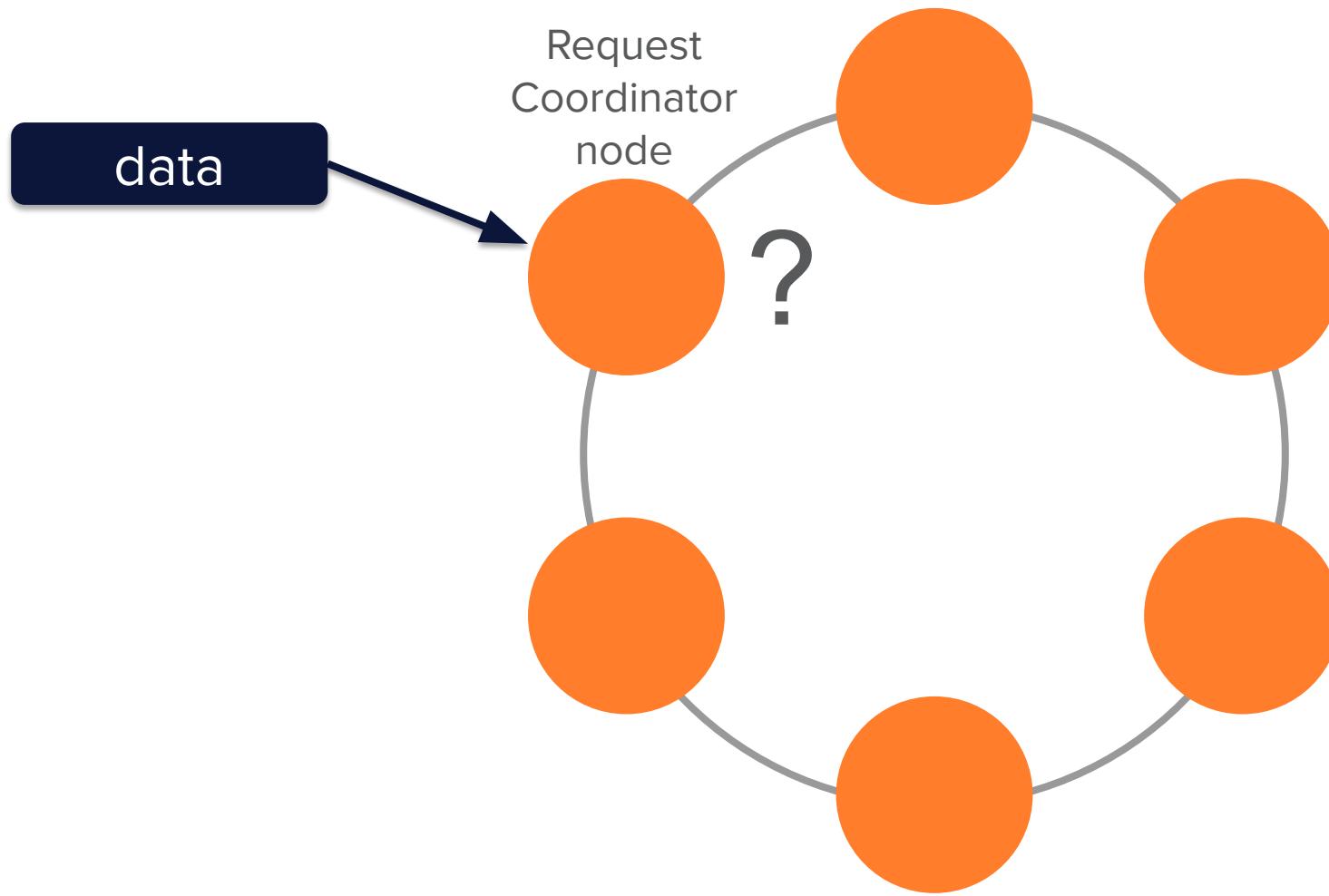


@DataStaxDevs #DataStaxDeveloperDay

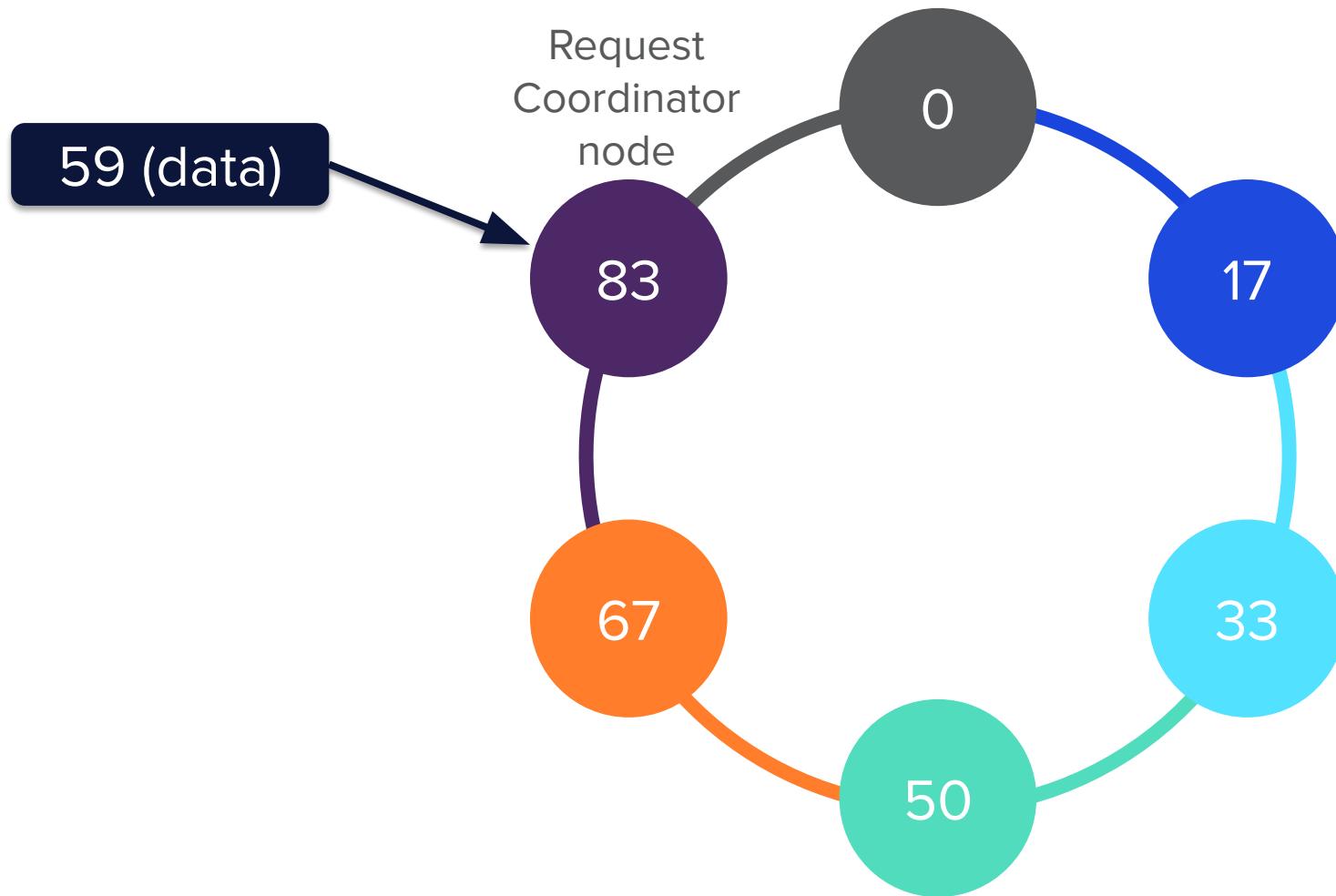
<https://community.datastax.com>



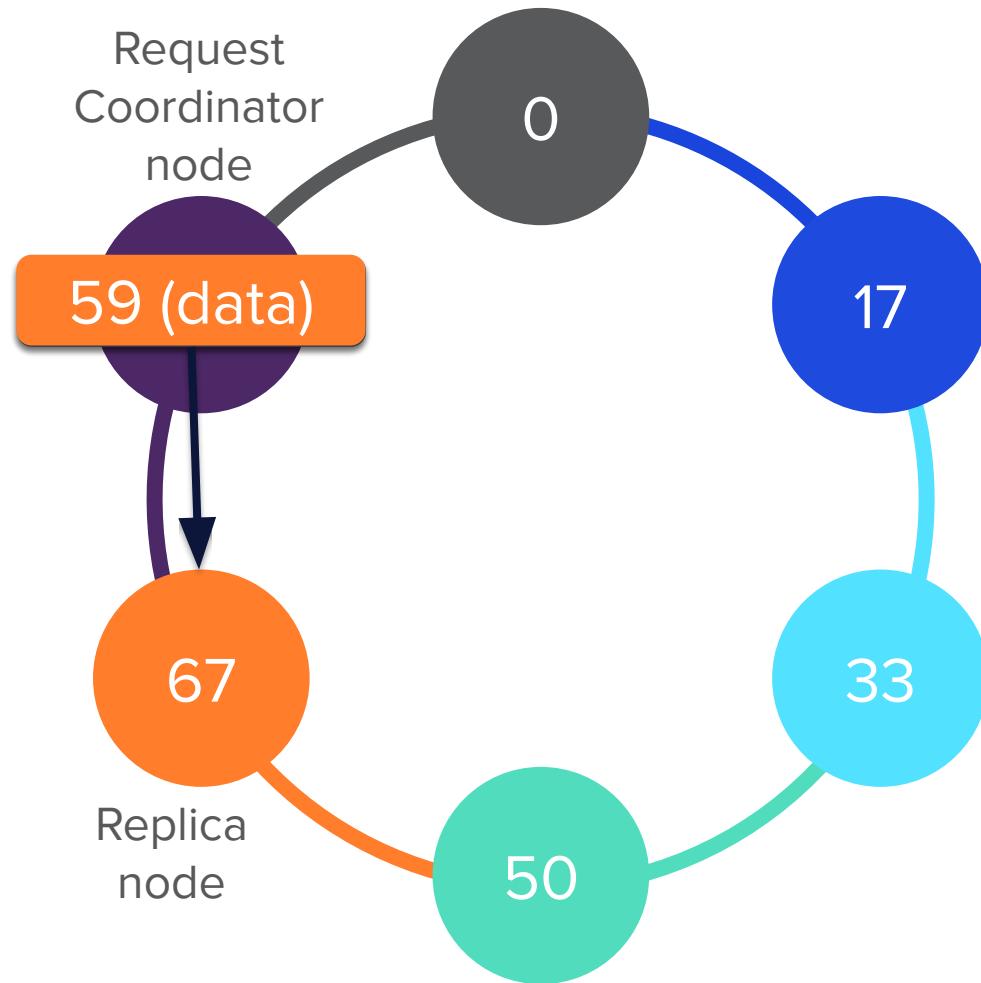
How the Ring Works



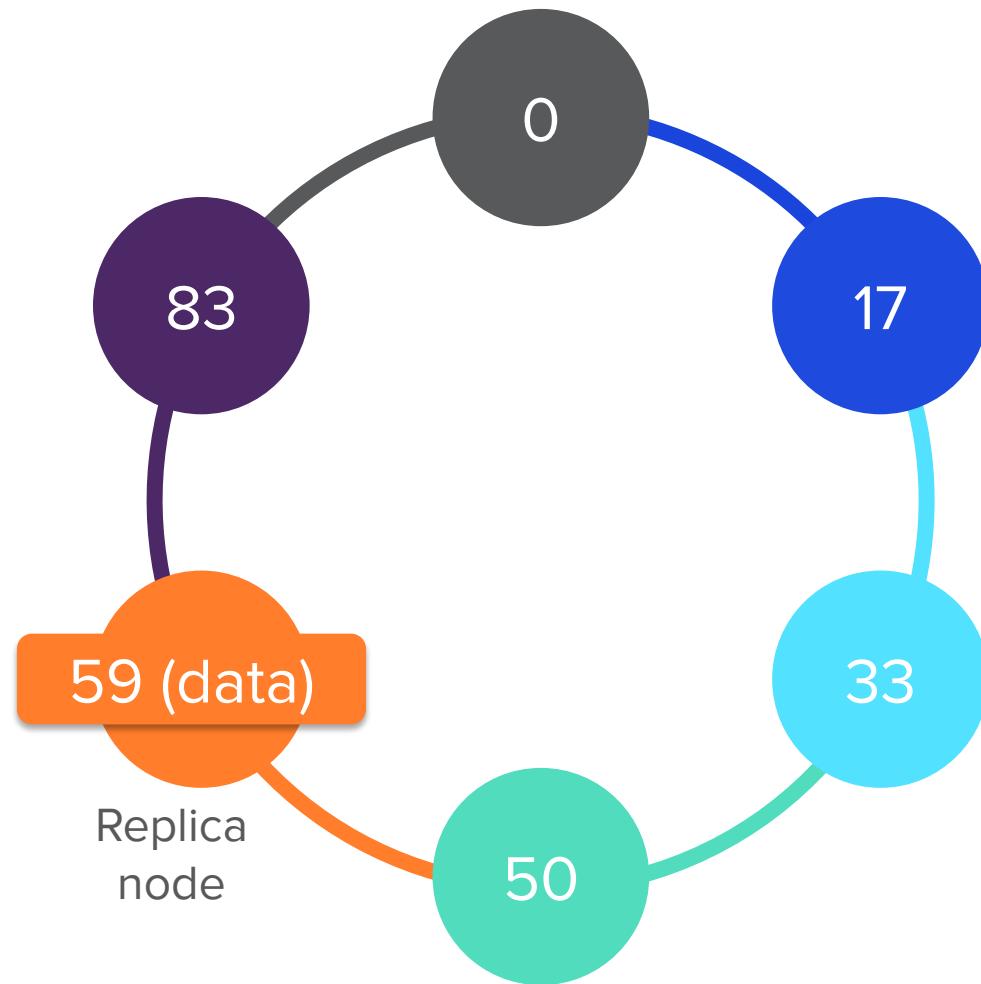
How the Ring Works



How the Ring Works



How the Ring Works



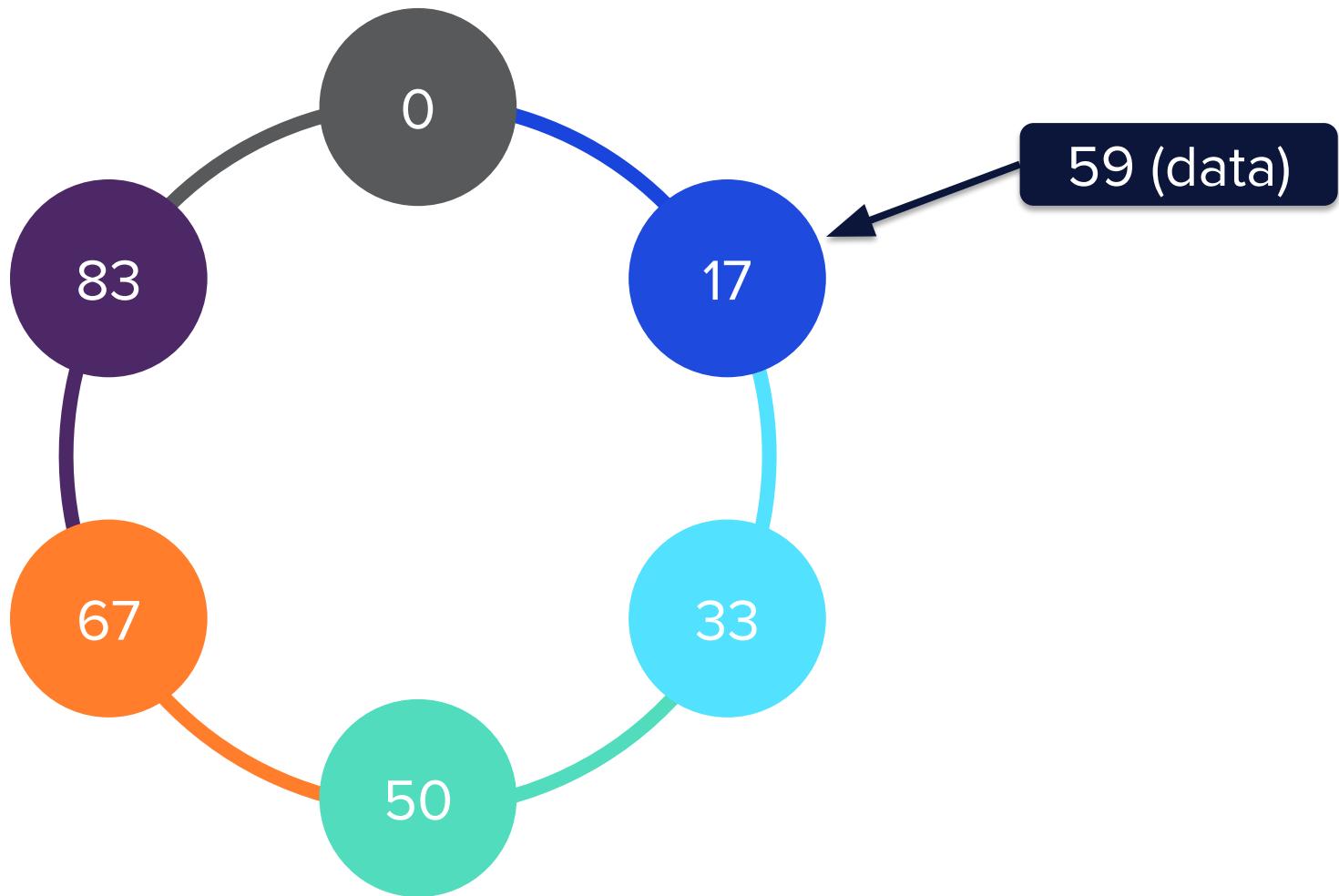
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



Replication within the Ring

RF = 1



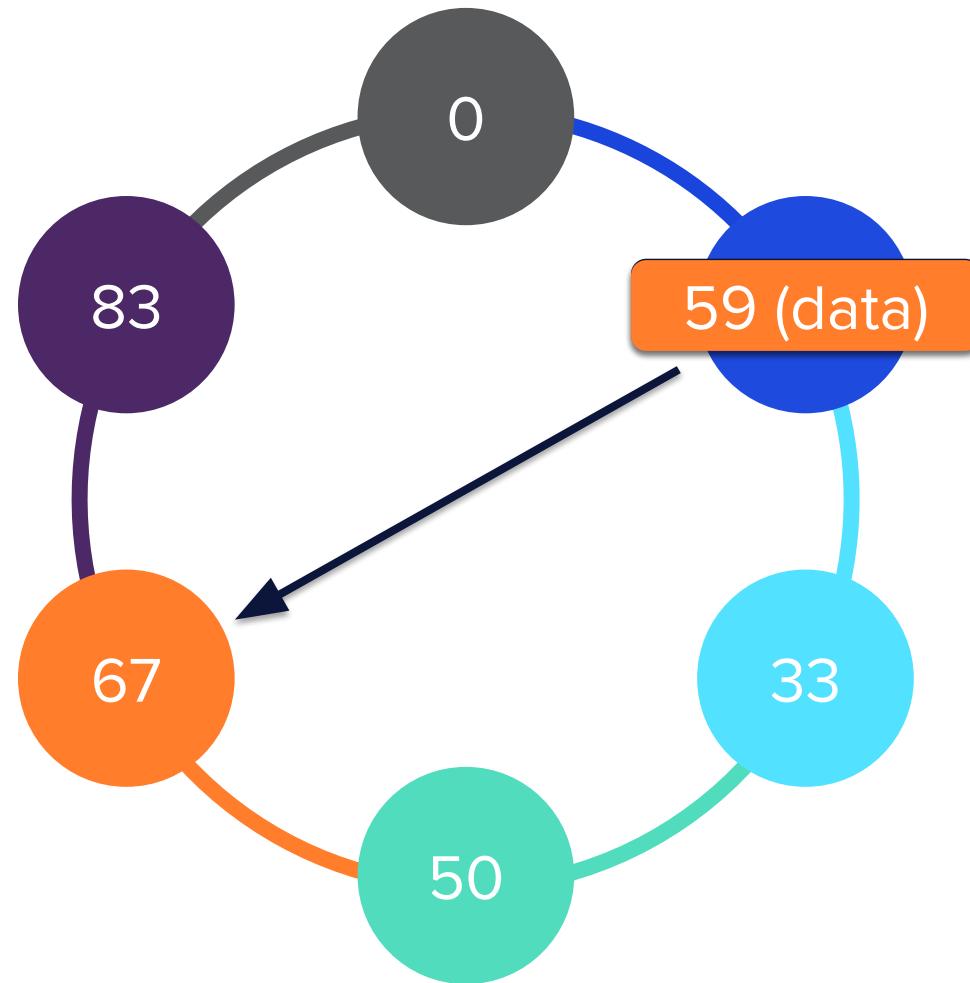
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



Replication within the Ring

RF = 1



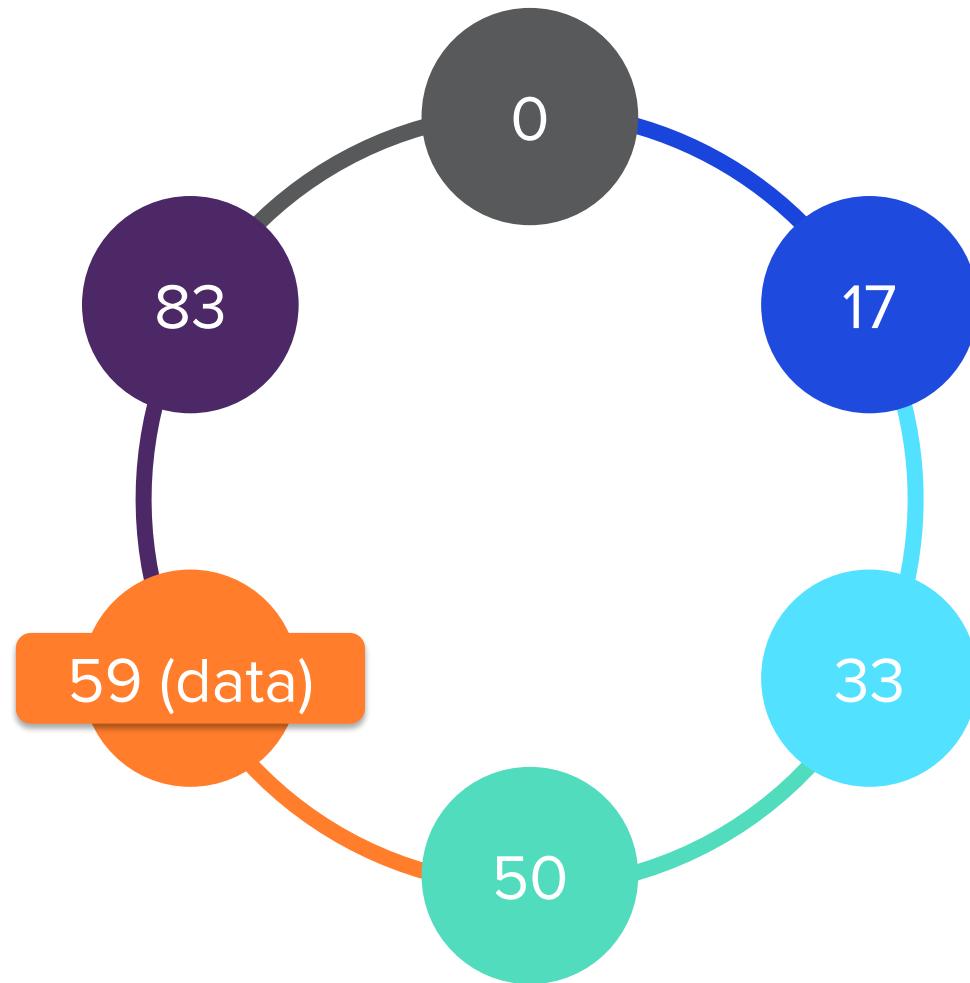
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



Replication within the Ring

RF = 1



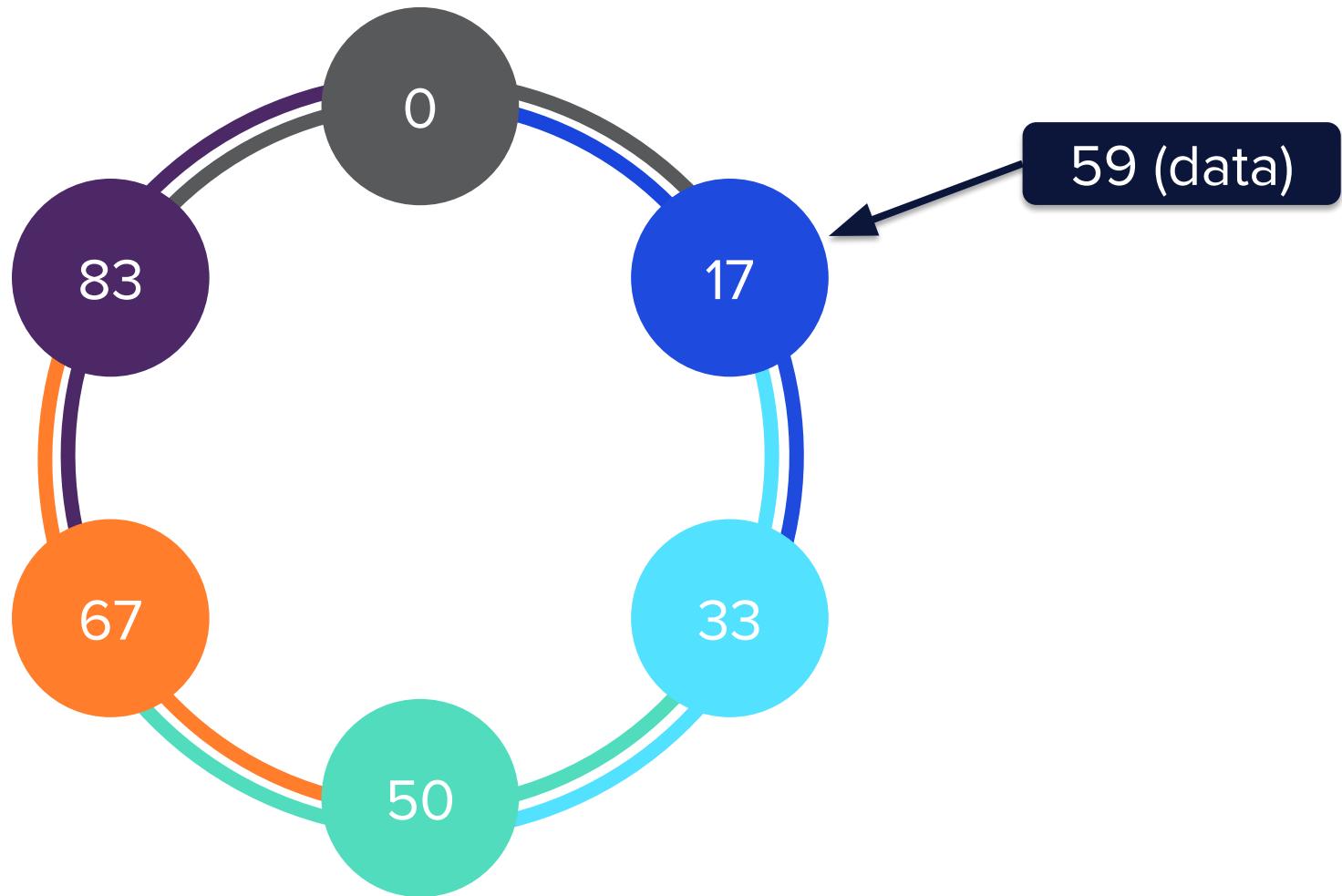
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



Replication within the Ring

RF = 2



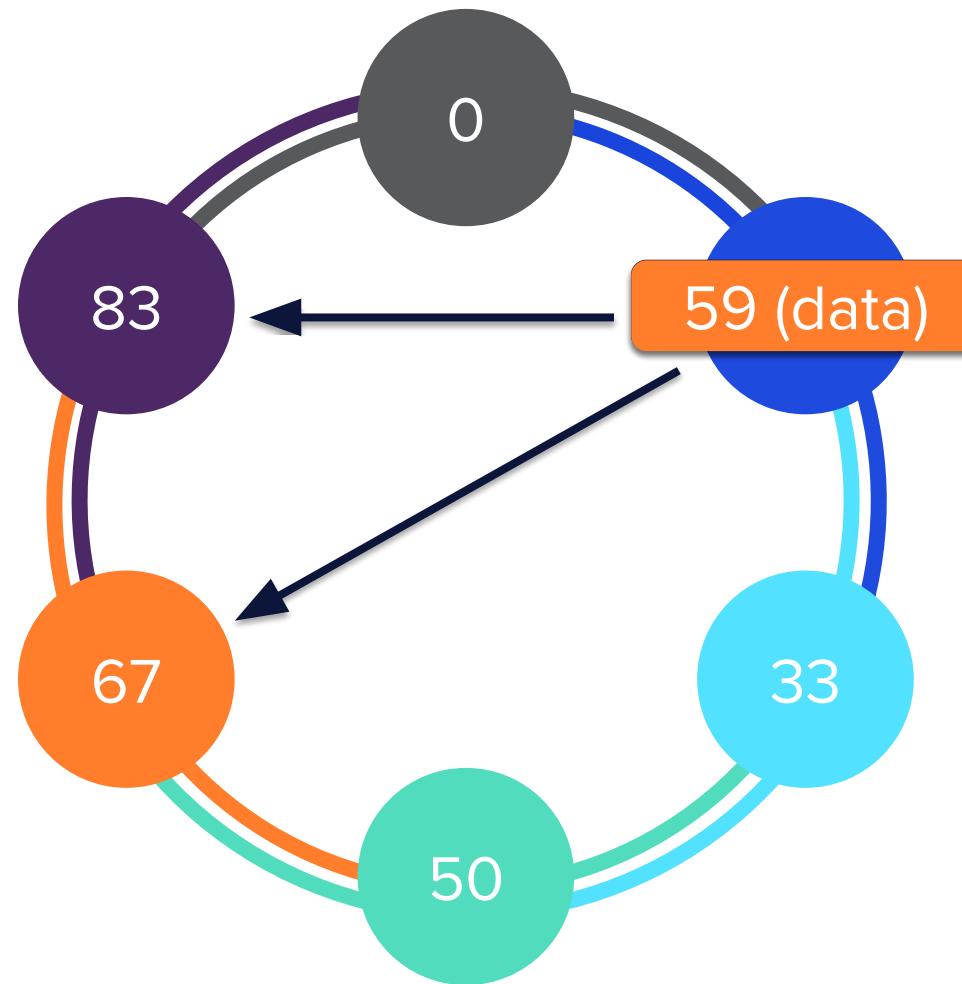
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



Replication within the Ring

RF = 2



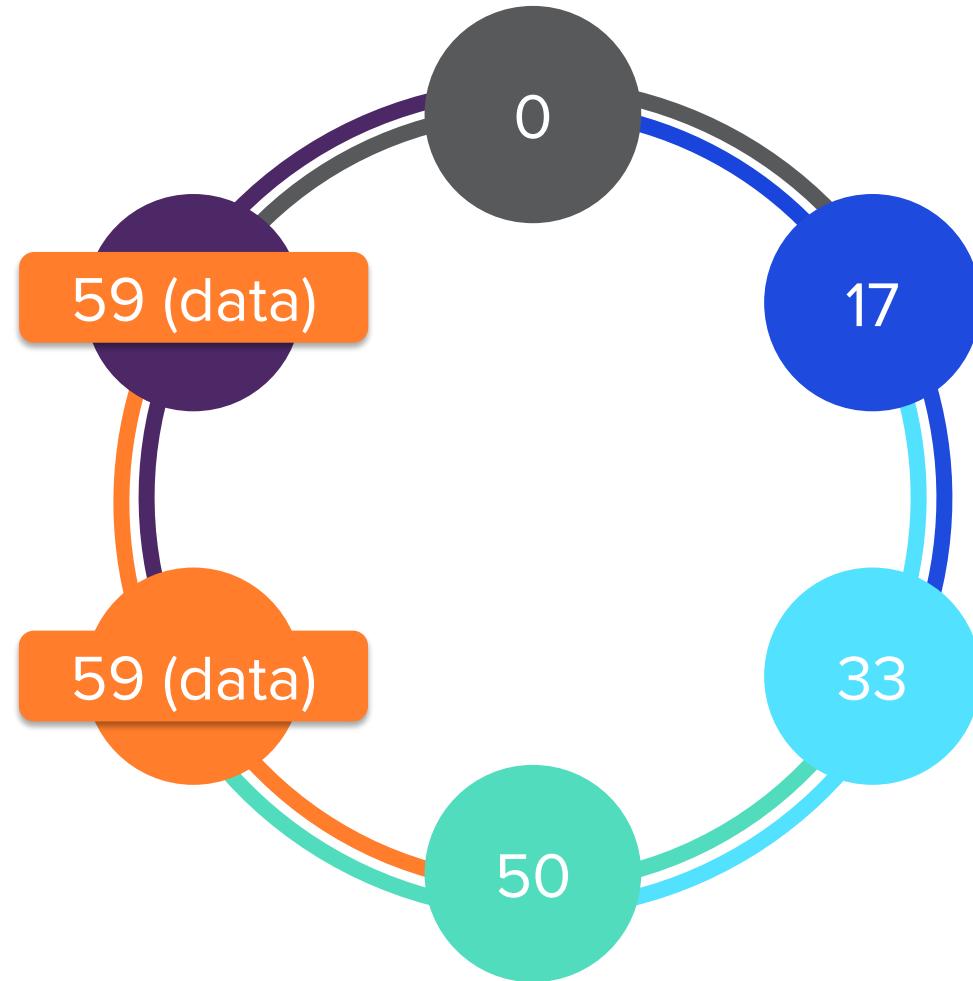
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



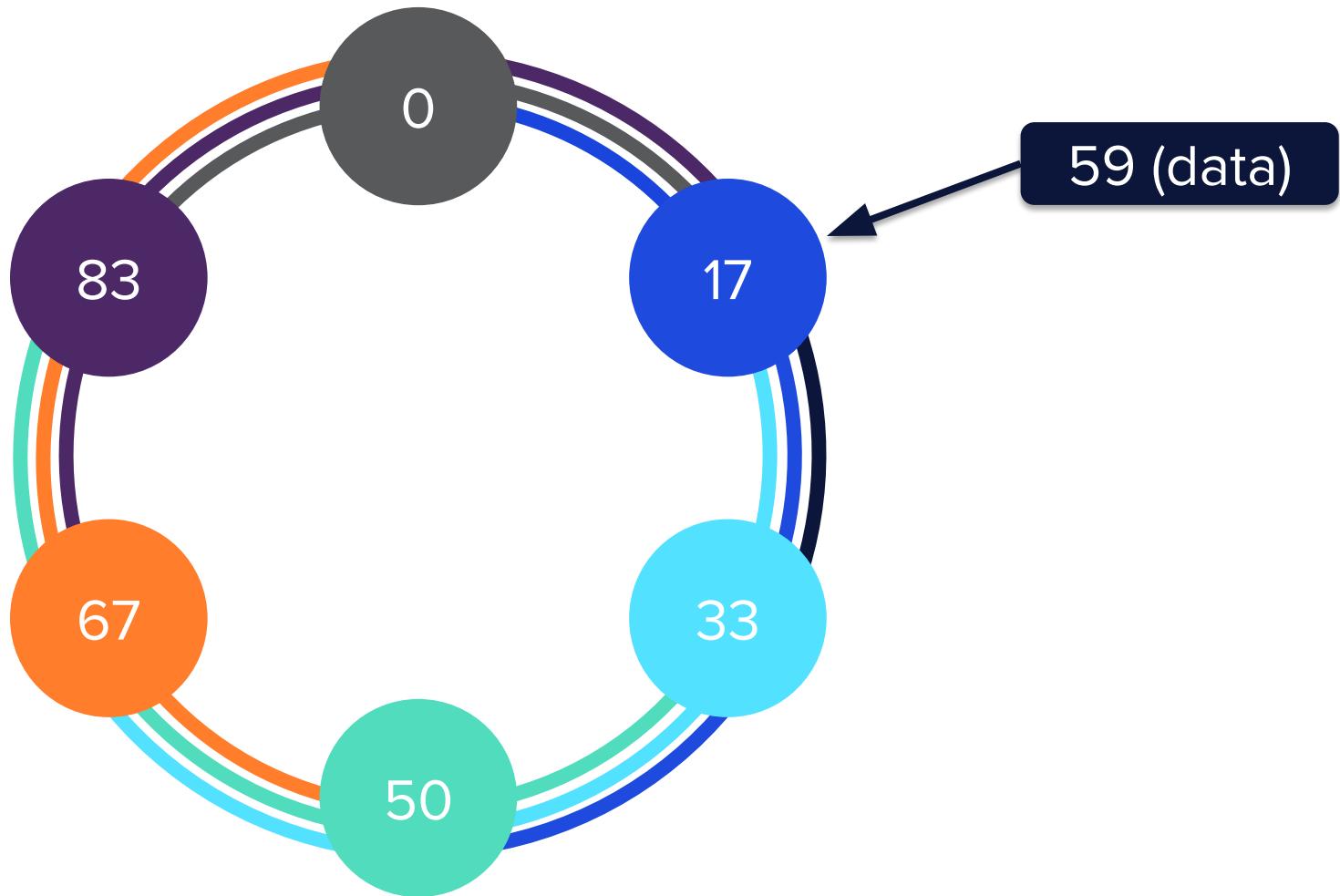
Replication within the Ring

RF = 2



Replication within the Ring

RF = 3



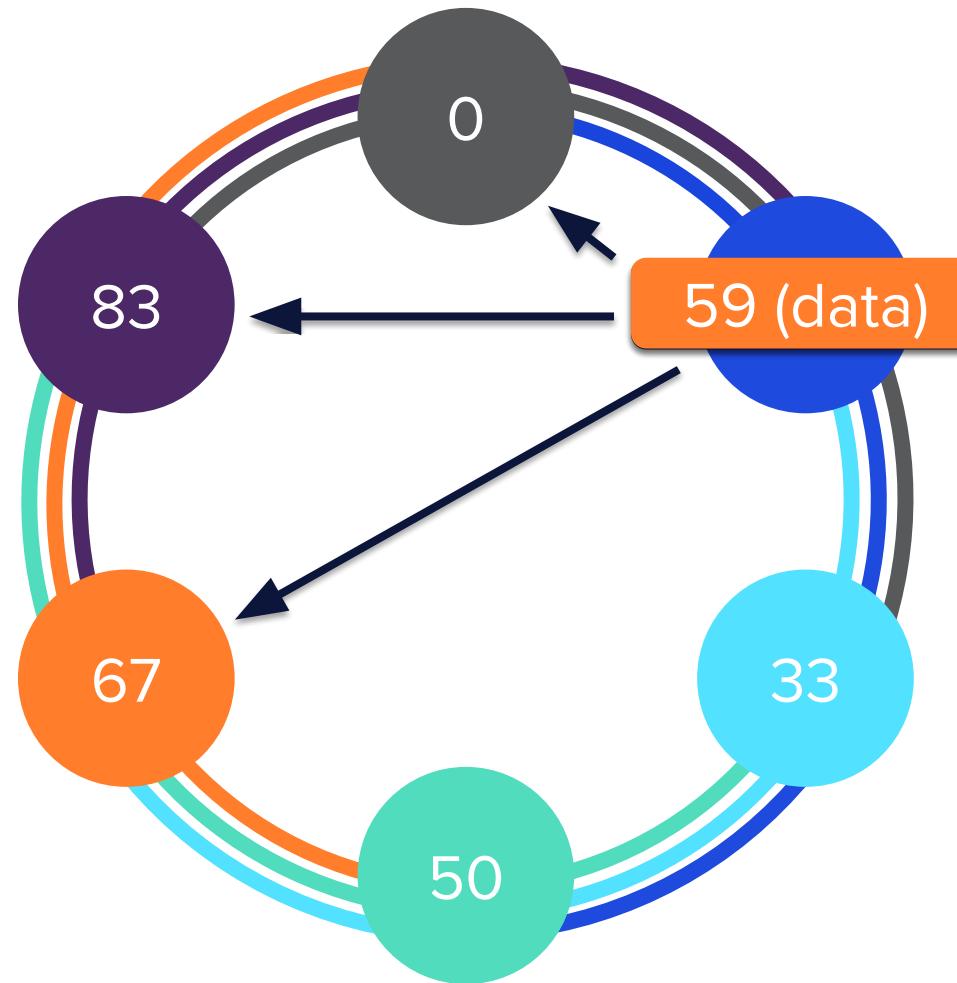
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



Replication within the Ring

RF = 3



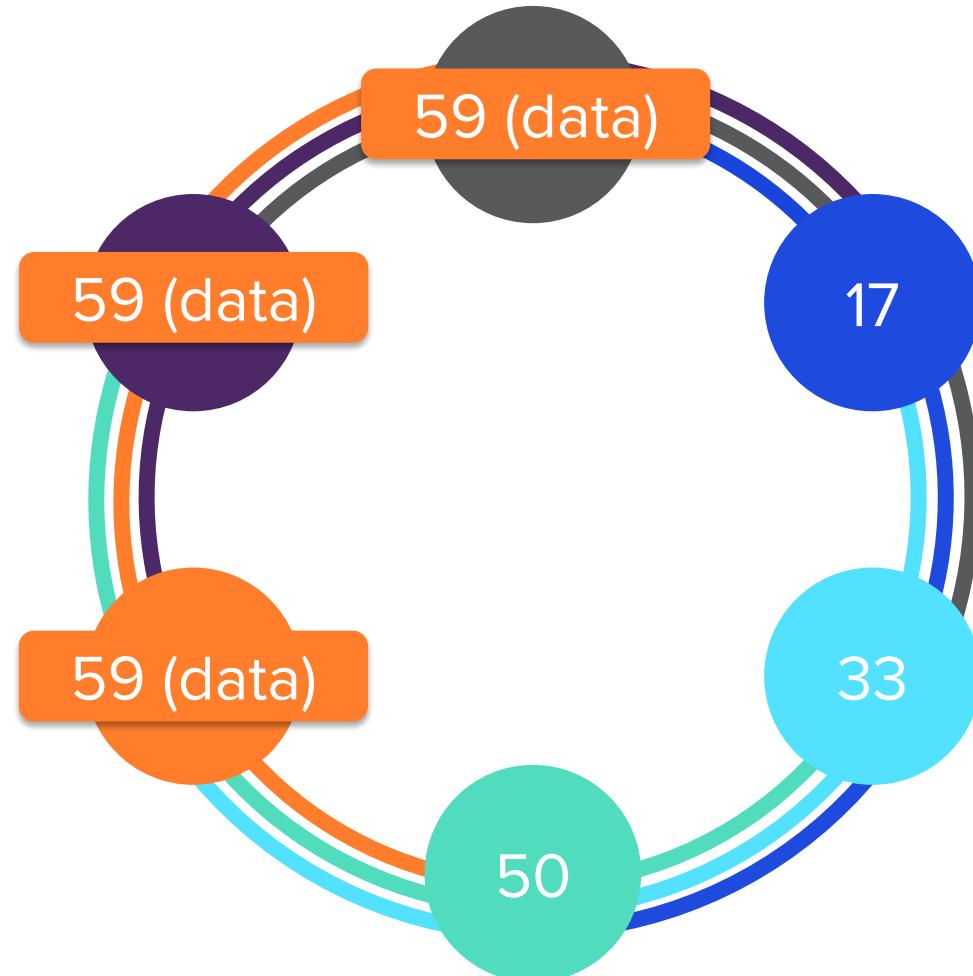
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



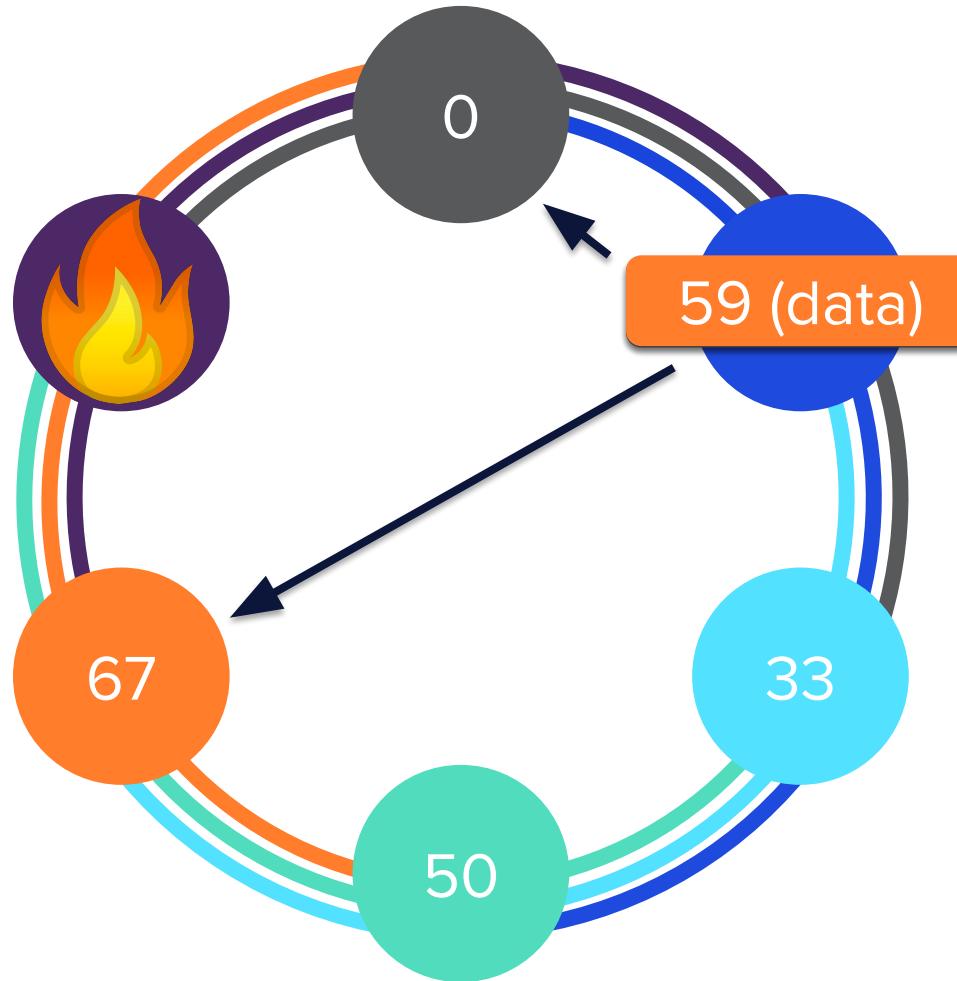
Replication within the Ring

RF = 3



Node Failure

RF = 3



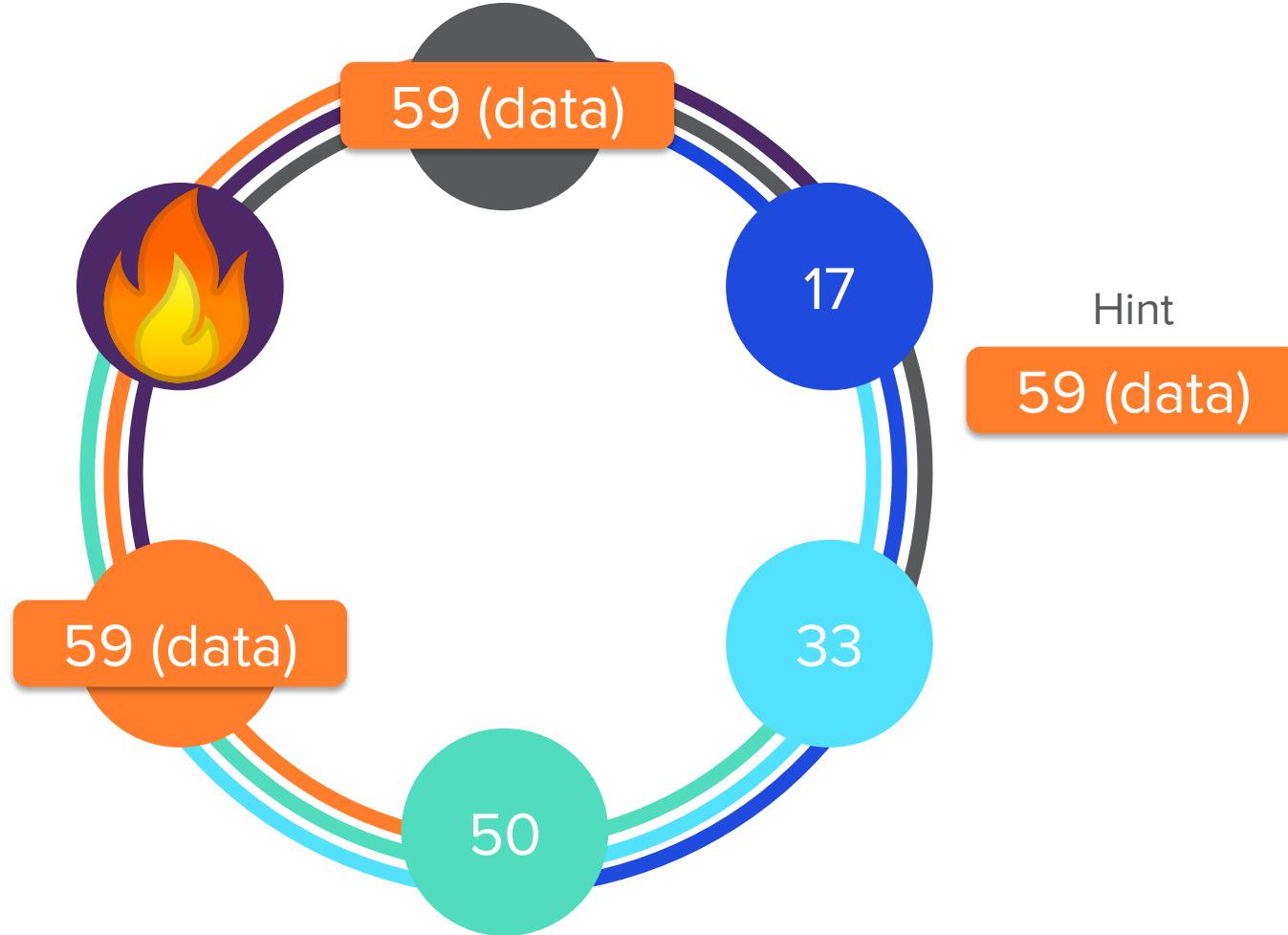
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



Node Failure

RF = 3



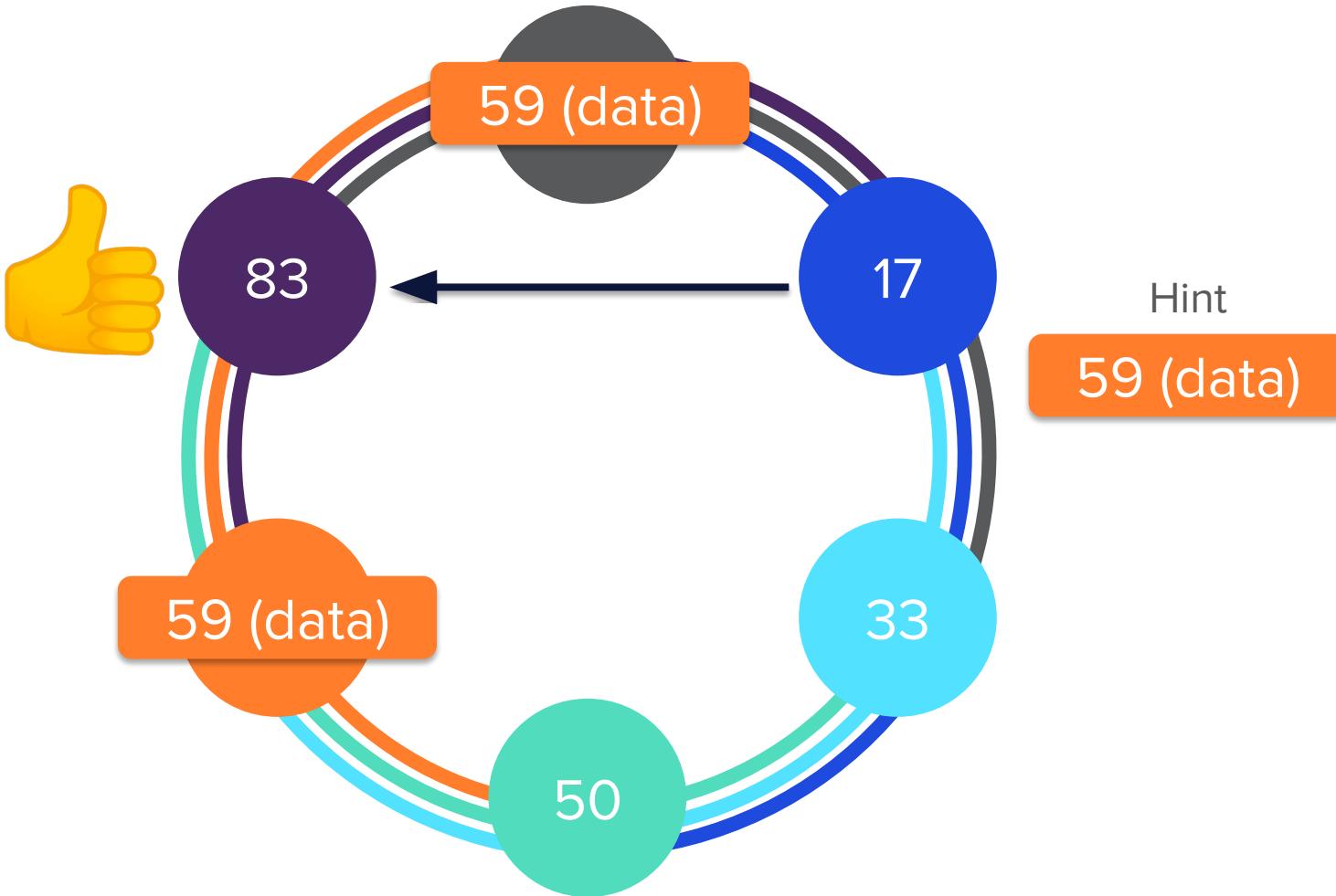
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



Node Failure

RF = 3



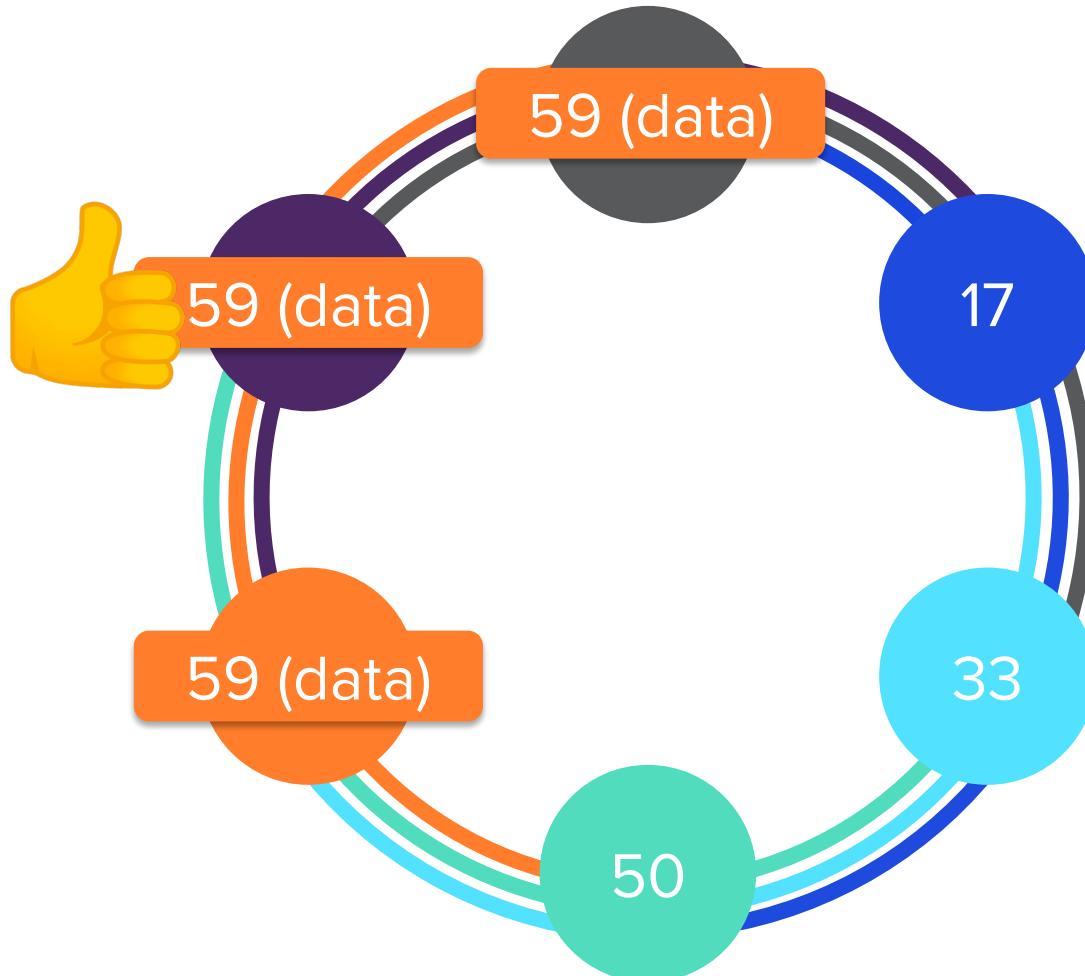
@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



Node Failure – Recovered!

RF = 3

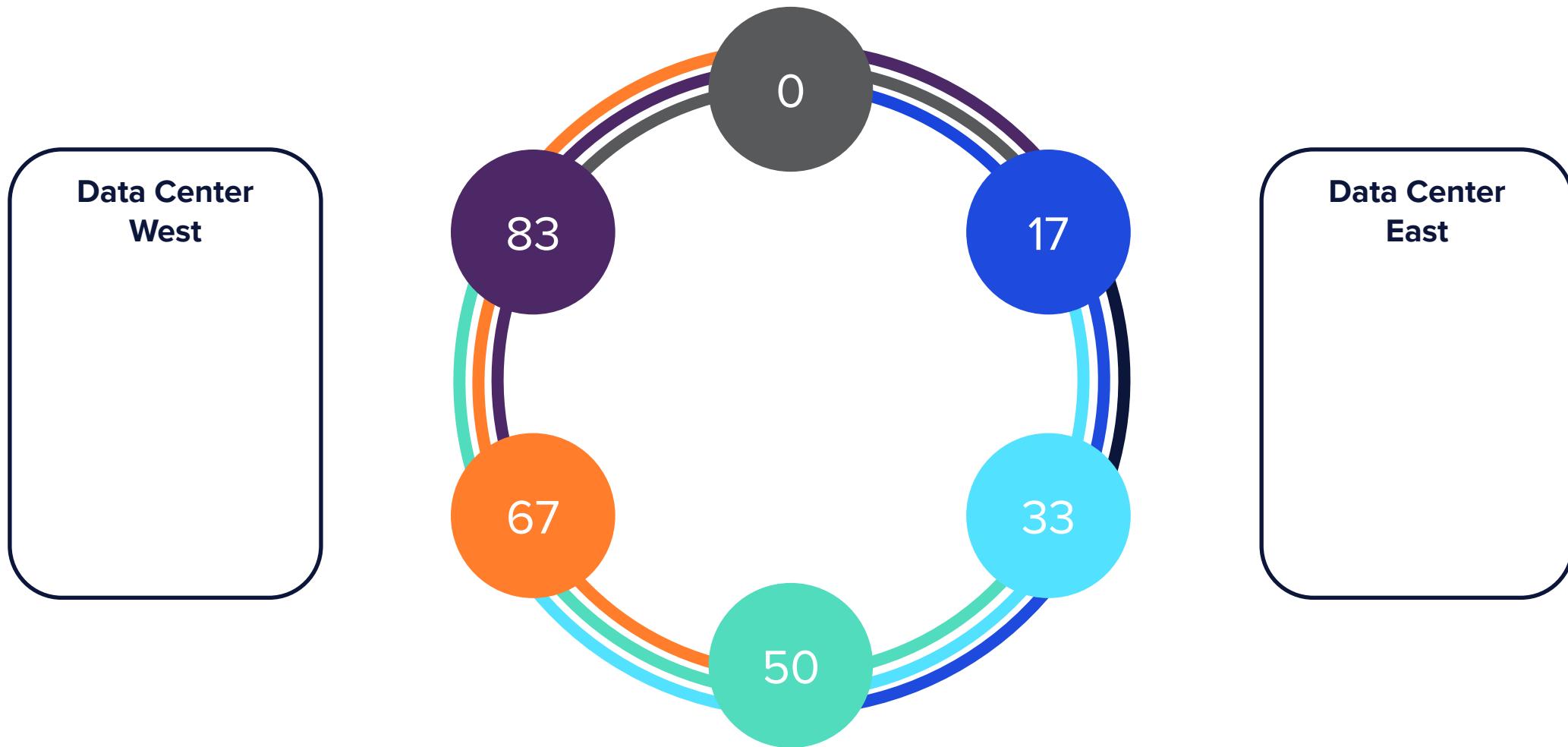


@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



Multi-Data Center Replication

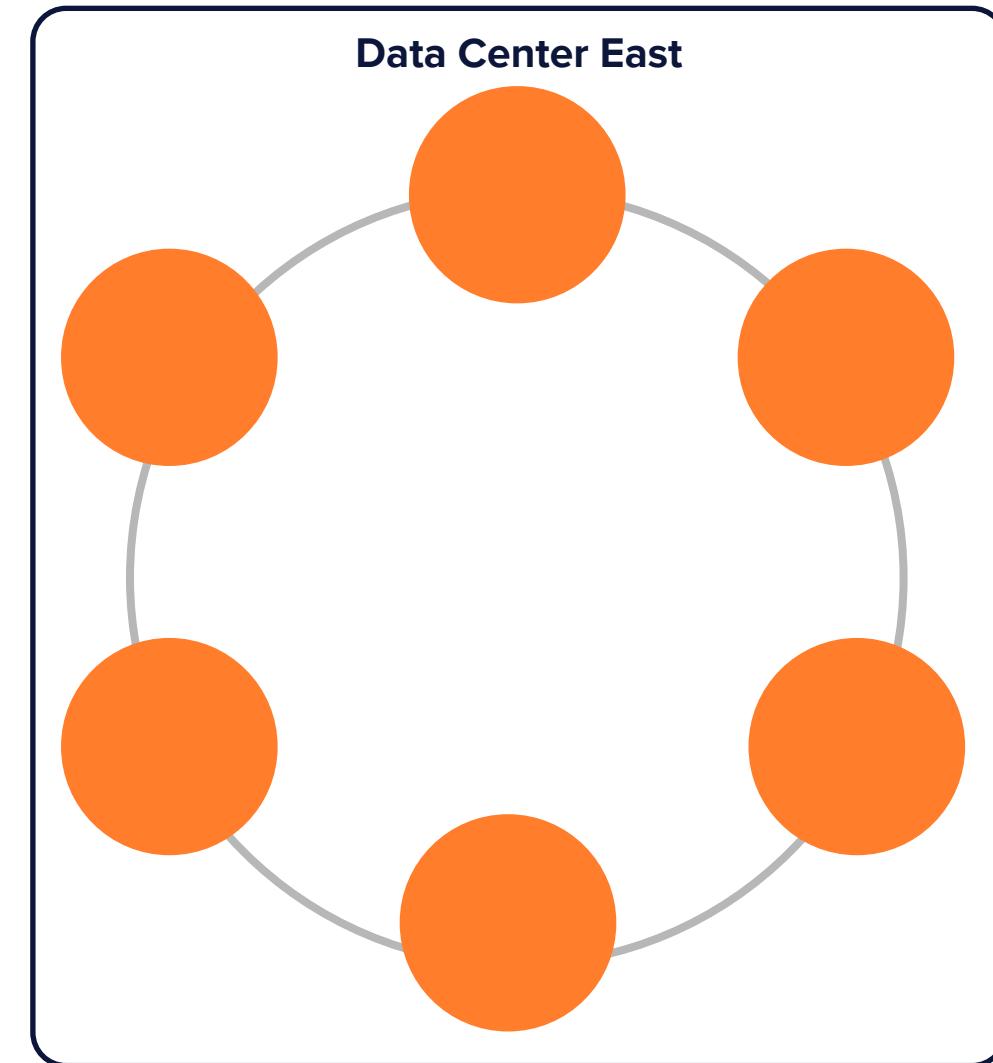
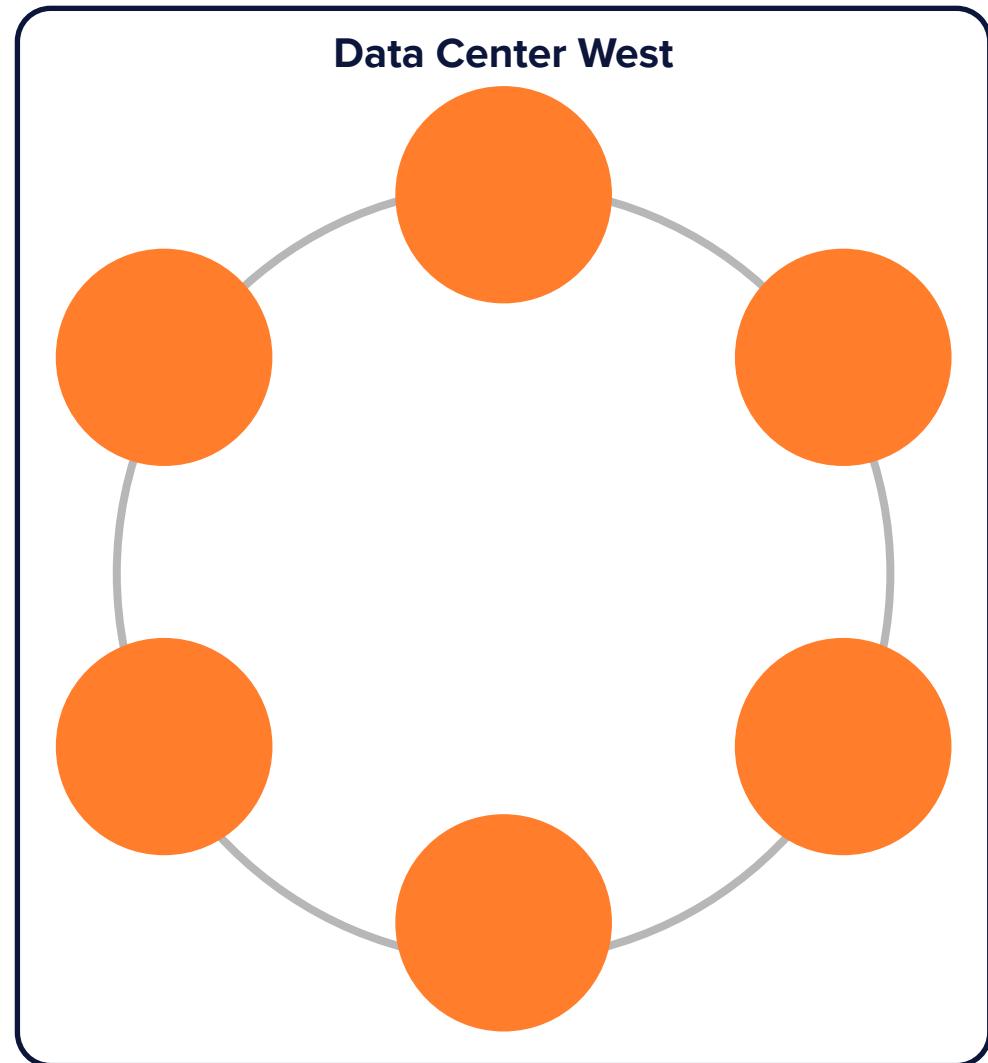


@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



Multi-Data Center Replication (RF=2 in each DC)

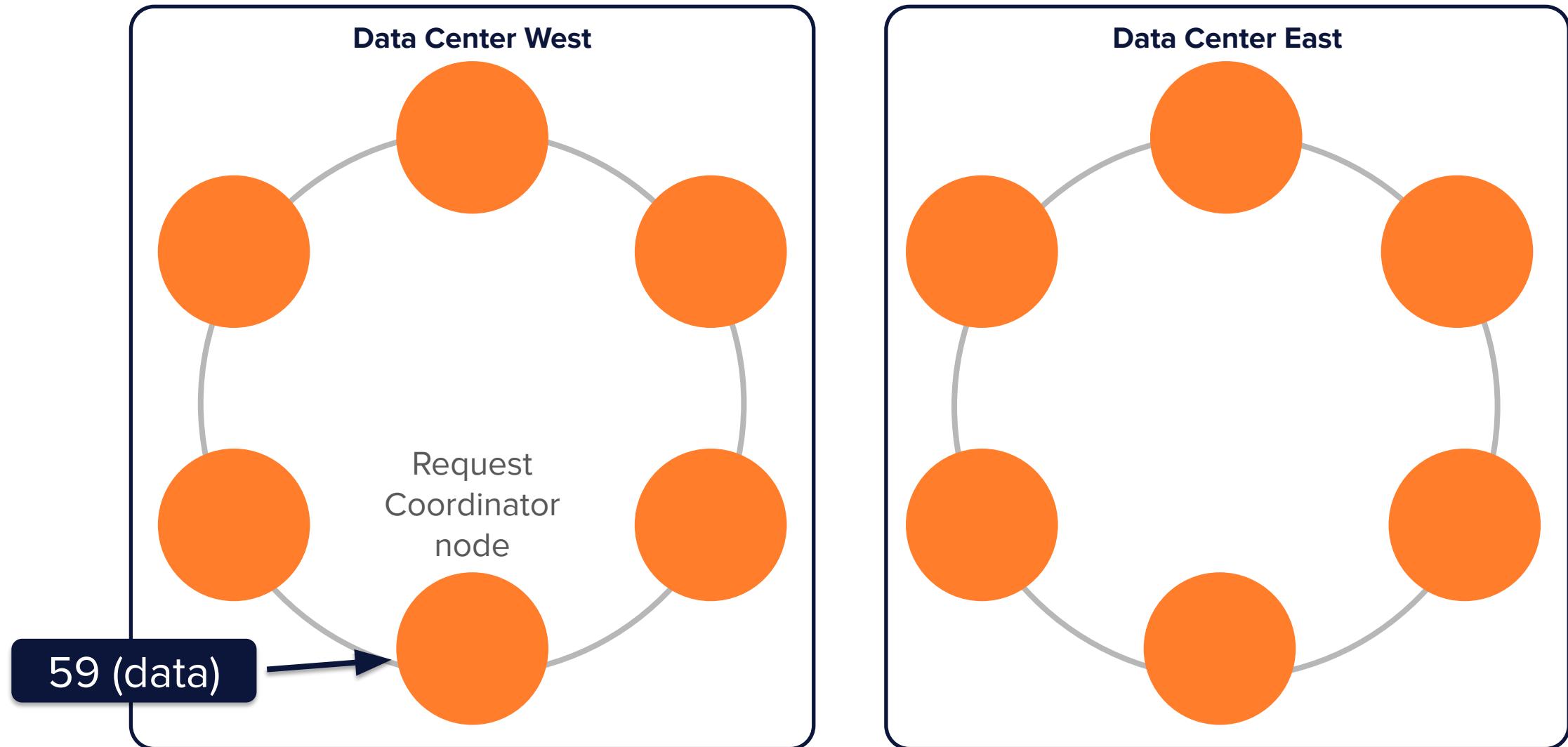


@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>



Multi-Data Center Replication (RF=2 in each DC)

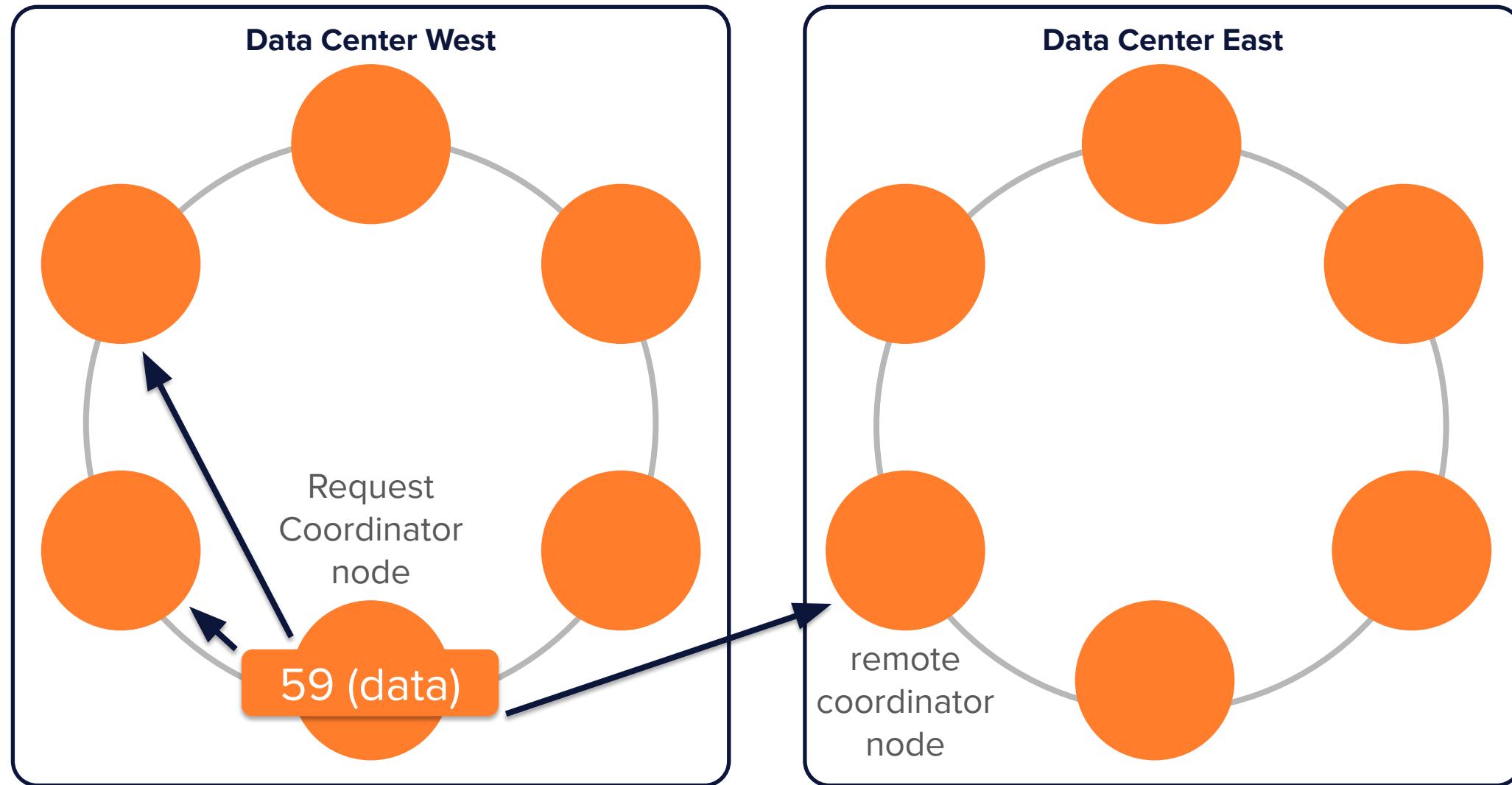


@DataStaxDevs #DataStaxDeveloperDay

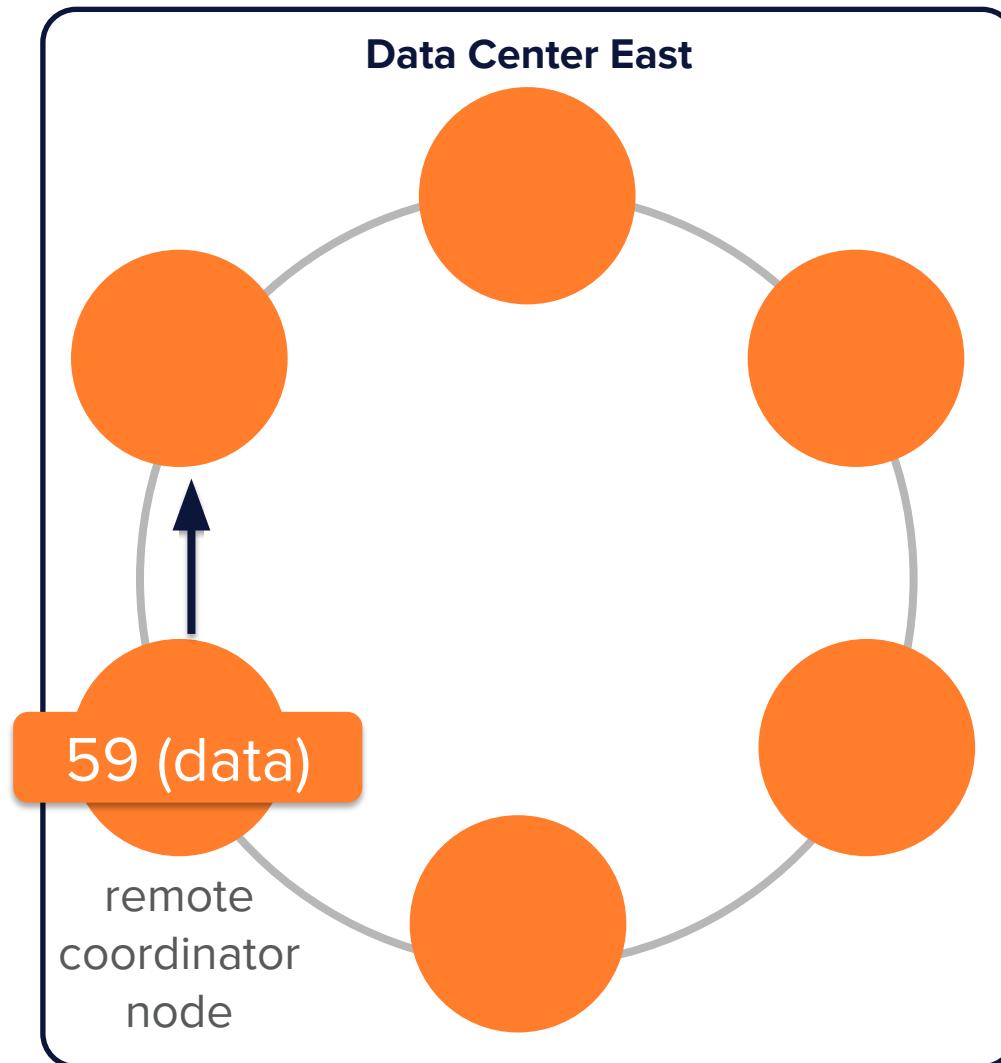
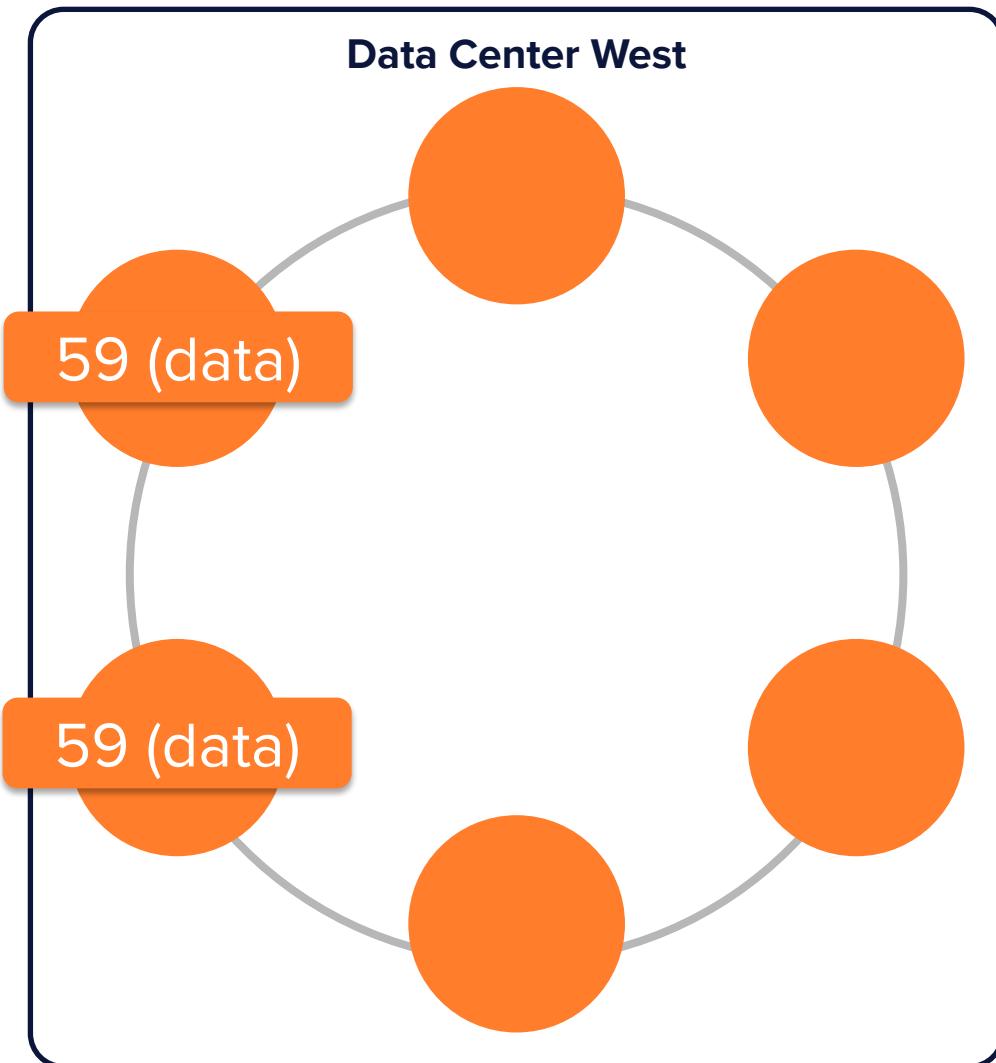
<https://community.datastax.com>



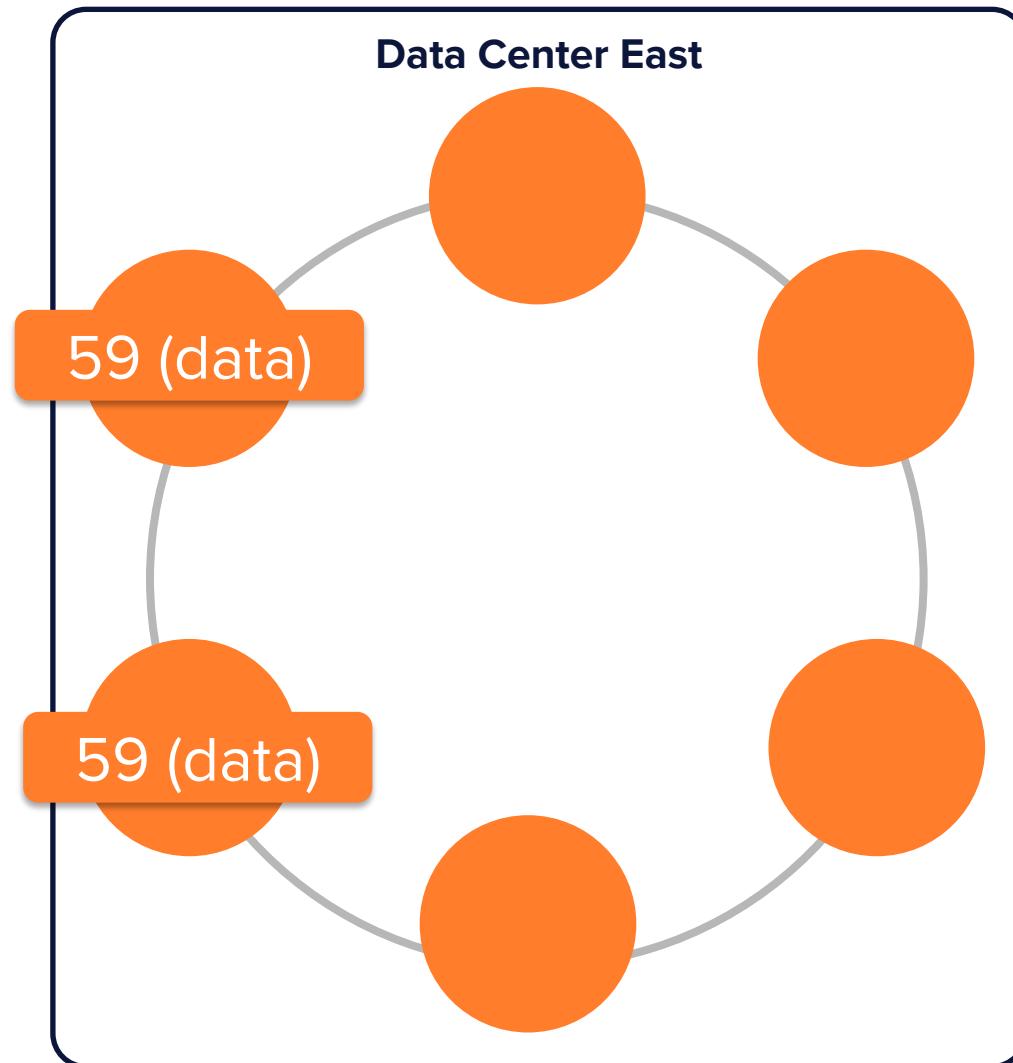
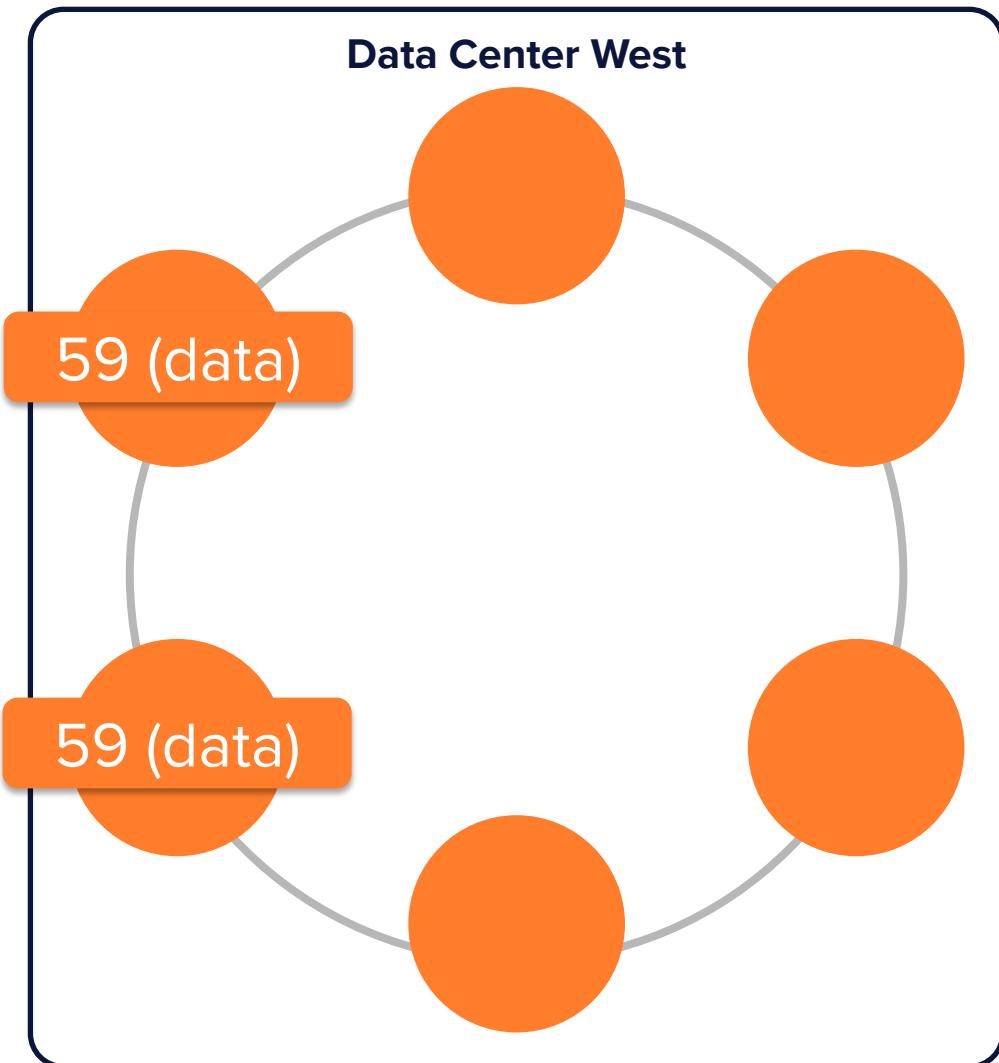
Multi-Data Center Replication (RF=2 in each DC)



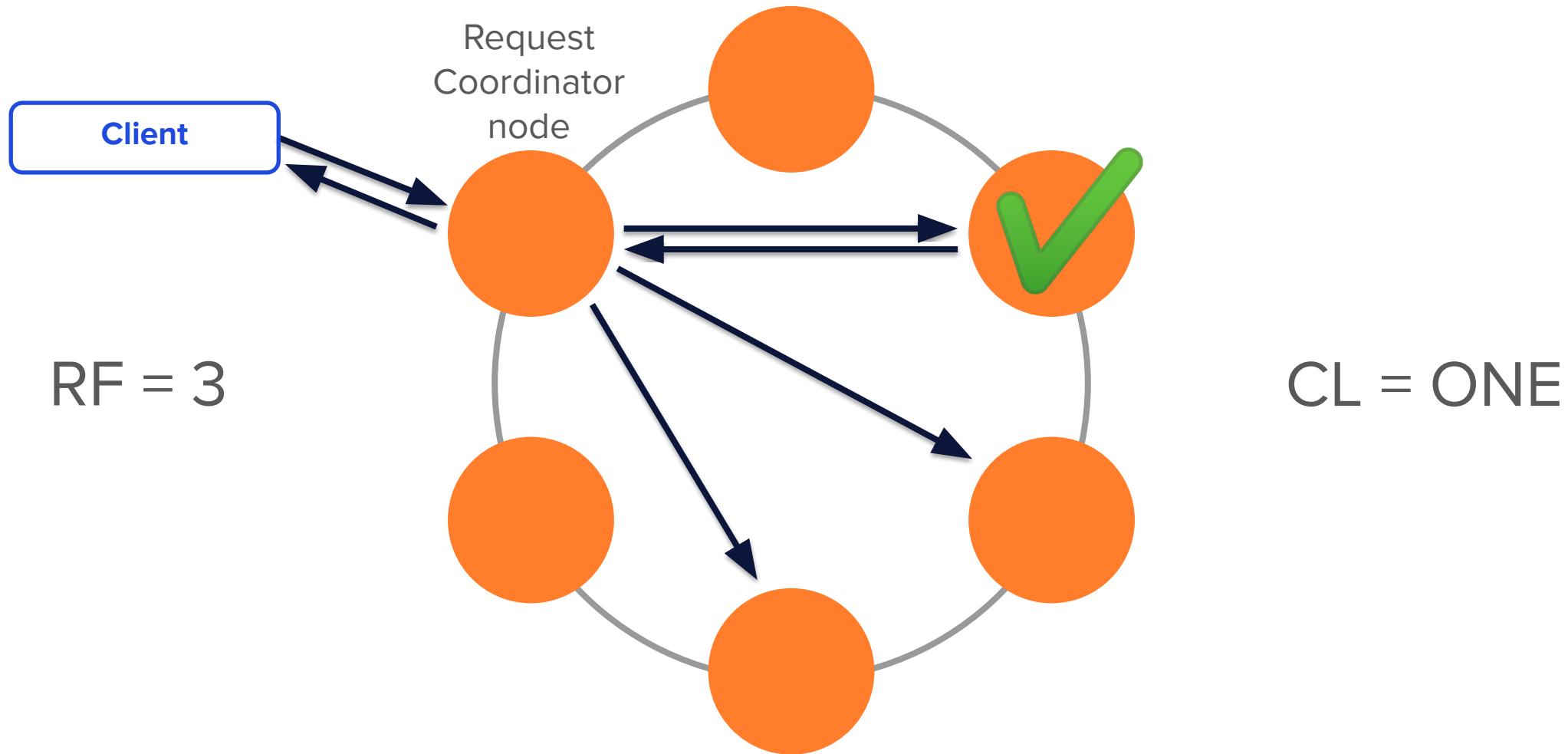
Multi-Data Center Replication (RF=2 in each DC)



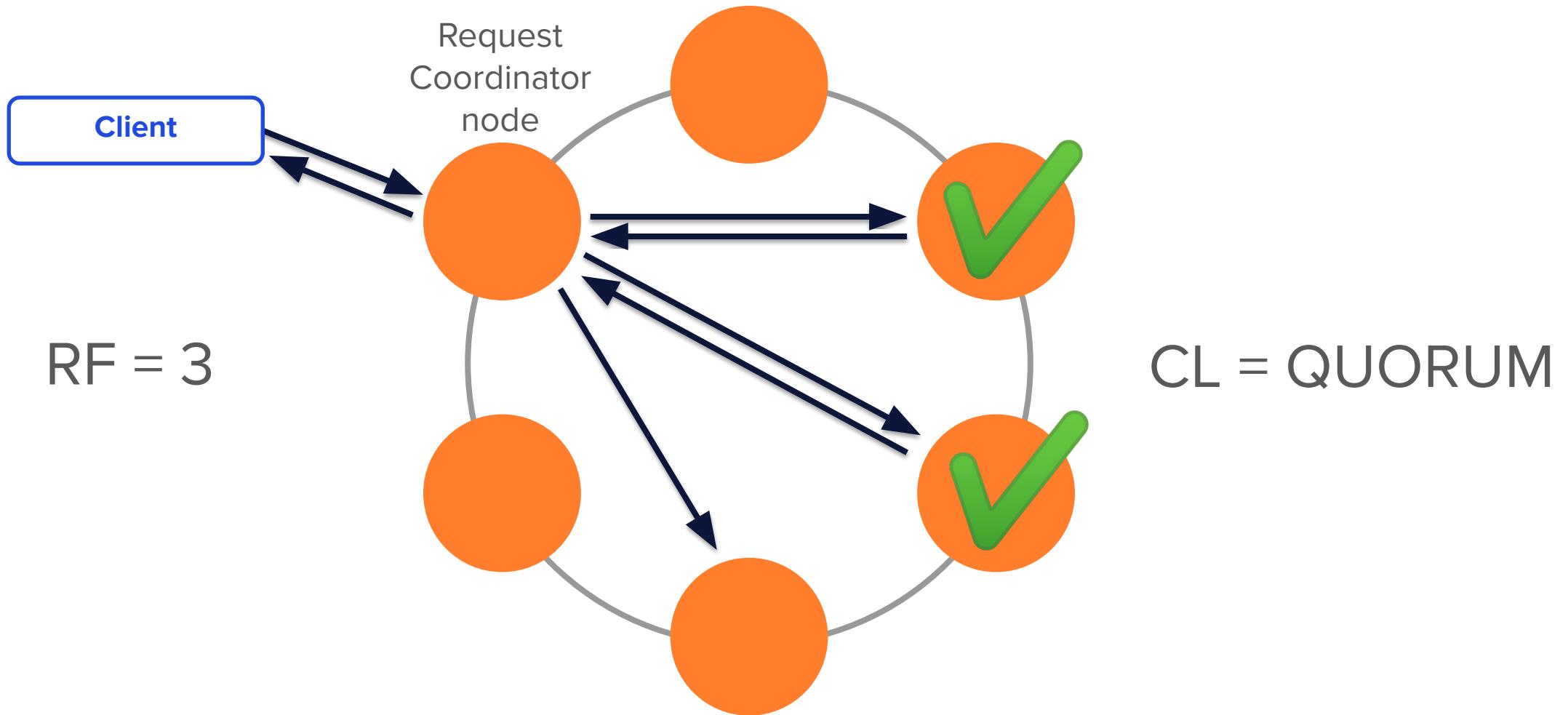
Multi-Data Center Replication (RF=2 in each DC)



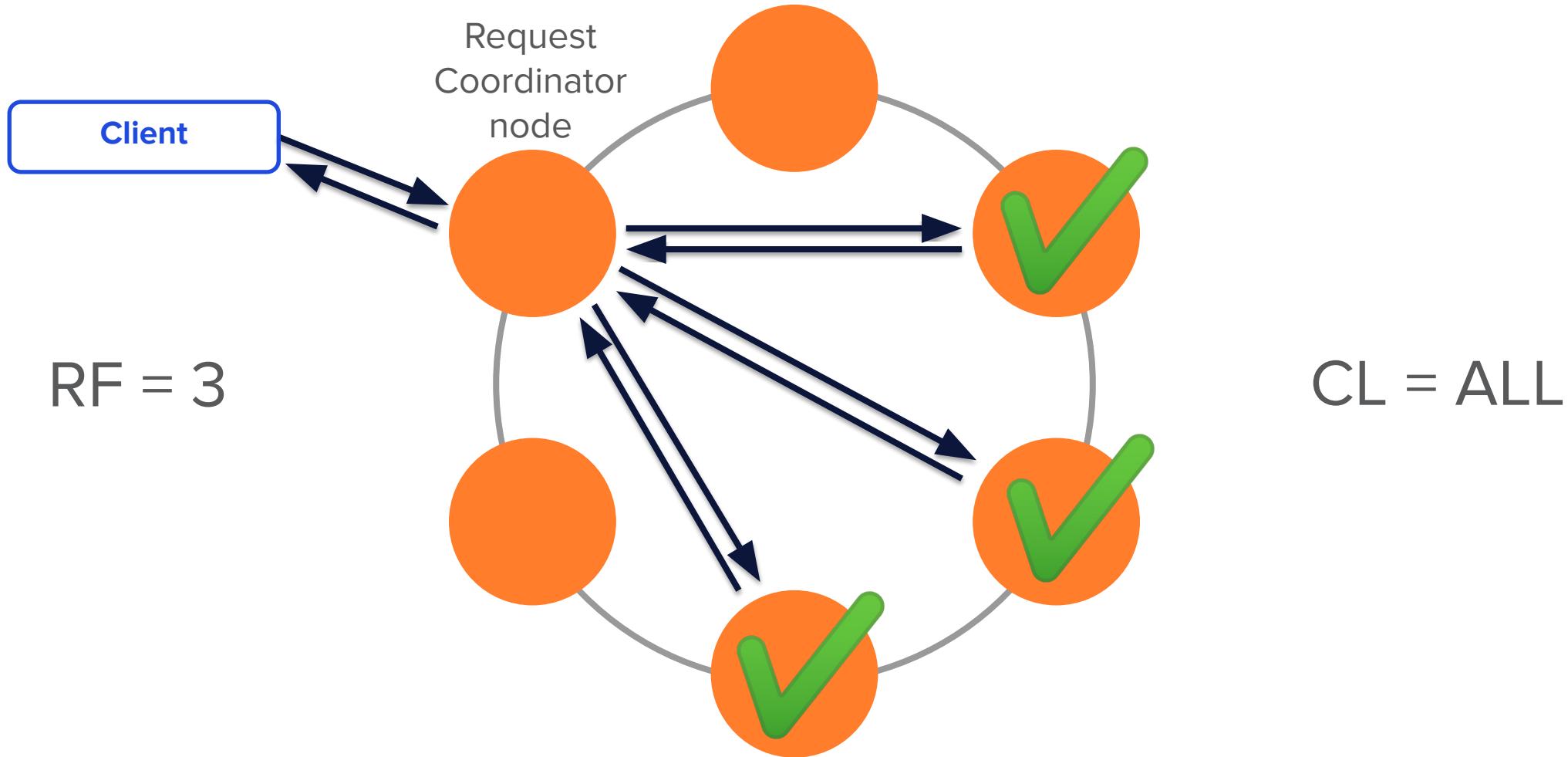
Consistency Levels



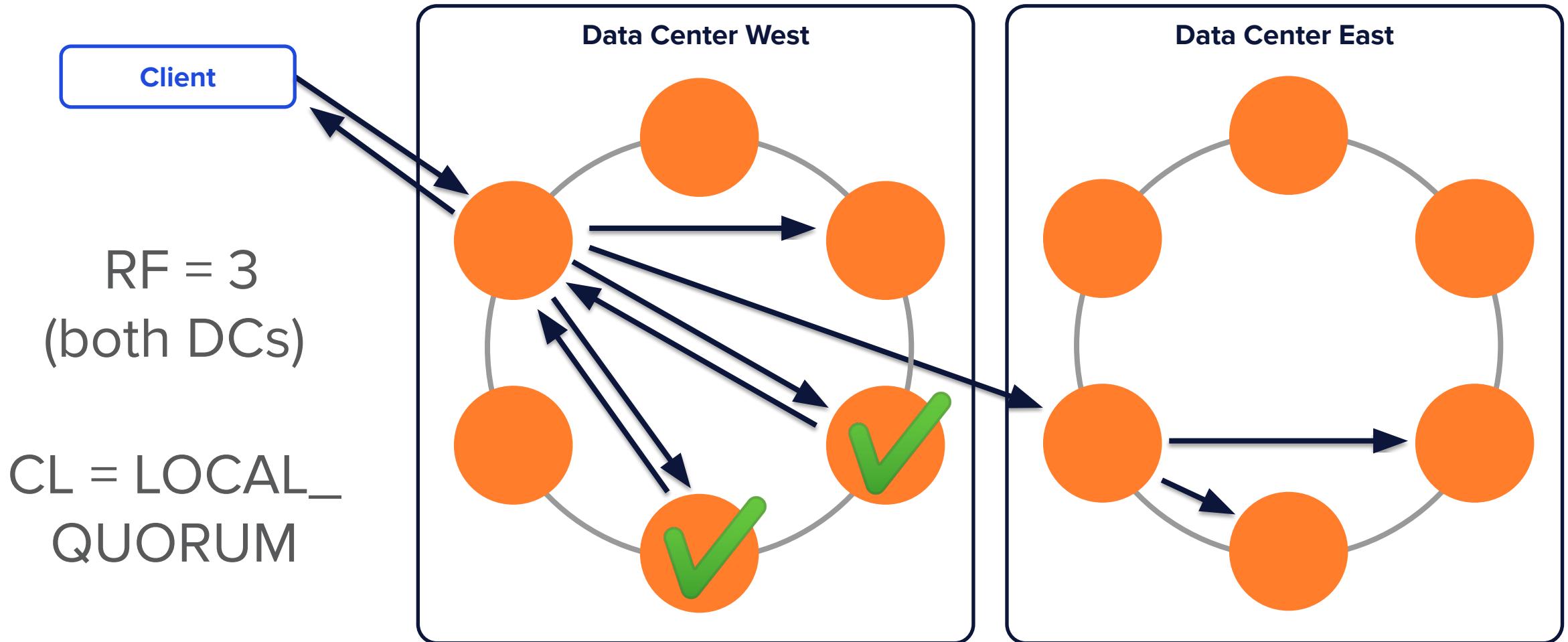
Consistency Levels



Consistency Levels



Consistency Across Data Centers

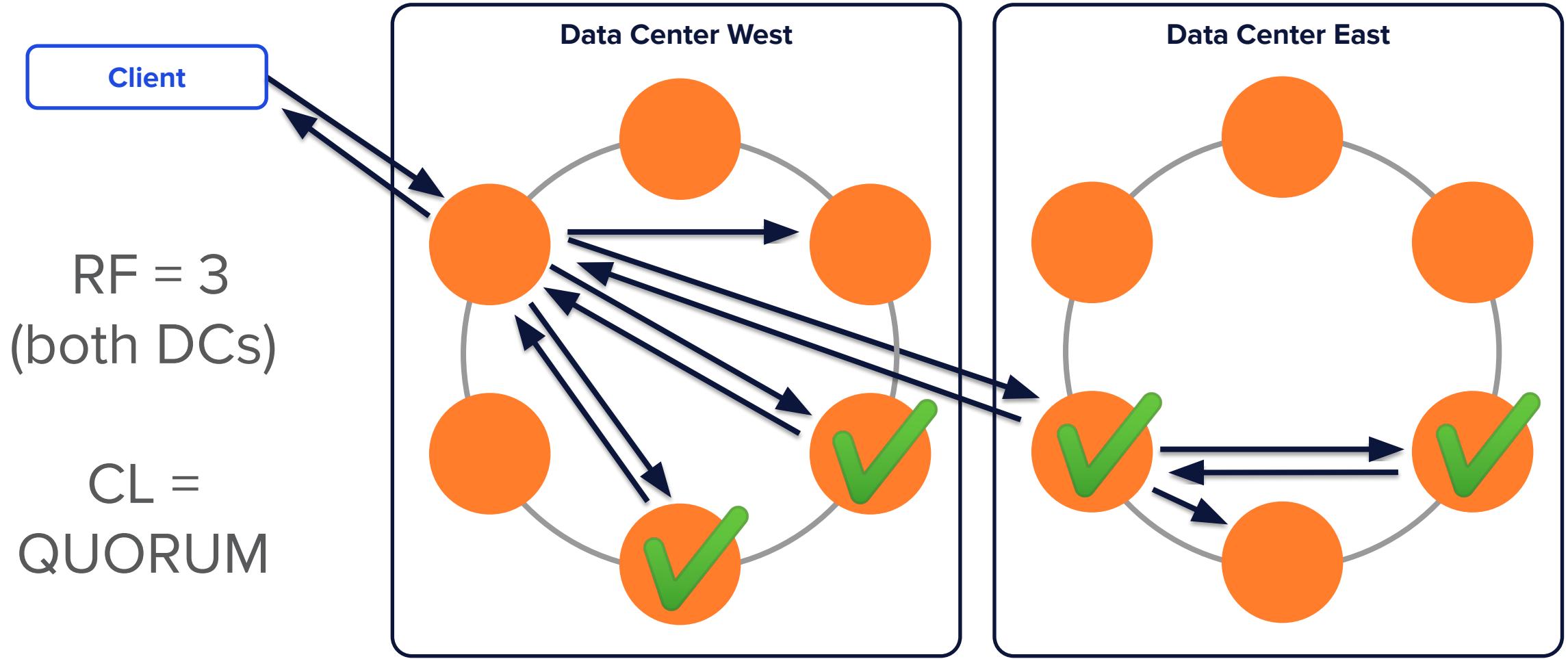


@DataStaxDevs #DataStaxDeveloperDay

<https://community.datastax.com>

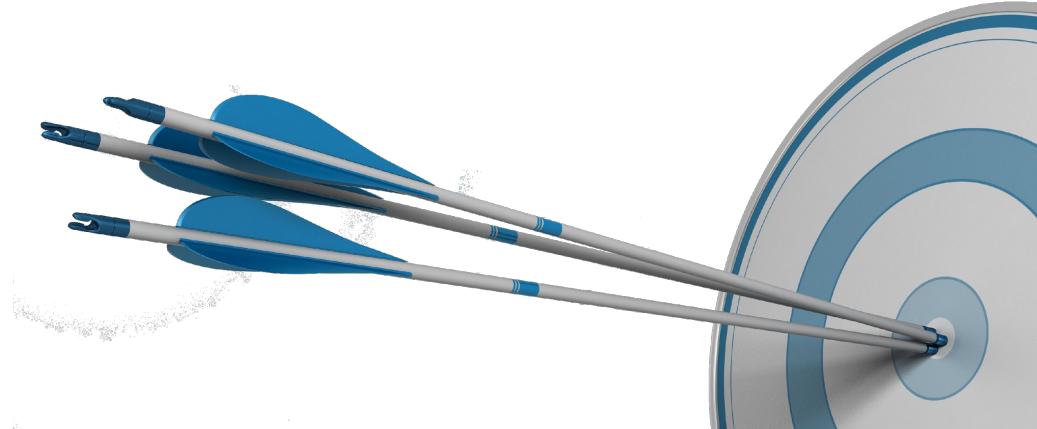


Consistency Across Data Centers

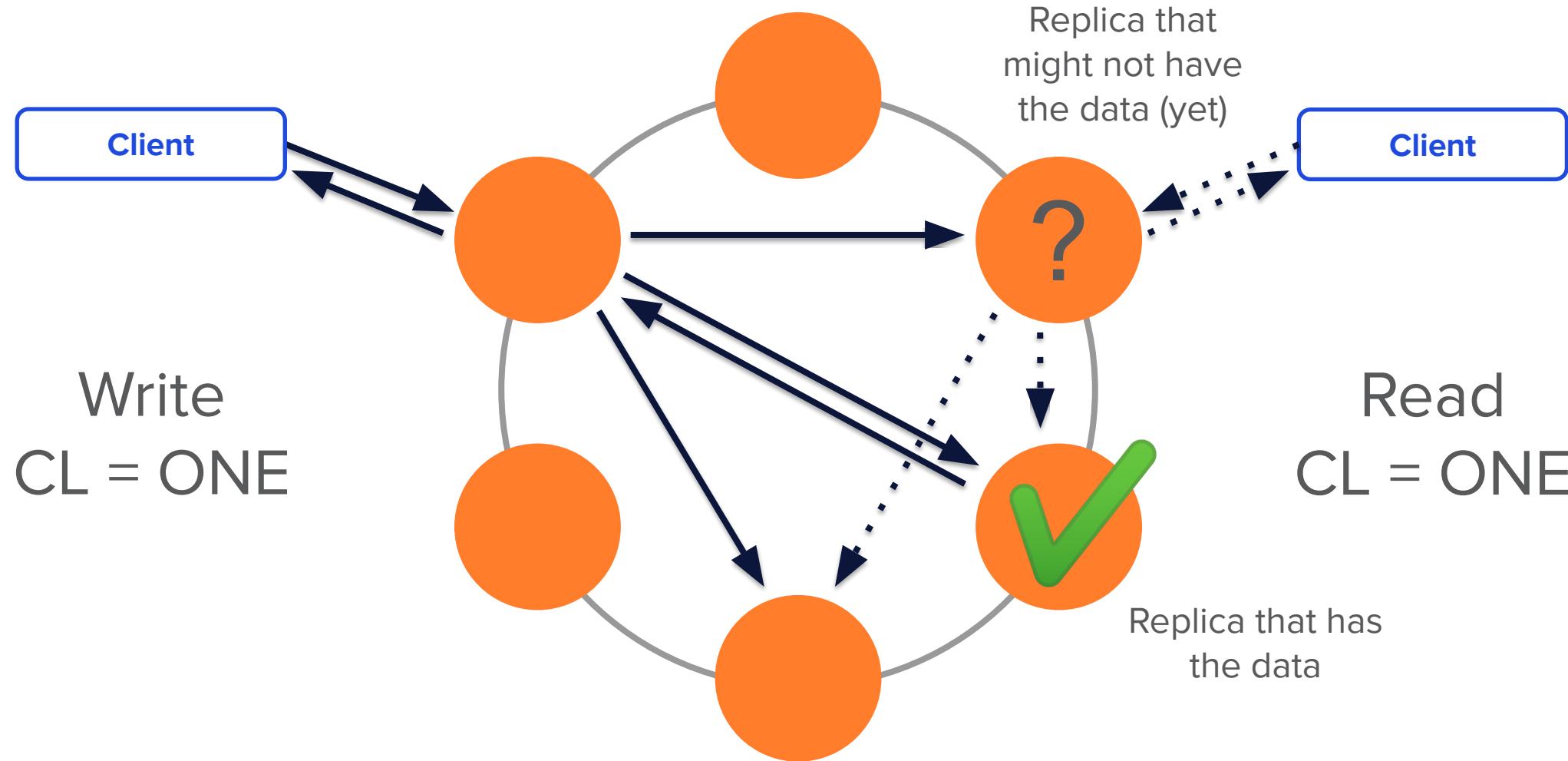


Cassandra Data Availability Concepts

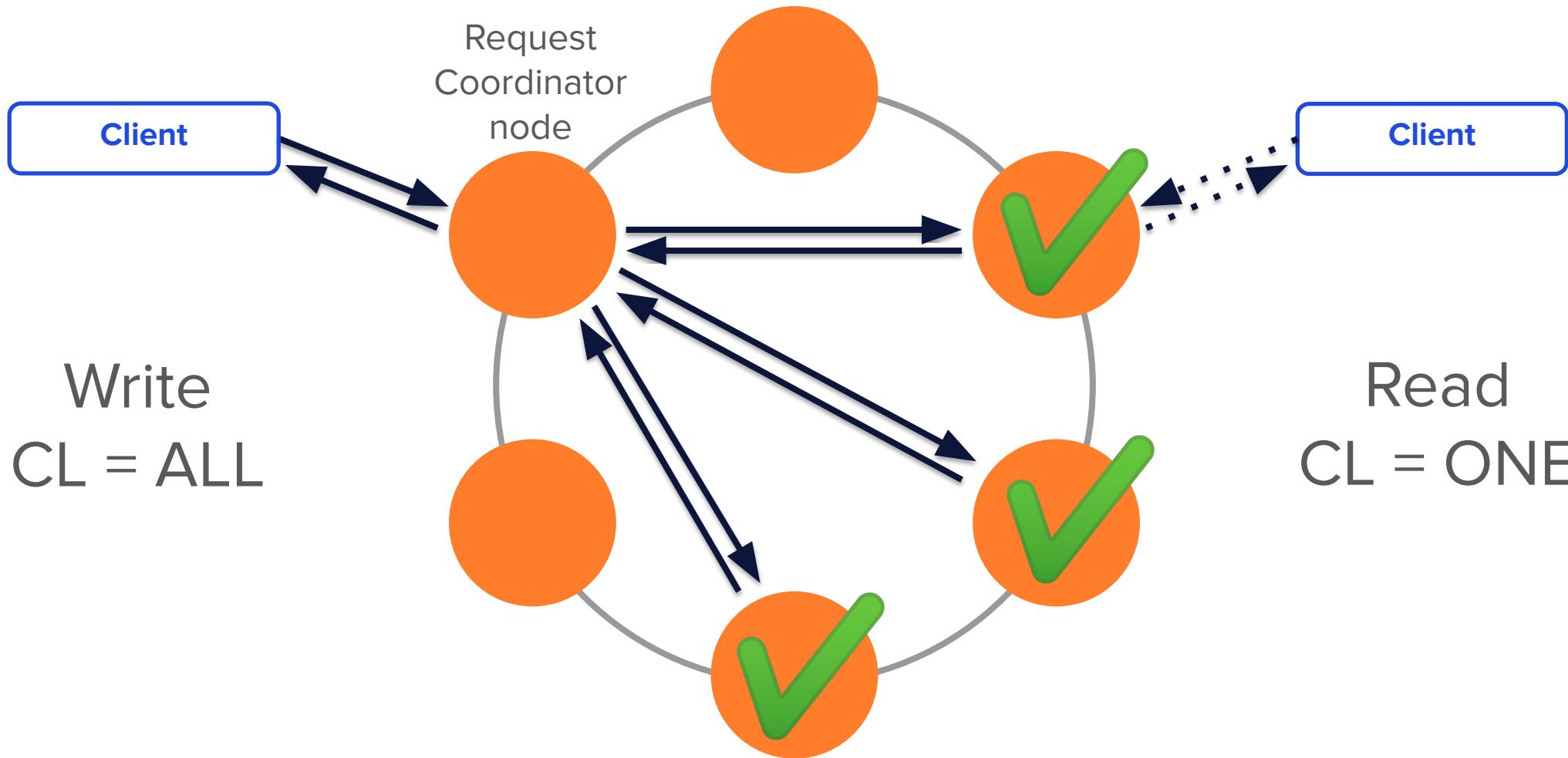
- Replication Factor – the number of copies of your data
 - Increasing replication increases chances of availability
 - Increasing replication increases chances of inconsistency
- Consistency Level – the number of acknowledged copies read/written
- Immediate Consistency: number of writes + number of reads > replication factor
- EXAMPLE:
 - Replication Factor = 3
 - Read/Write Consistency Level = QUORUM
 - $2 + 2 = 4 > 3$ 



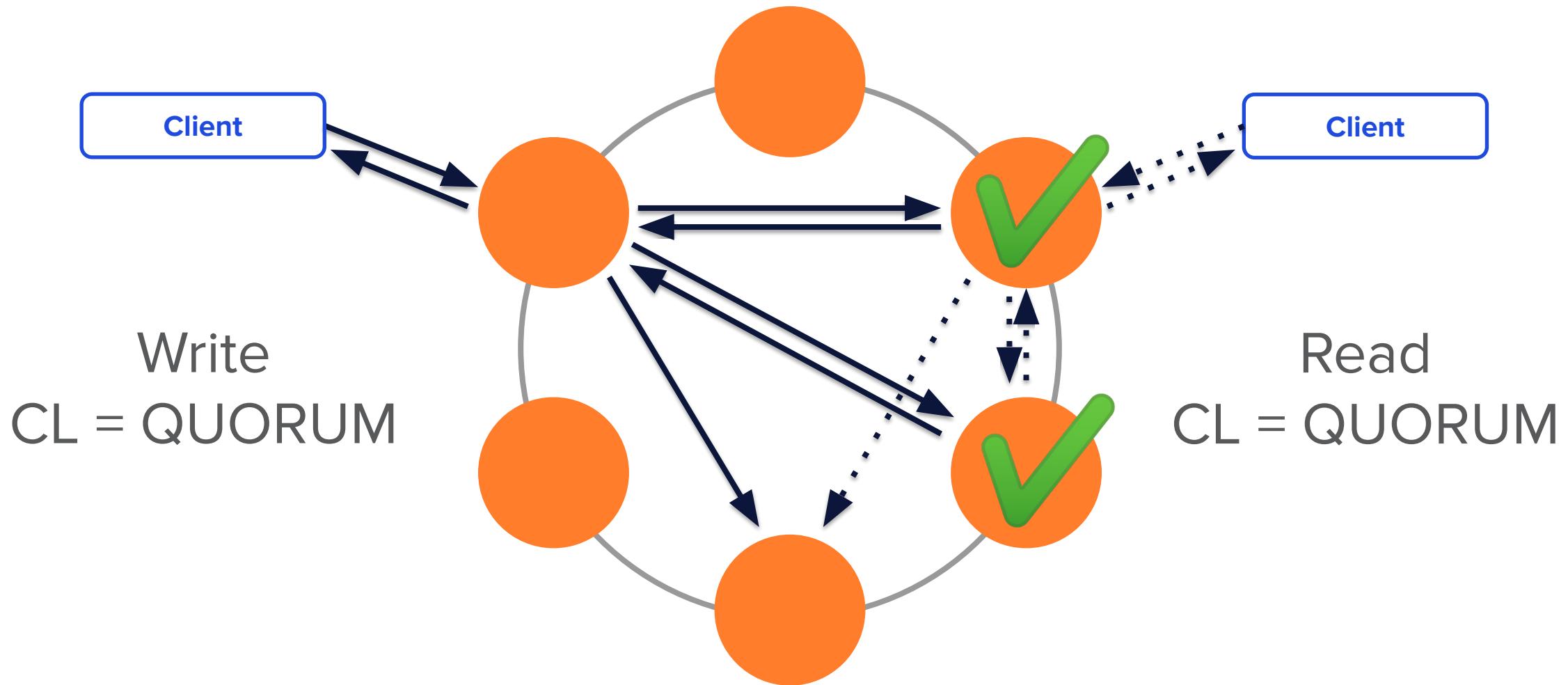
Weak(er) Consistency



Immediate Consistency – One Way



Immediate Consistency – A Better Way



Consistency Settings

- Weakest to strongest

Setting	Description
ANY	Storing a hint at minimum is satisfactory
ONE, TWO, THREE	Checks closest node(s) to coordinator
QUORUM	Majority vote, $(\text{sum_of_replication_factors} / 2) + 1$
LOCAL_ONE	Closest node to coordinator in same data center
LOCAL_QUORUM	Closest quorum of nodes in same data center
EACH_QUORUM	Quorum of nodes in each data center, applies to writes only
ALL	Every node must participate

Consistency Tradeoffs

- The higher the consistency, the less chance you may get stale data
 - Pay for this with POSSIBLE latency
 - Depends on your situational needs
- IoT and Sensor systems with write-heavy workloads may benefit from CL=ONE for writes if data isn't required immediately



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	

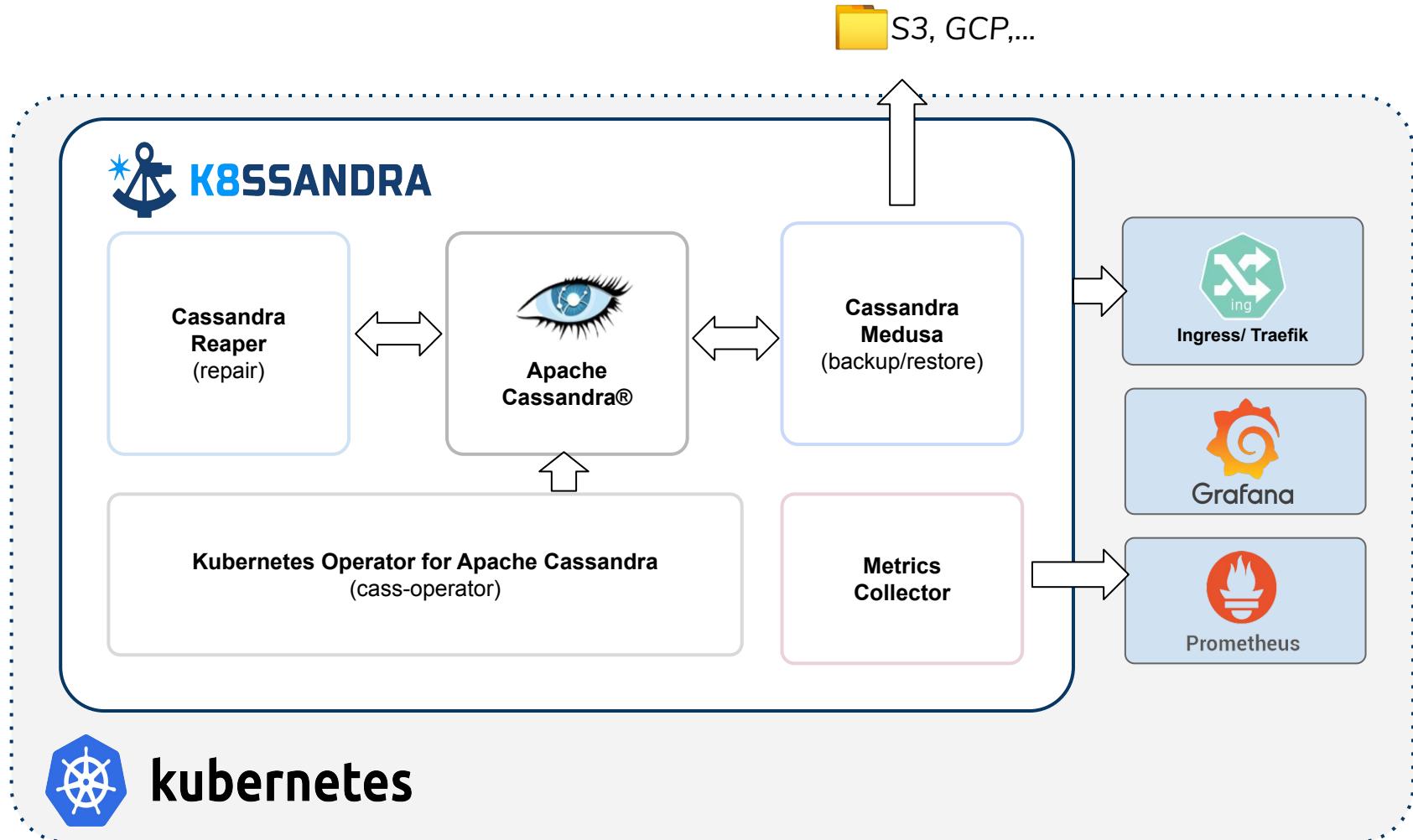
Cloud-native

Modern Cloud Applications



API Layer	
Hybrid-cloud	
Enterprise Data Layer	
Multi-cloud	

K8ssandra



Apache Cassandra™ with Kubernetes

- 1 Housekeeping and Quiz
- 2 Cassandra Basics
- 3 Setting Up and Monitoring Cassandra
- 4 Working with Data
- 5 Scaling Up and Down
- 6 Running Repairs
- 7 Backing Up and Restoring Data
- 8 Resources

Pre-work for today's workshop

- 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!!!!

Pre work for today's workshop

- 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!!!

Pre work for today's workshop

- Have access to your Datastax Learning cloud instance
- Be willing to learn
- Have a pulse. (unless you are undead)

K8ssandra



+



=



Setting up K8ssandra

```
helm repo add datastax https://datastax.github.io/charts  
helm repo update  
helm install k8ssandra datastax/k8ssandra-cluster
```

Setting up K8ssandra

That's it!

Setting up K8ssandra

No, for real. That's it! We now have setup:

Cassandra**Reaper**

Cassandra-Medusa



K8ssandra : Features to Name a Few

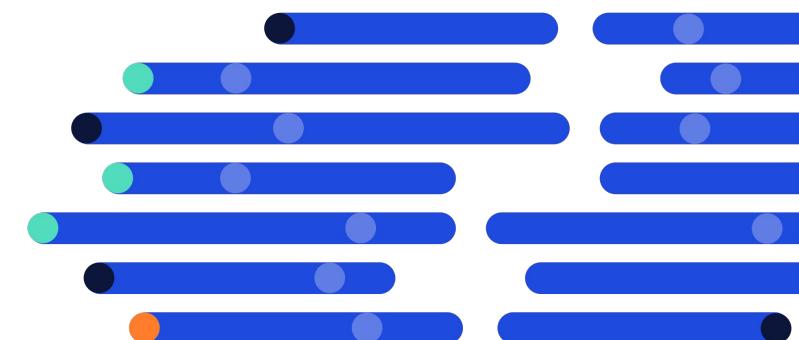
- Proper token **ring initialization**, with only one node bootstrapping at a time
- **Seed node management** -
 - one per rack, or three per datacenter, whichever is more
- Server configuration integrated into the **CassandraDatacenter CRD**
 - Rolling reboot nodes by changing the CRD
 - Store data 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)
- Automated repairs
- Scaling up and down simply
- Built in monitoring and dashboards

Online Workshops



Section #1 : Setting up and Monitoring Cassandra

<https://github.com/DataStax-Academy/kubecon2020>



menti.com



Kubernetes Quiz Time! - 6 questions



Available on the iPhone
App Store

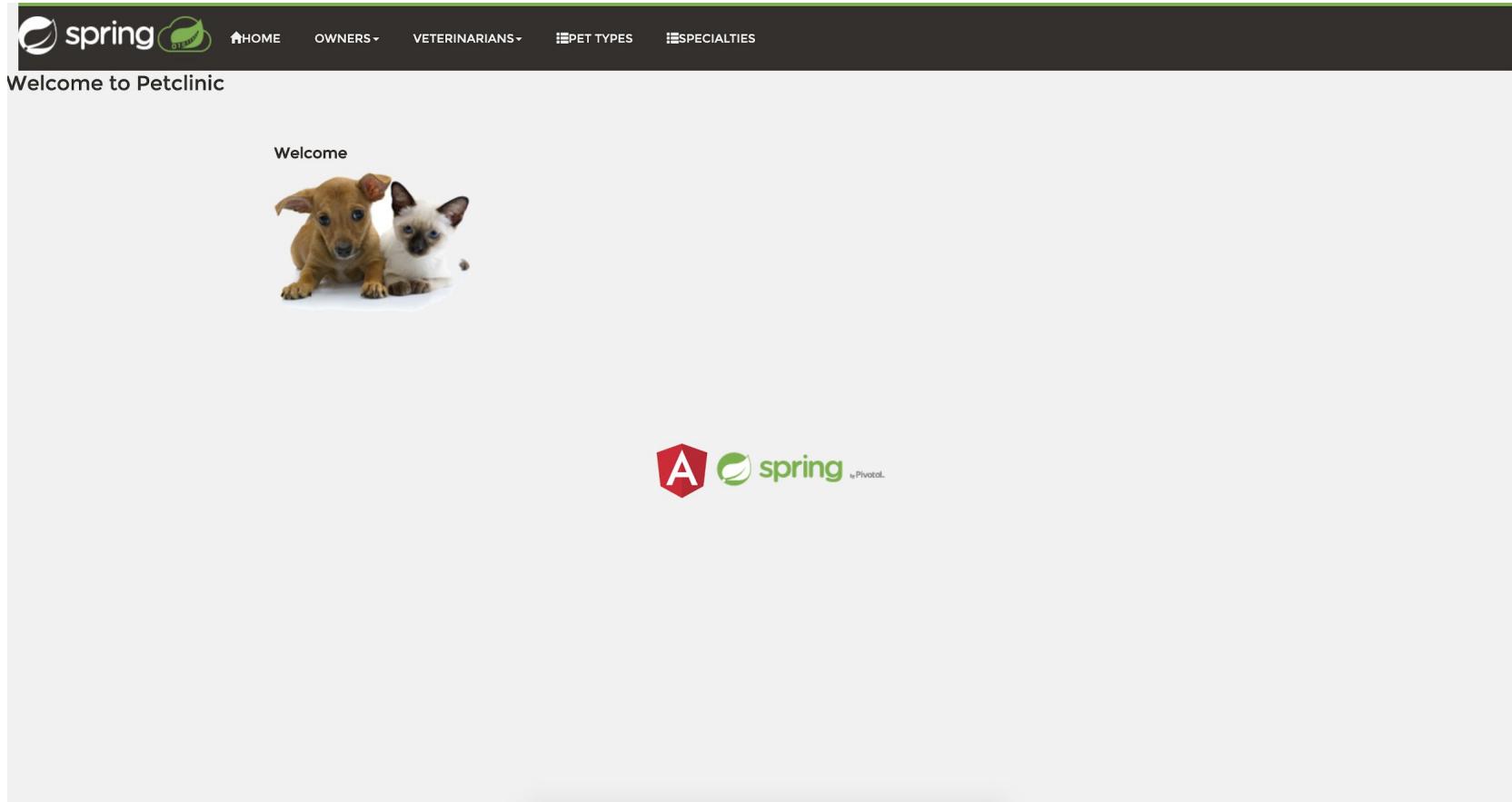
GET IT ON
Google play

Apache Cassandra™ with Kubernetes

- 1 Housekeeping and Quiz
- 2 Cassandra Basics
- 3 Setting Up and Monitoring Cassandra
- 4 Working with Data
- 5 Scaling Up and Down
- 6 Running Repairs
- 7 Backing Up and Restoring Data
- 8 Resources

Setting up K8ssandra

Pet Clinic App



Setting up K8ssandra

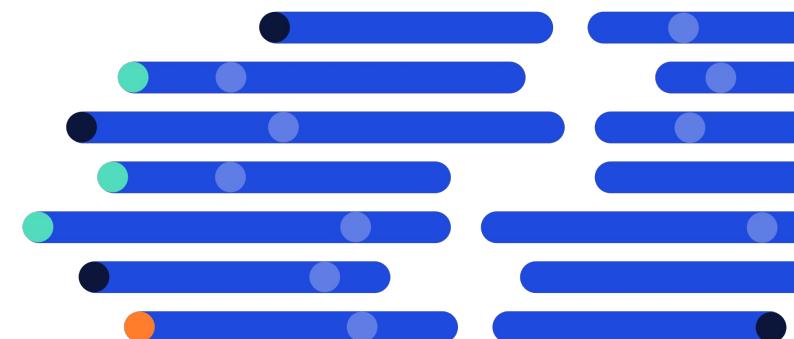
Let's Get To It!

Online Workshops



Section #2 : Working With Data

<https://github.com/DataStax-Academy/kubecon2020#2-working-with-data>



Apache Cassandra™ with Kubernetes

- 1 Housekeeping and Quiz
- 2 Cassandra Basics
- 3 Setting Up and Monitoring Cassandra
- 4 Working with Data
- 5 Scaling Up and Down
- 6 Running Repairs
- 7 Backing Up and Restoring Data
- 8 Resources

Scaling K8ssandra

There are a few ways we can scale our cluster. We will use one of them.

Scaling K8ssandra

1. Config files

```
# 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

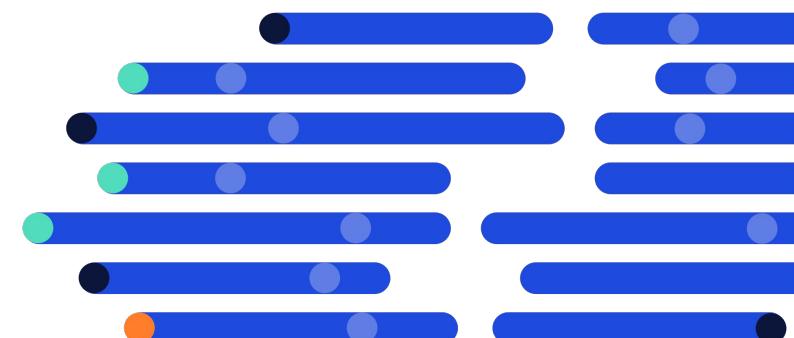
```
helm upgrade k8ssandra-cluster-a k8ssandra/k8ssandra-cluster --set size=2
```

Online Workshops



Section #3 : Scaling Up and Down

<https://github.com/DataStax-Academy/kubecon2020#3-scaling-up-and-down>



Apache Cassandra™ with Kubernetes

- 1 Housekeeping and Quiz
- 2 Cassandra Basics
- 3 Setting Up and Monitoring Cassandra
- 4 Working with Data
- 5 Scaling Up and Down
- 6 Running Repairs
- 7 Backing Up and Restoring Data
- 8 Resources

Running Repairs

Repairs are a way to keep data across a distributed system in sync.

TODO get image of repair process

Running Repairs

The screenshot shows the CassandraReaper 2.0.5 web application interface. The left sidebar contains navigation links: Clusters, Schedules, Repairs (selected), Snapshots, Live Diagnostic (beta), and Logout. The main area is titled "Cluster" and shows a single cluster named "k8ssandra". The cluster summary indicates "Nodes: 1", "Total load: 157.4 kB", and "Running repairs: 0". Below this, there are "Forget cluster" and "Info" buttons. The cluster details section shows a single node in "dc1" with a "default" keyspace containing 157.4 kB of data. The top right of the interface includes "Switch theme" and "Report a bug" links.

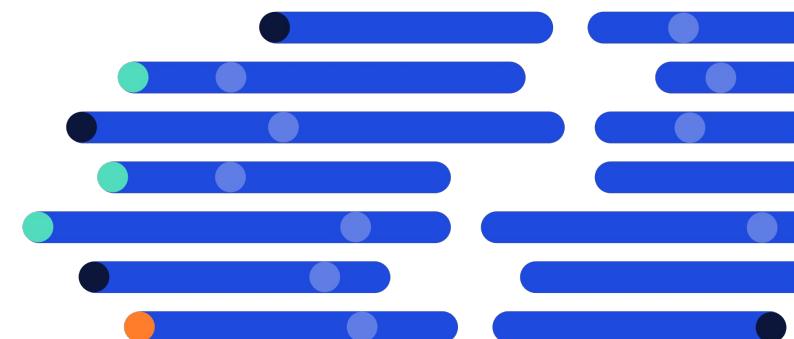
Online Workshops



Section #4 :

Running Repairs

<https://github.com/DataStax-Academy/kubecon2020#4-running-repair>



Apache Cassandra™ with Kubernetes

- 1 Housekeeping and Quiz
- 2 Cassandra Basics
- 3 Setting Up and Monitoring Cassandra
- 4 Working with Data
- 5 Scaling Up and Down
- 6 Running Repairs
- 7 Backing Up and Restoring Data
- 8 Resources

Backup and Restore

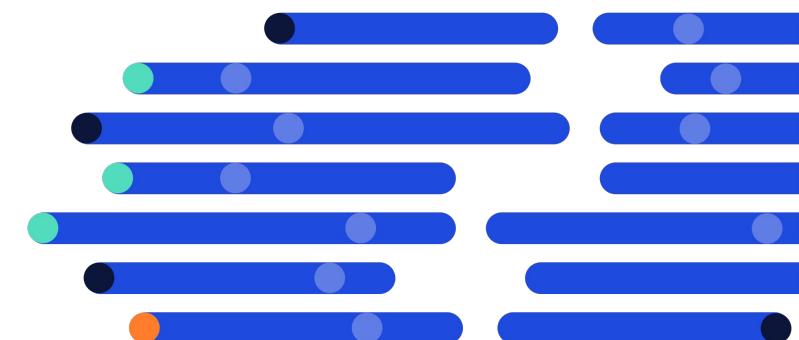
TODO Placeholder slide for showing backups

Online Workshops



Section #5 : Backing Up and Restoring Data

<https://github.com/DataStax-Academy/kubecon2020#5-backing-up-and-restoring-data>



Apache Cassandra™ with Kubernetes

- 1 Housekeeping and Quiz
- 2 Cassandra Basics
- 3 Setting Up and Monitoring Cassandra
- 4 Working with Data
- 5 Scaling Up and Down
- 6 Running Repairs
- 7 Backing Up and Restoring Data
- 8 Resources

The Project
k8ssandra.io/preview

Project GitHub
github.com/k8ssandra/k8ssandra

K8ssandra Roadmap

<https://k8ssandra.io/docs/roadmap/>

- Preconfigured alerts for metrics
 - Support sending to a configured single email address
 - Support sending to multiple addresses
 - Support sending to multiple addresses based on fired alert
- Annotations in metrics system signaling operations performed on the cluster
 - Restarts
 - Upgrades
 - Backups
 - Node lifecycle calls
- Extended authentication options for metrics system
- Network policies to isolate all components as appropriate
- Support for monitoring repair process via metrics system
- Support for monitoring backups process via metrics system
- Centralized logging
 - ELK
 - Loki
- Distributed tracing via Jaeger
- Load/stress/perf testing with nosqlbench - guidance on how to use this tool:
 - To find optimal load testing values (number of threads / inflight)
 - To determine node requirements
- Documentation enhancements
 - Istio best practices guide
 - Linkerd best practices guide
- Migrations
 - Connecting to existing on-prem datacenters / clusters
- Spark connection
- Kafka connection
- Data loading
- Service Broker
- Serverless / Faas
 - Kubeless
 - Knative
- Dynamics secrets with Vault
 - Roles via Cassandra plugin
 - Rotating TLS certificates for clients, nodes, ingress, etc.

Developer Resources

LEARN

Free learning resources at datastax.com/dev

DataStax Academy - datastax.com/dev/academy

Cassandra certifications - datastax.com/dev/certifications

ASK/SHARE

Join community.datastax.com

Ask/answer community user questions - share your expertise

SUBSCRIBE !

Follow us @DataStaxDevs

We are on Twitter - Youtube - Twitch!

Slides and code for this course are available at

<https://github.com/DataStax-Academy/kubecon-cassandra-workshop>

REVIEW

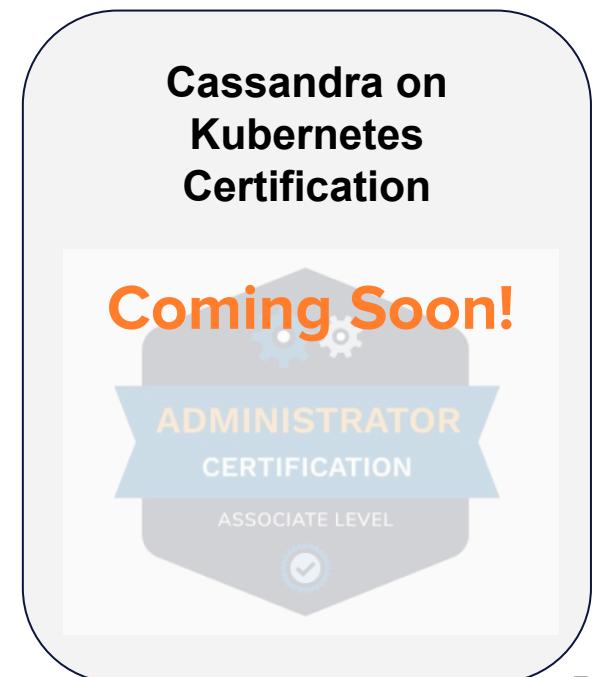
Accelerate

<https://www.datastax.com/accelerate/why-tomorrows-cassandra-deployments-will-be-on-kubernetes>



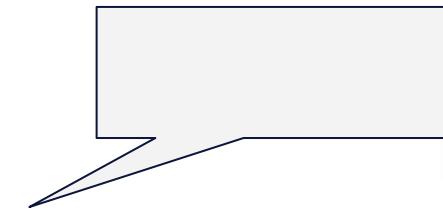
Cassandra on Kubernetes Certification!

- Interested in diving deeper with Cassandra, Kubernetes, K8ssandra?
- We're creating a new certification program for running Cassandra on Kubernetes
- Sign up for news and updates at <https://datastax.com/dev/certifications>



Free Learning Resources at datastax.com/dev

- Hands-on learning content
 - datastax.com/dev/kubernetes
 - datastax.com/cloud-native
- Self-paced courses at datastax.com/dev/academy
 - DS201: Foundations of Apache Cassandra™
 - DS210: Operations with Apache Cassandra™
 - DS220: Practical Application Data Modeling with Apache Cassandra™
 - **And MORE!**



Cassandra The Definitive Guide

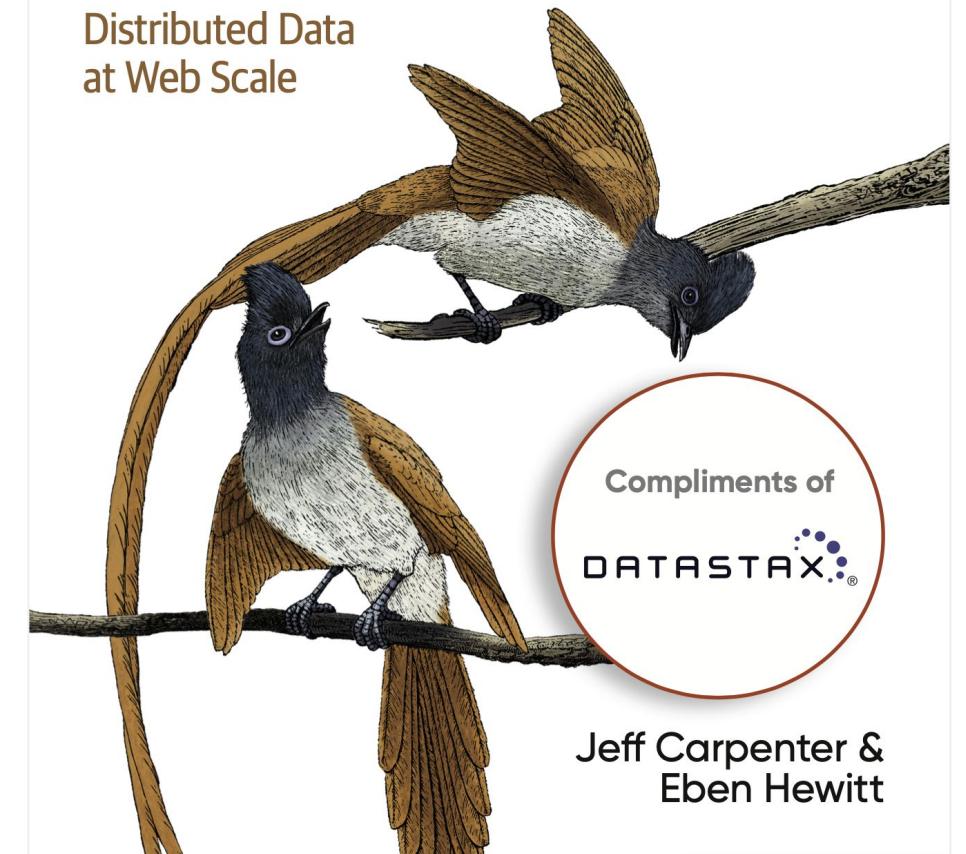
<https://www.datastax.com/resources/ebook/oreilly-cassandra-definitive-guide>

O'REILLY®

Cassandra The Definitive Guide

Distributed Data
at Web Scale

Third
Edition

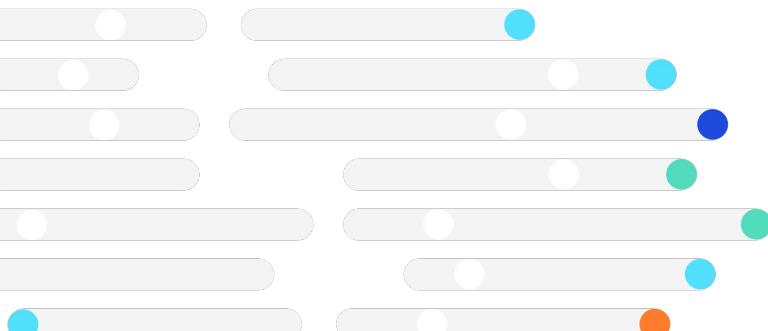


menti.com



Available on the iPhone
App Store

GET IT ON
Google play



Thank you

SURVEY