

# Developers

## App Development with Apache Cassandra™

Bootcamp Week 3: Full Stack Back End App Development with Apache Cassandra



# Theme Colors



**DS Black**  
Text and  
background 1



**White**  
Text and  
background 2



**Body Text**  
Text and  
background 3



**Light Gray**  
Text and  
background 4



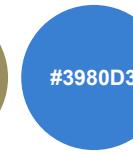
**Astra Red**  
Accent 1



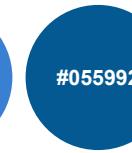
**Enterprise**  
Blue  
Accent 2



**Luna Gold**  
Accent 3



**DS Blue**  
Accent 4



**Blue 1**  
Accent 5



**Blue 2**  
Accent 6



**Blue 3**  
Custom

---

## Dark Line

**DS Black #3A3A42**  
Text and background 1

---

## Light Line

**#E6E6E6**  
Custom

# Iconography

Click into icon group to **change accent color** (do not change gray)



Migration to Astra



Global scale



Cloud native



Microservices



Login screen



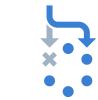
Orbit



Launch



Indexes



Active everywhere



APIs



Zero downtime



Scale



Integration



Performance



Developers



Developer



Developer persona



Architect persona



Operator persona



Relevance



Data-driven



Availability



Data



Security-1



Security-2



Security-3



IoT



Ecommerce



Agility



Analytics



OPs



Backup recovery



Logic



AI-ML-1



AI-ML-2



Migrate-1



Migrate-2



Users/  
customers



Leader



Cluster



Cedrick  
Lunven

Aleksandr  
Volochnev

Jack  
Fryer

Kirsten  
Hunter

Stefano  
Lottini

David  
Gilardi

Ryan  
Welford

Rags  
Srinivas

Sonia  
Siganporia

R

S



DataStax Developers Crew

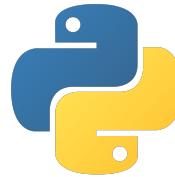
# Director of Developer Relations



- Trainer
- Public Speaker
- Developers Support
- Developer Applications
- Developer Tooling
  
- Creator of ff4j ([ff4j.org](http://ff4j.org))
- Maintainer for 8 years+
  
- Happy developer for 14 years
- Spring Petclinic Reactive & Starters
- Implementing APIs for 8 years



# Developer Advocate



- Developer/Architect
- Apache Cassandra™ certified
- Background in computational physics
- Distributed systems
- Love to teach and communicate

[dtsx.io/stefano](http://dtsx.io/stefano)

**Stefano Lottini**



@stefano-lottini



@hemidactylus



@rsprrs



# Senior Developer Advocate



STARGATE



- Training
- Teaching
- Support
- Public Speaking
- Reference Applications
- Cassandra Certified
- \*Cough (Oracle 8 certified)
- Happy developer for 25 years
- Coded in a ton, but Java/js mainstays
- Advanced User



# Developer Advocate



STARGATE



{ REST }



ryanwelford



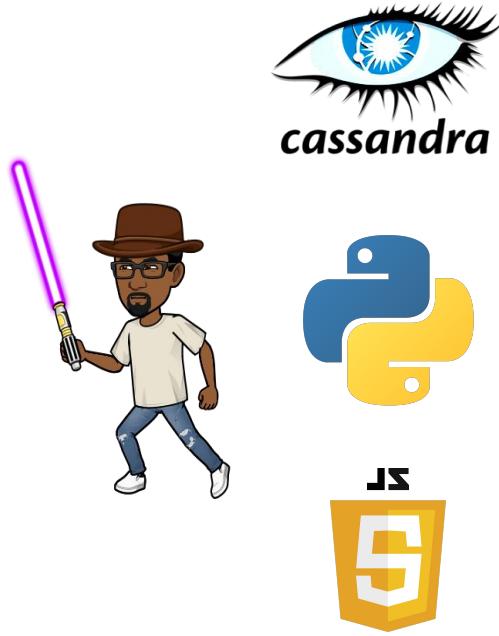
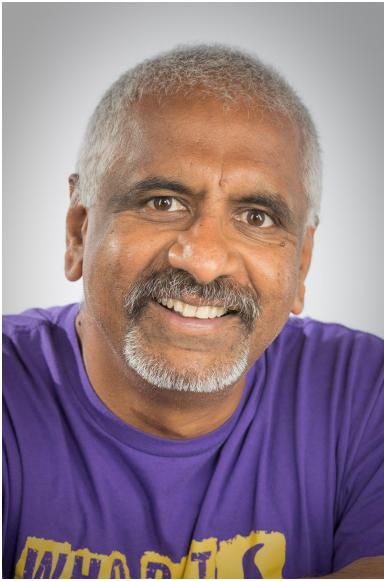
RyanWelford

- Front-End dev
- Training
- Teaching
- Support
  
- Happy developer for 5 years
- Love developer awesome user experiences



Ryan Welford

# Developer Advocate



- Developer/Architect
- Mechanical Engineer (so many moons ago)
- Distributed systems
- Love to teach and communicate
- Inner loop == developer productivity



Raghavan "Rags" Srinivas



@rags



@ragsns



@ragss

# 01



Week 1 reminders  
**HouseKeeping**

# 02

**title**  
**Power of Purpose-built DB**

# 04

**CAP Theorem**  
**Law of Distributed Systems**

# 03

**Intro to Cassandra**  
**Peer-to-Peer Database**

# 05

**Tables and Partitions**  
**How do you structure data**

# 06

**What's next?**  
**Quiz, Homework, Next week**



**Agenda**



## WEEK 1

January 5th - January 11th



## WEEK 2

January 12th - January 18th

## WEEK 3

January 19th-January 25th



## WEEK 4

January 26th-February 1st



Introducing 2022 Bootcamp

## #1 Introduction to NoSQL Databases and Apache Cassandra™

- ~2H - Live lessons (same but different timezones)
  - January 5th 6 PM CET (= 9am PST = 11 am CST = 12 pm EST)
  - January 6th 9 AM CET (= 12am PST = 1am CST = 3am EST)
- ~3H - Homeworks
  - Hands-on with DBaaS Cassandra service AstraDB
  - Learning materials: Cassandra Fundamentals exercises 1 to 8
  - Fill the week 1 test



**WEEK 2**  
January 12h - January 18th

**WEEK 3**  
January 19th-January 25th



**WEEK 4**  
January 26th-February 1st



**Introducing 2022 Bootcamp**

## #1 Introduction to NoSQL Databases and Apache Cassandra™

- ~2H - Live lessons (same but different timezones)
  - January 5th 6 PM CET (= 9am PST = 11 am CST = 12 pm EST)
  - January 6th 9 AM CET (= 12am PST = 1am CST = 3am EST)
- ~3H - Homeworks
  - Hands-on with DBaaS Cassandra service AstraDB
  - Learning materials: Cassandra Fundamentals exercises 1 to 8
  - Fill the week 1 test



# WEEK 3

January 19th-January 25th

## #2 Design Cassandra Data Models for Full-Stack Applications

- ~2H - Live lessons (twice the same for different timezones)
  - January 12th 6 PM CET (= 9am PST = 11 am CST = 12 pm EST)
  - January 13th 9 AM CET (= 12am PST = 1am CST = 3am EST)
- ~3H - Homeworks
  - Hands-on with DBaaS Cassandra service AstraDB
  - Learning materials: Cassandra Fundamentals exercises 9 to 14
  - Learning materials: DS220 Modules 1 to 20 (Table Features)
  - Fill the week 2 test



# WEEK 4

January 26th-February 1st



Introducing 2022 Bootcamp

## #1 Introduction to NoSQL Databases and Apache Cassandra™

- ~2H - Live lessons (same but different timezones)
  - January 5th 6 PM CET (= 9am PST = 11 am CST = 12 pm EST)
  - January 6th 9 AM CET (= 12am PST = 1am CST = 3am EST)
- ~3H - Homeworks
  - Hands-on with DBaaS Cassandra service AstraDB
  - Learning materials: Cassandra Fundamentals exercises 1 to 8
  - Fill the week 1 test

## #2 Design Cassandra Data Models for Full-Stack Applications

- ~2H - Live lessons (twice the same for different timezones)
  - January 12th 6 PM CET (= 9am PST = 11 am CST = 12 pm EST)
  - January 13th 9 AM CET (= 12am PST = 1am CST = 3am EST)
- ~3H - Homeworks
  - Hands-on with DBaaS Cassandra service AstraDB
  - Learning materials: Cassandra Fundamentals exercises 9 to 14
  - Learning materials: DS220 Modules 1 to 20 (Table Features)
  - Fill the week 2 test

## #3 Build a Full-Stack Backend with a NoSQL database

- ~2H - Live lessons (same but different timezones)
  - January 19th 6 PM CET (= 9am PST = 11 am CST = 12 pm EST)
  - January 20th 9 AM CET (= 12am PST = 1am CST = 3am EST)
- ~3H - Homeworks
  - Coding with DBaaS Cassandra service AstraDB
  - DS220 Modules 21 to 40 (Collections-> Physical Data Model)
  - Fill the week 3 test

**WEEK 4**  
January 26th-February 1st



## #1 Introduction to NoSQL Databases and Apache Cassandra™

- ~2H - Live lessons (same but different timezones)
  - January 5th 6 PM CET (= 9am PST = 11 am CST = 12 pm EST)
  - January 6th 9 AM CET (= 12am PST = 1am CST = 3am EST)
- ~3H - Homeworks
  - Hands-on with DBaaS Cassandra service AstraDB
  - Learning materials: Cassandra Fundamentals exercises 1 to 8
  - Fill the week 1 test

## #2 Design Cassandra Data Models for Full-Stack Applications

- ~2H - Live lessons (twice the same for different timezones)
  - January 12th 6 PM CET (= 9am PST = 11 am CST = 12 pm EST)
  - January 13th 9 AM CET (= 12am PST = 1am CST = 3am EST)
- ~3H - Homeworks
  - Hands-on with DBaaS Cassandra service AstraDB
  - Learning materials: Cassandra Fundamentals exercises 9 to 14
  - Learning materials: DS220 Modules 1 to 20 (Table Features)
  - Fill the week 2 test

## #3 Build a Full-Stack Backend with a NoSQL database

- ~2H - Live lessons (same but different timezones)
  - January 19th 6 PM CET (= 9am PST = 11 am CST = 12 pm EST)
  - January 20th 9 AM CET (= 12am PST = 1am CST = 3am EST)
- ~3H - Homeworks
  - Coding with DBaaS Cassandra service AstraDB
  - DS220 Modules 21 to 40 (Collections-> Physical Data Model)
  - Fill the week 3 test

## #4 Build Api and Microservices for Apache Cassandra™

- ~2H - Live lessons (same but different timezones)
  - January 26th 6 PM CET (= 9am PST = 11 am CST = 12 pm EST)
  - January 27th 9 AM CET (= 12am PST = 1am CST = 3am EST)
- ~3H - Homeworks
  - Coding with DBaaS Cassandra service AstraDB
  - Learning materials: DS220 Modules 42 to 56
  - Fill the week 4 test



Introducing 2022 Bootcamp

1

Attend one of the 2 LIVE STREAMED workshop  
(Wednesday or Thursday)



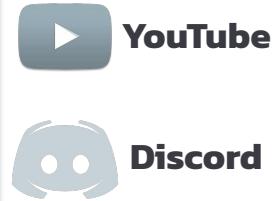
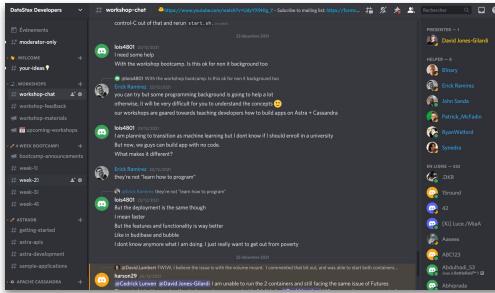
4-week bootcamp Housekeeping



**Livestream:** youtube.com/DataStaxDevs

**Questions:** <https://dtsx.io/discord>

#### Agenda



## Games and quizzes: [menti.com](https://menti.com)

How much experience do you have with the Spring Framework ?



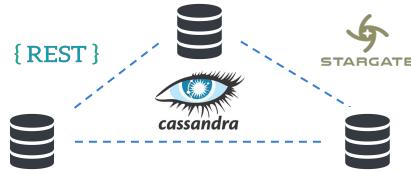
- 1 Attend one of the 2 LIVE STREAMED workshop  
(Wednesday or Thursday)
- 2 Complete the workshop labs



4-week bootcamp Housekeeping



## Database + GraphQL + PlayGround



DataStax  
**Astra DB**



The screenshot shows the Gitpod IDE interface. On the left is the Explorer view displaying project files like "StargateDemoApplication.java", "ManifestDemoApplication.java", and "gitpod.Dockerfile". The center is the Editor view with code for "StargateDemoApplication.java". Below the editor are tabs for PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE. To the right are icons for npm, node.js, Maven, and Java. The bottom features the Gitpod logo.

The screenshot shows a GitHub repository page for "DataStax-Examples / todo-astra-jamstack-netlify". It displays a pull request, a commit history, and sections for Readme, Releases, and a sidebar for "astra.datastax.com/register". The GitHub logo is centered below the repository view.

O'REILLY  
KATACODA KATACODA OVERVIEW & SOLUTIONS

Connect to Astra (Cassandra as a Service) with CQL Shell

Step 1 of 5

### Create your Astra DB Database

If you don't have an Astra account, set one up - it's easy!

Go to the Astra DB page in your browser [astra.datastax.com](https://astra.datastax.com).

Let's create the database. Follow the steps outlined here. To make it easy, we have recommended the values you should use for this scenario.

NOTE: If you already have an Astra DB database with values that differ from what we suggested, you may have to adapt some of the operations in this scenario accordingly, or create an additional response with the designated values.

If you don't already have an Astra DB database, when you log in for the first time you'll see a screen that looks like the image below.

Choose Plan & Provider



The screenshot shows the Katacoda interface. At the top is a navigation bar with "TRY O'REILLY", "KATACODA", "TERMINAL", and "LOGOUT". Below is a main content area with a heading "Create your Astra DB Database" and a "Connect to Astra (Cassandra as a Service) with CQL Shell" button. To the right is a "Terminal" window showing a command-line interface. A "Help" section titled "KATACODA OVERVIEW & SOLUTIONS" provides instructions for creating a database.

2

## Complete Workshops Labs



- 1** Attend one of the 2 LIVE STREAMED workshop  
(Wednesday or Thursday)
- 2** Complete the workshop labs
- 3** Complete the Learning materials



4-week bootcamp Housekeeping

The screenshot shows the DataStax Academy course page for DS220. At the top, it says "Course Content" and "Introduction". Below that, there are three sections: "Data Modeling Overview", "Data Modeling Overview Quiz", and "Relational Vs. Apache Cassandra". Each section has a "Start" button. To the right, there's a progress bar showing "Not Started 0/56" and a circular progress indicator at 0%. Below the progress bar are sections for "Badges" and "Competencies".

The screenshot shows the DataStax /dev website's "Learning Series Topics" page for "Cassandra Fundamentals". It features a blue header with the title "Cassandra Fundamentals" and a "GET STARTED" button. Below the header, there's a decorative background with a hexagonal grid pattern. On the left, there's a sidebar with "Learning Series Topics" and a duration of "APPROXIMATELY 4H". On the right, there's a list of topics numbered 01 to 11, each with a small icon and a brief description.

The screenshot shows a GitHub repository named "DataStax-Examples / todo-astra-jamstack-netlify". The repository has 177 commits ahead of "tjake/master". It includes branches like "master" and "astra", and files like "functions", "public", and "src". A green "Group" button is visible. The repository has 230 commits and 20 stars. It also includes sections for "astra datastax.com/register" and "Releases".



3

Do your homeworks

- 1** Attend one of the 2 LIVE STREAMED workshop (Wednesday or Thursday)
- 2** Complete the workshop labs
- 3** Complete the Learning materials
- 4** Pass the weekly test (google form)



4-week bootcamp Housekeeping



## Intro to NoSQL Homework

Welcome and thank you! Here you can submit your homework for the datastax developers "Intro to NoSQL" workshop. In case of any questions please contact organisers at <https://dtsx.io/aleks> or just send an email to [aleksandr.volochnev@datastax.com](mailto:aleksandr.volochnev@datastax.com)

- Workshop materials: <https://github.com/datastaxdevs/workshop-introduction-to-nosql>
- Discord chat: <https://dtsx.io/discord>

[cedrick.lunven@datastax.com](mailto:cedrick.lunven@datastax.com) [Switch account](#) 

The name and photo associated with your Google account will be recorded when you upload files and submit this form. Only the email you enter is part of your response.

\* Required

Email \*

Your email

4

Pass the Weekly Test

# menti.com



Go to [www.menti.com](http://www.menti.com) and use the code 3491 9972

## Inequality predicates are allowed on ...

A bar chart titled "Inequality predicates are allowed on ...". The y-axis represents the count of inequality predicates, ranging from 1 to 15. The x-axis categories are "All table columns", "Partition key columns", "clustering key columns", and "No inequality predicates are allowed".

Column Type	Count
All table columns	4
Partition key columns	3
clustering key columns	15
No inequality predicates are allowed	1

Below the chart, there is a video player interface showing a video of a person speaking. The video player includes controls like play/pause, volume, and a progress bar indicating 2:10:19 / 2:26:05. The title of the video is "Big paycheck".

Go to [www.menti.com](http://www.menti.com) and use the code 3491 9972

## Leaderboard

User ID	User Name	Profile Picture
4821 p	spanda	
4820 p	Agent X9	
4775 p	Sam	
4711 p	CCedrickThePresenter	
4468 p	shubham	
4371 p	aaa	
3895 p	vignesh	
3877 p	adry	
3861 p	Millie	
3812 p	Puggie	

Below the leaderboard, there is a video player interface showing a video of a person speaking. The video player includes controls like play/pause, volume, and a progress bar indicating 2:11:07 / 2:26:05. The title of the video is "Big paycheck".

# 01



**Intro to Bootcamp 2022**  
**HouseKeeping**

# 02

**Intro to NoSQL Databases**  
**Power of Purpose-built DB**

# 03

**Intro to Cassandra**  
**Peer-to-Peer Database**

# 04

**CAP Theorem**  
**Law of Distributed Systems**

# 05

**Tables and Partitions**  
**How do you structure data**

# 06

**What's next?**  
**Quiz, Homework, Next week**



**Agenda**

# Why NoSQL was invented?

Relational Databases dominated the database market for decades. Why change?



**Relational Database Management Systems weren't  
ready to cope with the new data volume.**



**Relational Database Management  
Systems weren't ready to cope with the  
new performance requirements.**

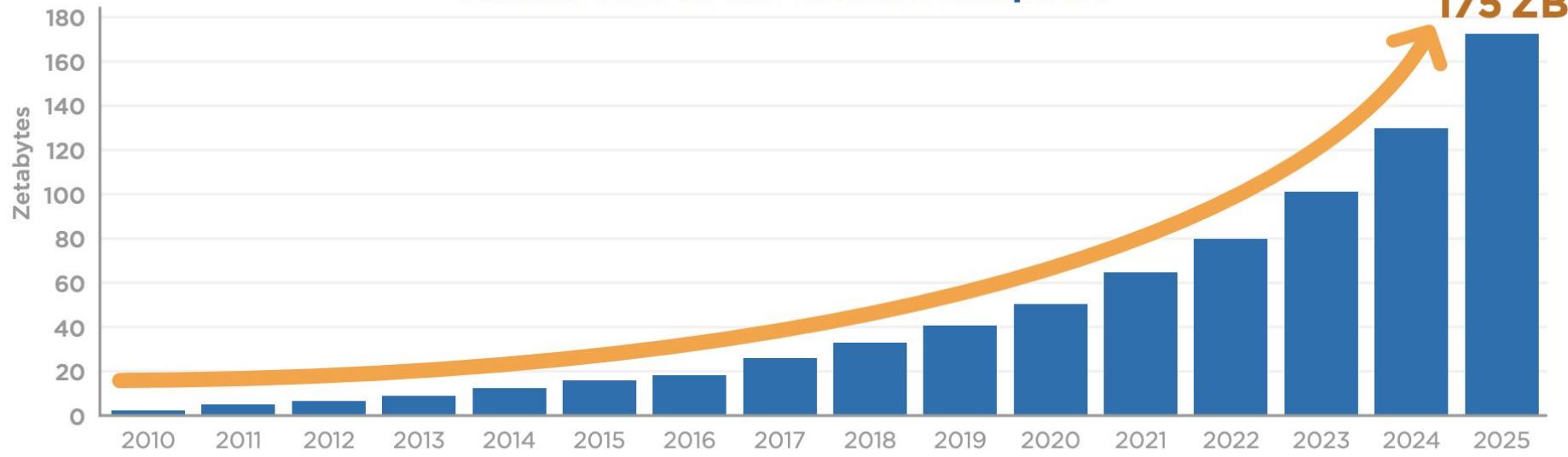


**Relational Database Management**  
**Systems weren't ready to cope with the**  
**requirements of data types and data**  
**relations.**

- Schemaless
- Relation as an entity
- Etc.



## Annual Size of the Global Datasphere



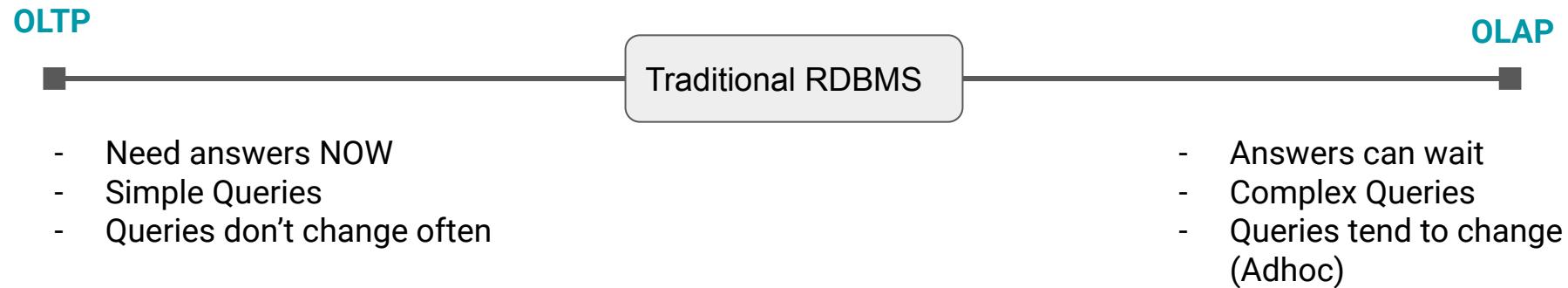
Source: Data Age 2025, sponsored by Seagate with data from IDC Global DataSphere, Nov 2018



Requirements changed



- OnLine Transaction Processing
- OnLine Analytical Processing



OLTP / OLAP



# Relational Databases



Any database that supports SQL and based on the relational model of data

# Not Only SQL

## Time-Series Databases

*Influx, OpenTSDB, Prometheus*

## Document Databases

*MongoDB, Elastic, DocDb*

## Tabular Databases

*Cassandra, Hbase, Bigtable*

## Ledger Databases

*Amazon QLDB, ksql\_db*

## Graph Databases

*Neo4j, Datastax Graph, Titan*

## Key/Value Databases

*Redis, Dynamo, Memcache*



SQL to NoSQL



# 01



**Intro to Bootcamp 2022  
HouseKeeping**

# 02

**Intro to NoSQL Databases  
Power of Purpose-built DB**

# 03

**Intro to Cassandra  
Peer-to-Peer Database**

# 04

**CAP Theorem  
Law of Distributed Systems**

# 05

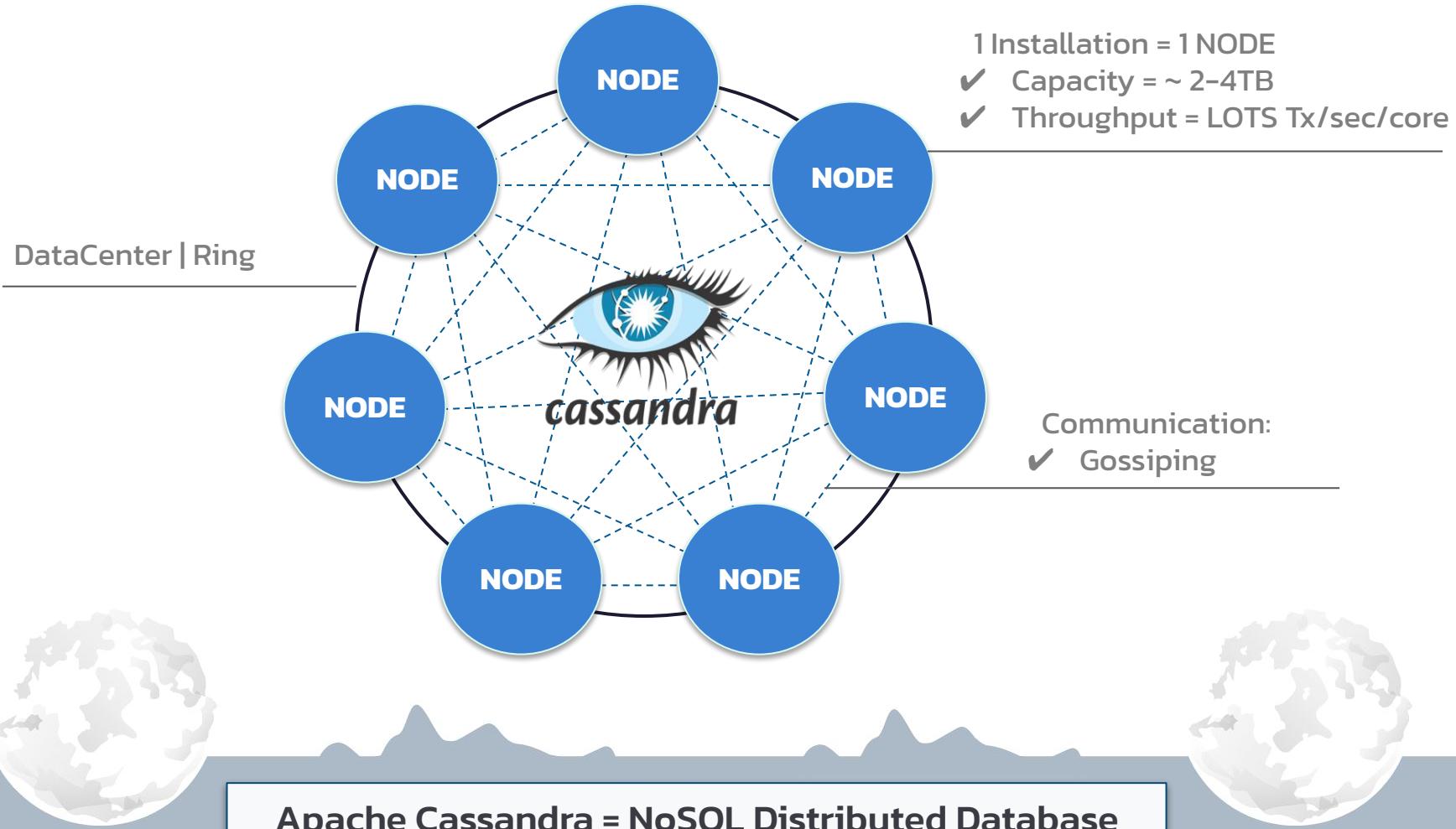
**Tables and Partitions  
How do you structure data**

# 06

**What's next?  
Quiz, Homework, Next week**



**Agenda**



## Apache Cassandra @ Netflix

- . 98% of streaming data is stored in Apache Cassandra
- . Data ranges from customer details to viewing history to billing and payments
- . Foundational datastore for serving millions of operations per second

- 30 million ops/sec on most active single cluster
- 500 TB most dense single cluster
- 9216 CPUs in biggest cluster

O(100) Clusters  
O(10000) Instances  
O(10,000,000) Replications per second  
O(100,000,000) Operations per second  
O(1,000,000,000,000,000) Petabytes of data

[dtsx.io/cassandra-at-netflix](http://dtsx.io/cassandra-at-netflix)

## Apple Scale

- 160K+ Apache Cassandra instances
- 100+ PB stored
- Several million ops / sec
- 1000s of clusters



And many others...



Cassandra Biggest Users (and Developers)

# Apache Cassandra

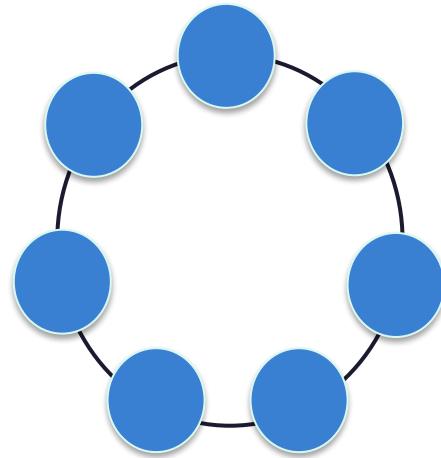
NoSQL Distributed Decentralised Database Management System



- Big Data Ready
- Read / Write Performance
- Linear Scalability
- Highest Availability
- Self-Healing and Automation
- Geographical Distribution
- Platform Agnostic
- Vendor Independent



**Partitioning** over distributed architecture makes the database capable to handle data of any size: we mean petabytes scale. Need more volume? Add more nodes.

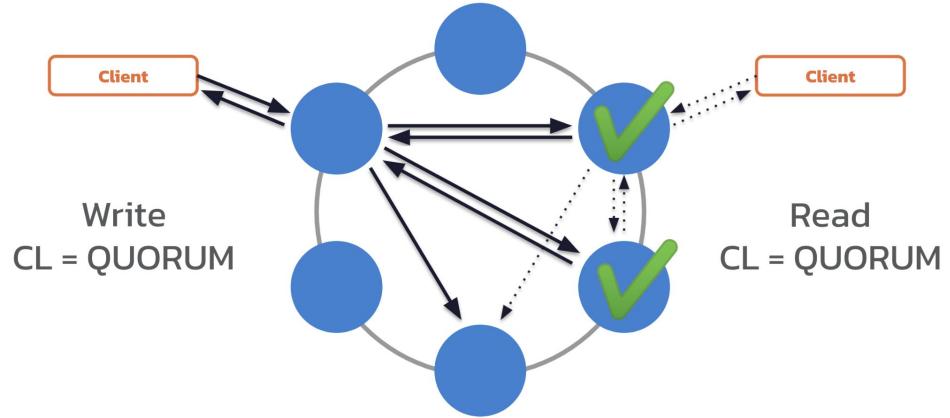


Big Data Ready



Even a single Cassandra node is very performant but a cluster consisting of multiple nodes and data centers brings throughput to the next level.

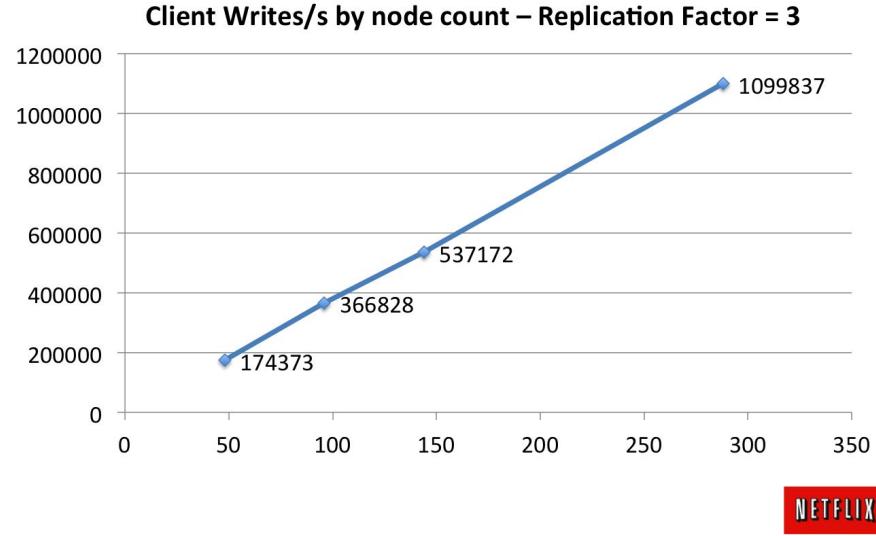
Decentralisation (**masterless architecture**) means that every node is able to deal with any request, read or write.



Read / Write Performance



- For volume or velocity, there are no limitations
- **Linear** - No overhead on new nodes, scales with your needs\*

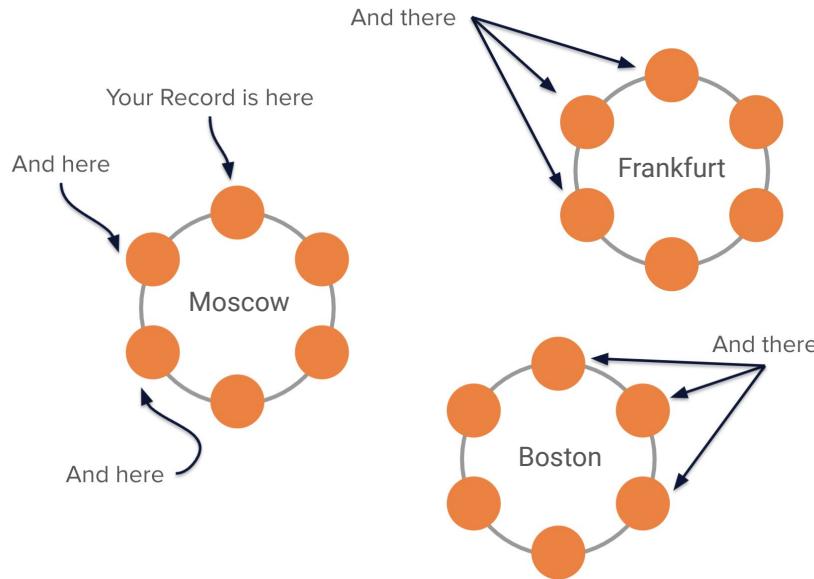


Linear Scalability



Replication, Decentralisation, and Topology-Aware Placement Strategy take care of possible downtimes:

- Multiple Live Replicas
- No Single Point of Failure
- Network topology-aware data placement
- Client-side Smart Reconnection and Strong Retry Mechanism



Highest Availability



Operations for a huge cluster can be exhausting so Apache Cassandra clusters are smart and able to scale, change data placement and recover automatically.

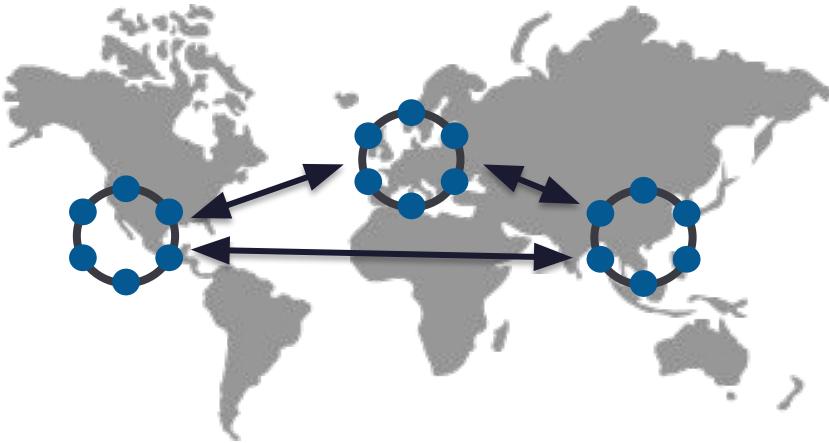


**Self-Healing and Automation**



Cassandra's trademark is multi-datacenter deployments, granting you an exceptional capability for disaster tolerance while keeping your data close to your clients - worldwide.

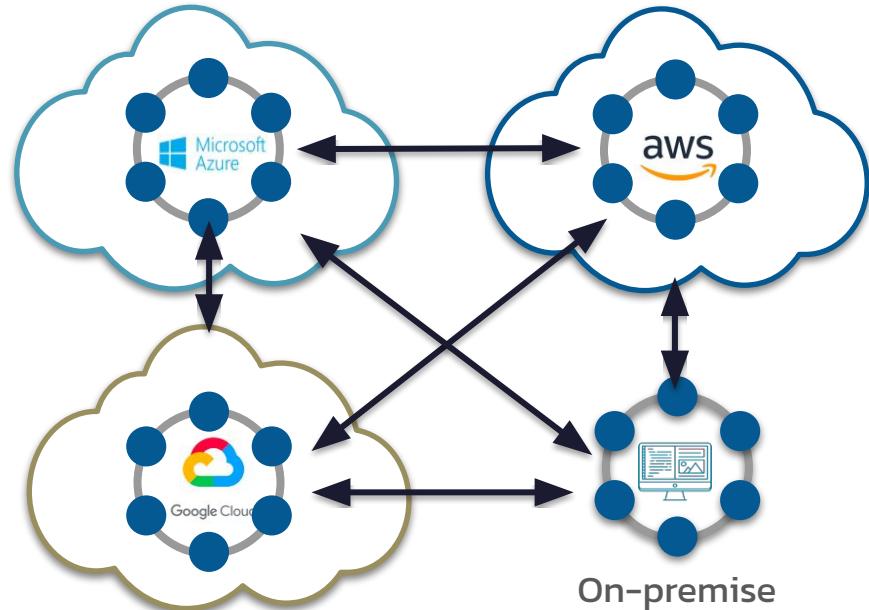
All DCs are active (available for both writes and reads)!



## Geographical Distribution



Apache Cassandra is **not bound to any platform** or service provider, helping you build hybrid-cloud and multi-cloud solutions with ease.



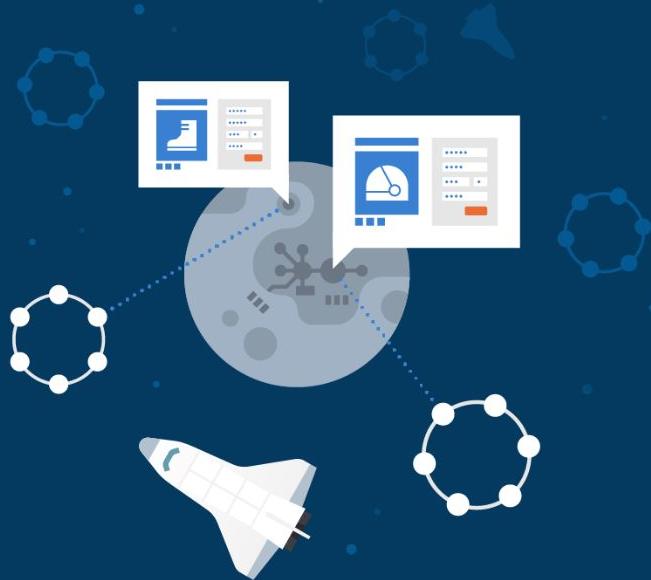
Platform Agnostic

Cassandra doesn't belong to any of commercial vendors but controlled by a non-profit Open Source **Apache Software Foundation**, already familiar to you by *Hadoop*, *Spark*, *Kafka*, *Zookeeper*, *Maven* and many other projects.



Vendor Independent





**Looks cool, right?**  
How does it all work?

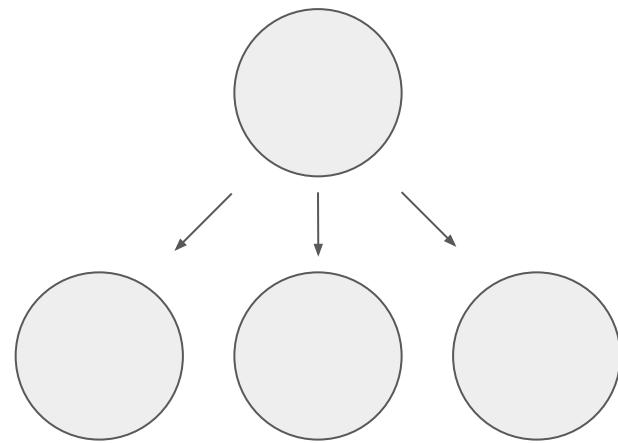


# All servers are created equal

*Cassandra Holy Book, P.I Paragraph I*



Leader Server (Write/Read)

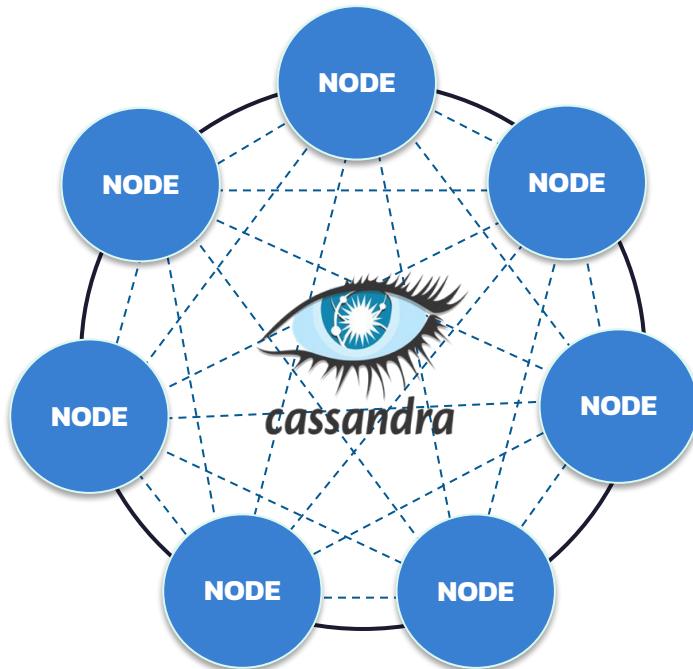


1. Single Point of Failure
2. Hard to scale for writes
3. Application needs to know where to write



Traditional Architecture





1. NO Single Point of Failure
2. Scales for writes and reads
3. Application can contact any node  
(in case of failure - just contact next one)



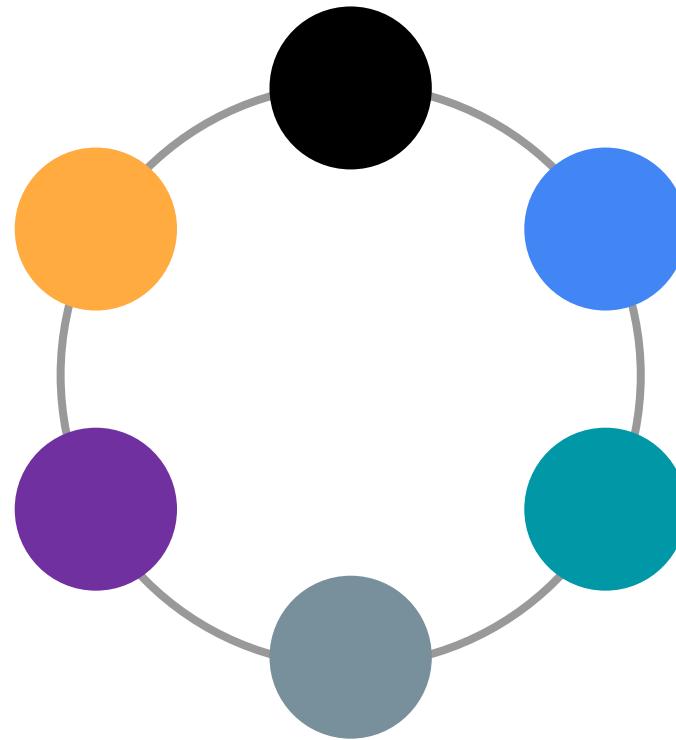
Master-less (Peer-to-Peer) Architecture

# Data is Replicated



$RF = ?$

Replication Factor  
means the number  
of nodes used to  
store each partition



**Replication Factor**

```
CREATE KEYSPACE population  
    WITH REPLICATION = {  
        'class' : 'NetworkTopologyStrategy',  
        'us-west-1' : 3,  
        'eu-east-2' : 5  
    };
```

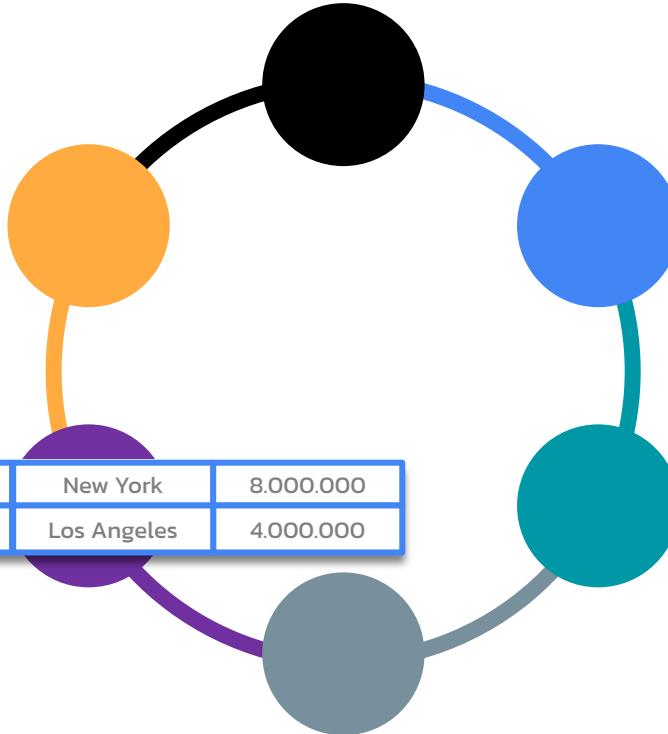
keyspace      replication strategy



**RF = 1**

Replication Factor 1  
means that every  
partition is stored  
on 1 node

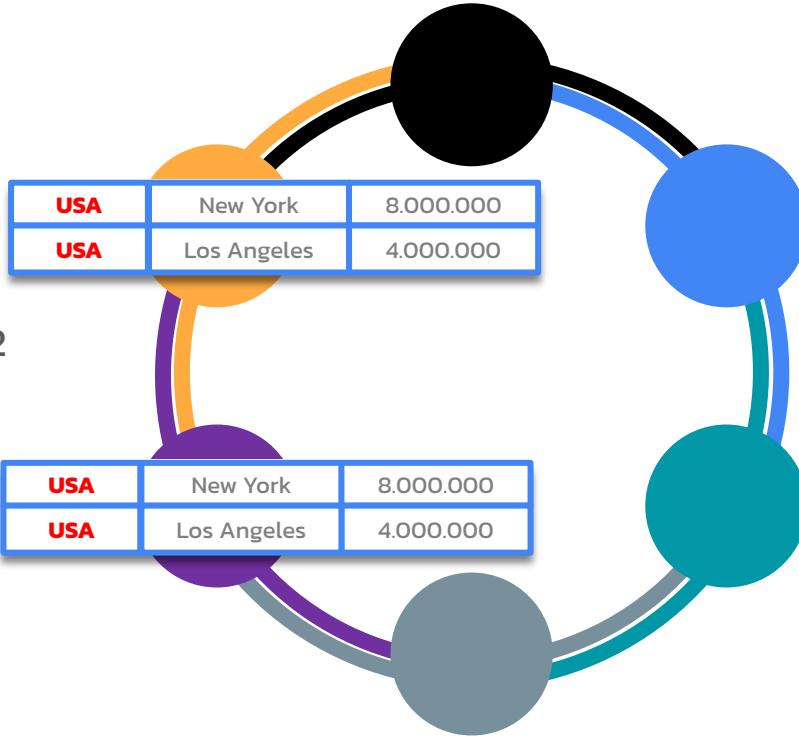
<b>USA</b>	New York	8.000.000
<b>USA</b>	Los Angeles	4.000.000



## Replication Factor

**RF = 2**

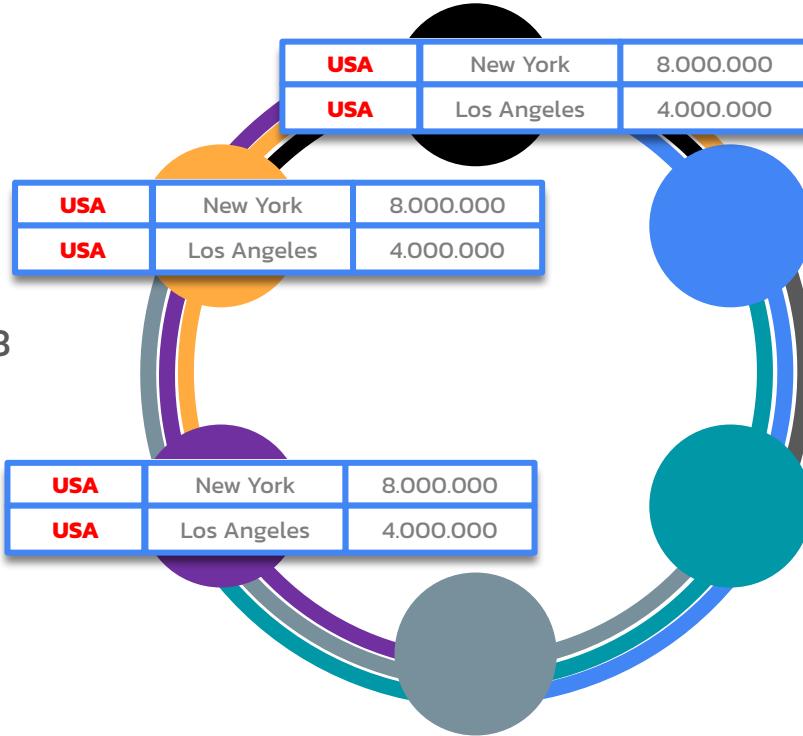
Replication Factor 2  
means that every  
partition is stored  
on 2 nodes



**Replication Factor**

**RF = 3**

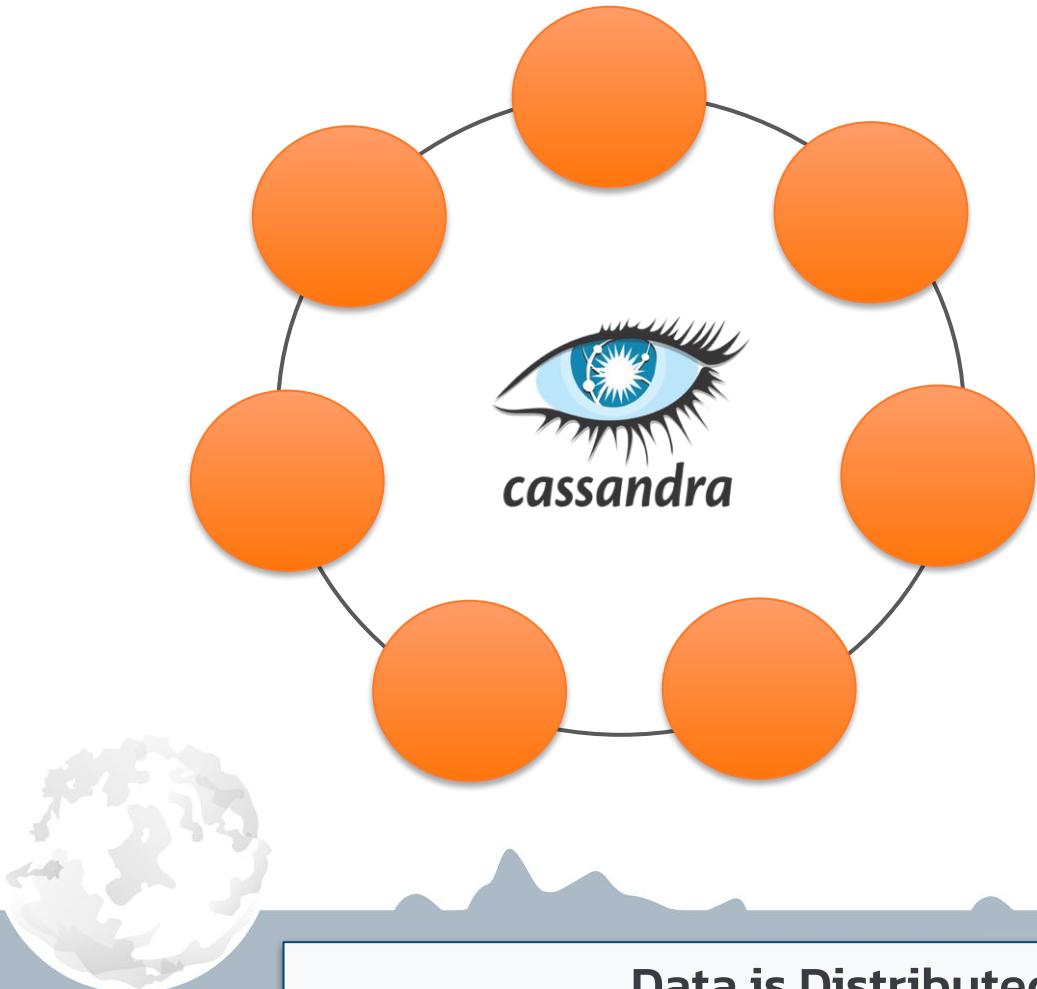
Replication Factor 3  
means that every  
partition is stored  
on 3 nodes



**Replication Factor**

# Data is Distributed





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

Partition Key

Data is Distributed

<b>USA</b>	New York	8.000.000
<b>USA</b>	Los Angeles	4.000.000

Country      City      Population

<b>DE</b>	Berlin	3.350.000
<b>DE</b>	Nuremberg	500.000

<b>FR</b>	Paris	2.230.000
<b>FR</b>	Toulouse	1.100.000

<b>UK</b>	London	9.200.000
-----------	--------	-----------

<b>JP</b>	Tokyo	37.430.000
-----------	-------	------------

<b>AU</b>	Sydney	4.900.000
<b>IN</b>	Mumbai	20.200.000

<b>CA</b>	Toronto	6.200.000
<b>CA</b>	Montreal	4.200.000



*cassandra*



Data is Distributed

```
Keyspace  
↓  
CREATE TABLE population.population_by_country_city (  
    country text,  
    city text,  
    population integer,  
    PRIMARY KEY ((country), city)  
);  
Table  
↓  
↑  
Partition key
```

Key-based Partitioning

Country	City	Population
USA	New York	8.000.000
USA	Los Angeles	4.000.000
CA	Toronto	6.200.000
CA	Montreal	4.200.000
DE	Berlin	3.350.000
DE	Nuremberg	500.000

Partition Keys

Partitioner  
Murmur3 Hashing

Country	City	Population
59	New York	8.000.000
59	Los Angeles	4.000.000
12	Toronto	6.200.000
12	Montreal	4.200.000
45	Berlin	3.350.000
45	Nuremberg	500.000

Tokens

Cassandra Nodes

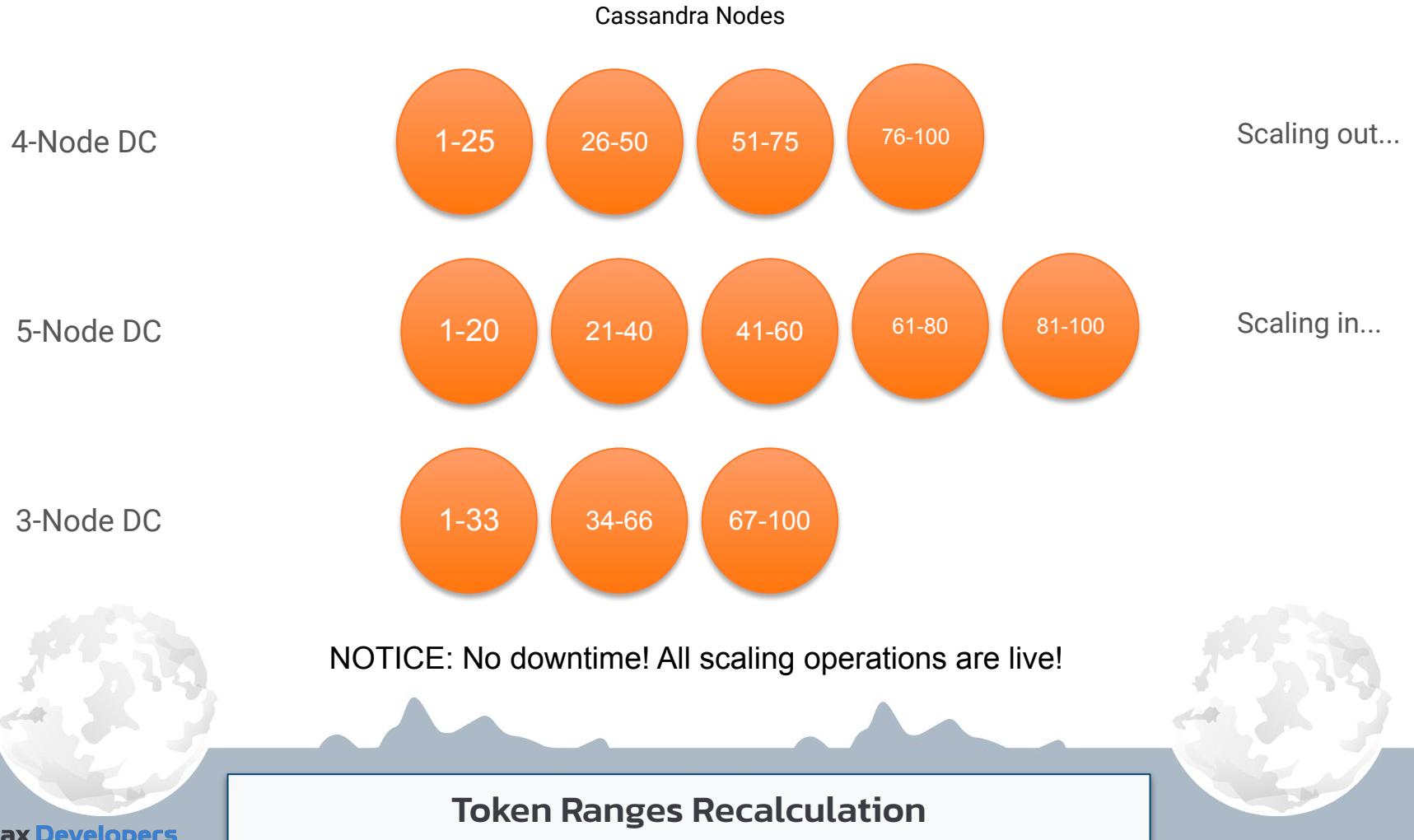


Partitioning and Token Ranges

Why partitioning?  
Because scaling doesn't have to be [s]hard!

Big Data doesn't fit to a single server, splitting it into chunks we can easily spread them over dozens, hundreds or even thousands of servers, adding more if needed.





Replication Factor = 1

1-25

26-50

51-75

76-100

Replication Factor = 2

1-25,  
26-50

26-50,  
51-75

51-75,  
76-100

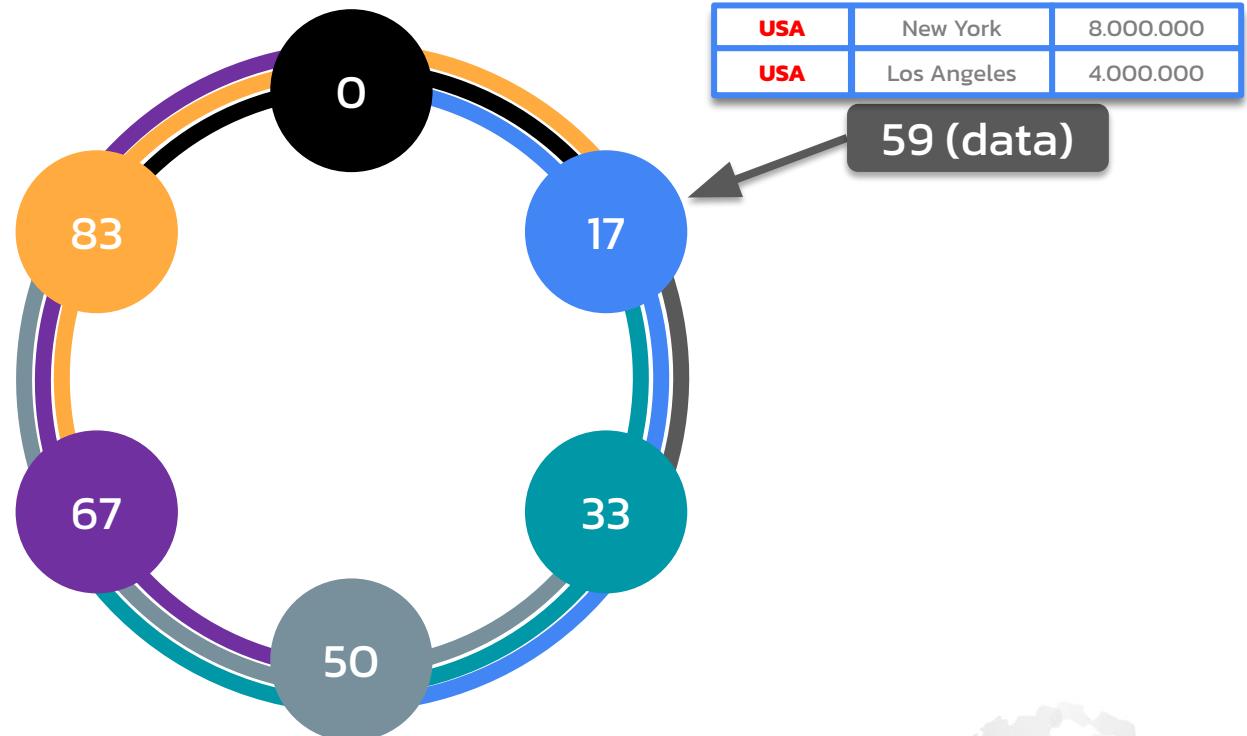
76-100,  
1-25

And so on...



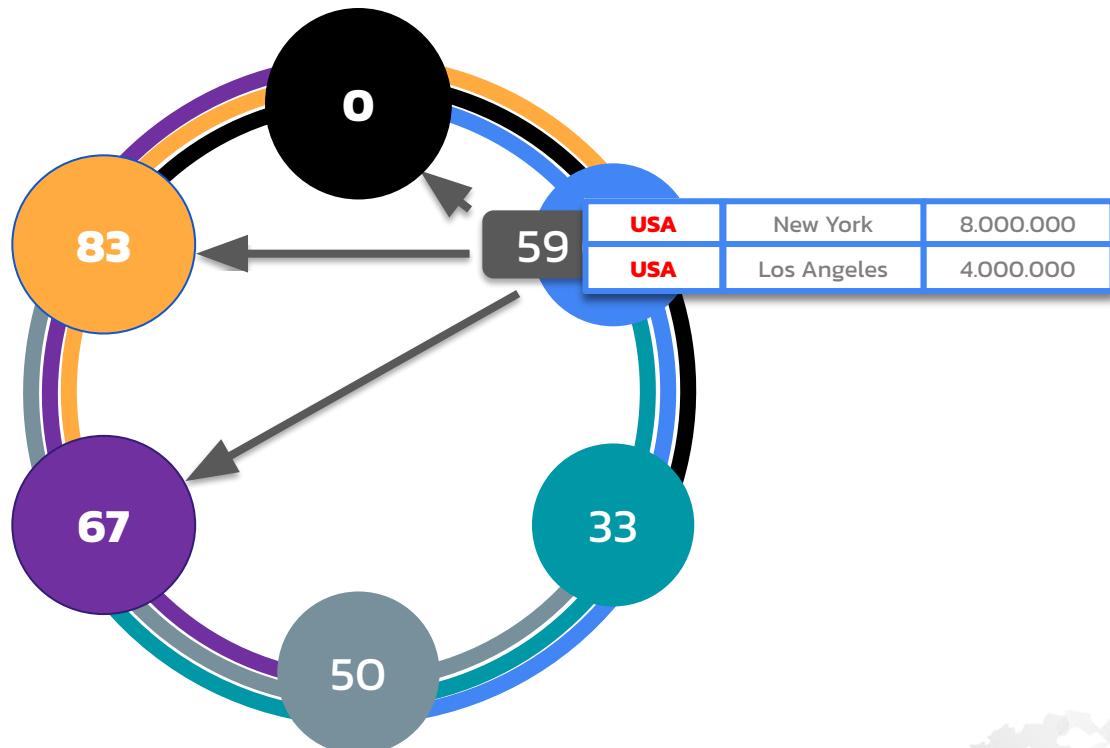
Partitioning + Replication

RF = 3



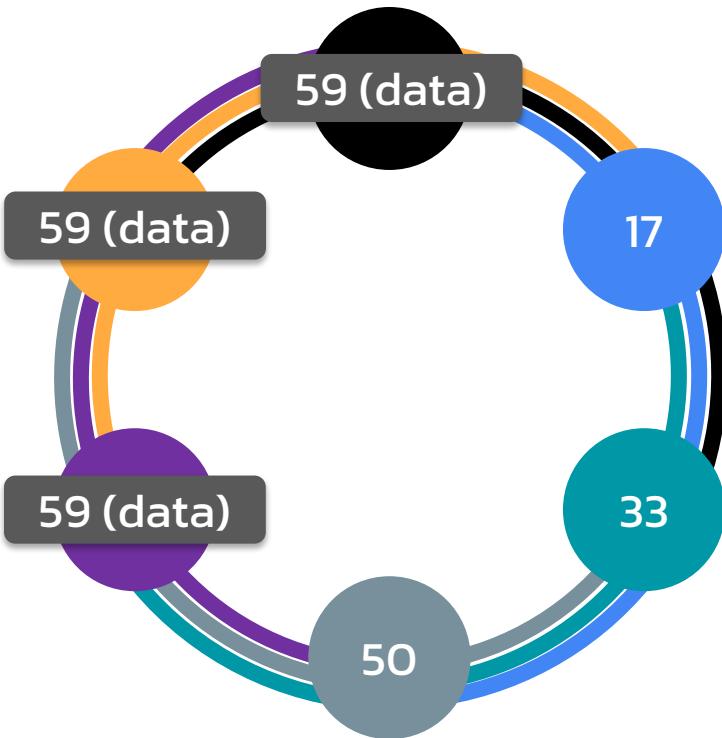
## Partitioning + Replication

RF = 3



## Partitioning + Replication

RF = 3



Partitioning + Replication

## **IMPORTANT**

Each Cassandra node and even each Cassandra driver knows Data Allocation in a cluster (it's called Token-Aware), so your application can contact literally ANY server and still get the answer.

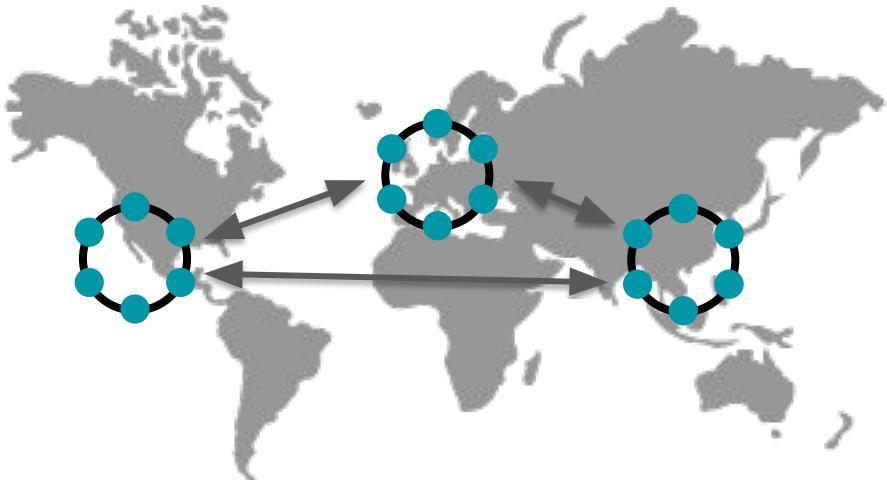




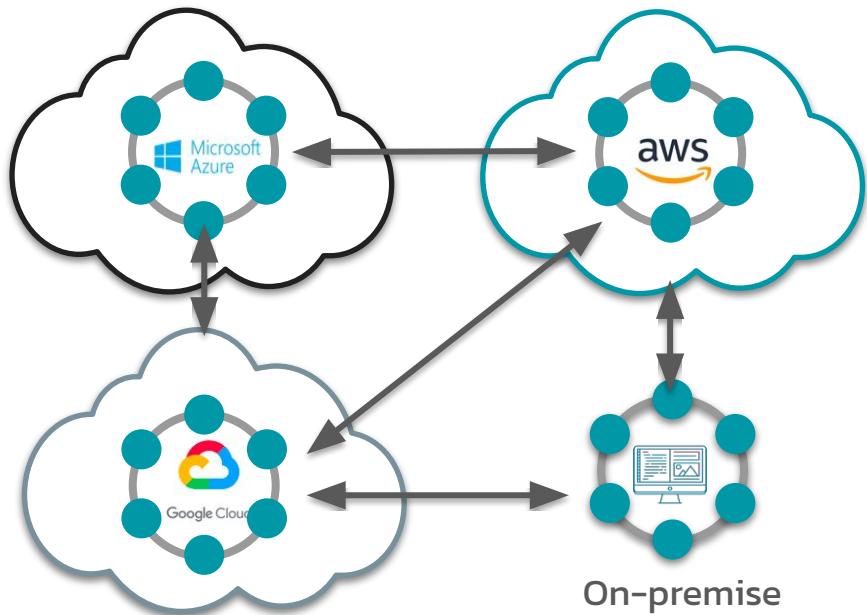
Data is GLOBALLY distributed



## Geographical Distribution



## Hybrid-Cloud and Multi-Cloud



Data is globally distributed

# 01



**Intro to Bootcamp 2022**  
**HouseKeeping**

# 02

**Intro to NoSQL Databases**  
**Power of Purpose-built DB**

# 03

**Intro to Cassandra**  
**Peer-to-Peer Database**

# 04

**CAP Theorem**  
**Law of Distributed Systems**

# 05

**Tables and Partitions**  
**How do you structure data**

# 06

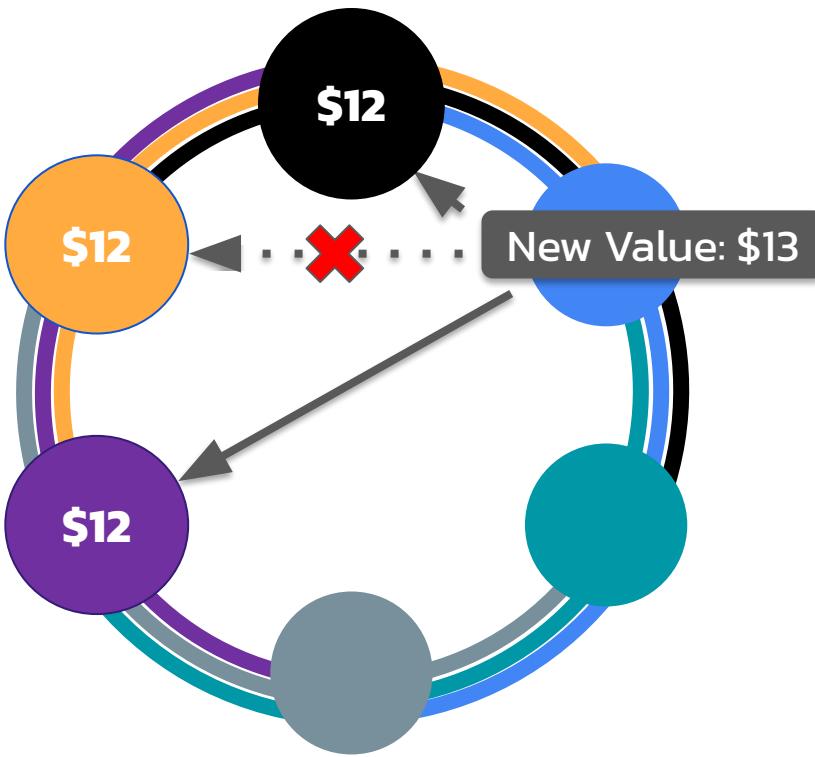
**What's next?**  
**Quiz, Homework, Next week**



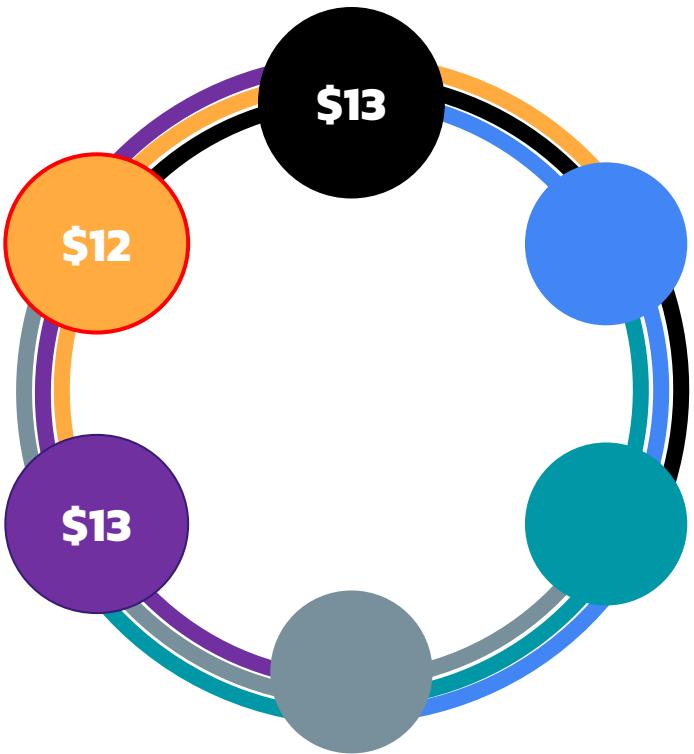
**Agenda**



# What is the biggest problem of replication?

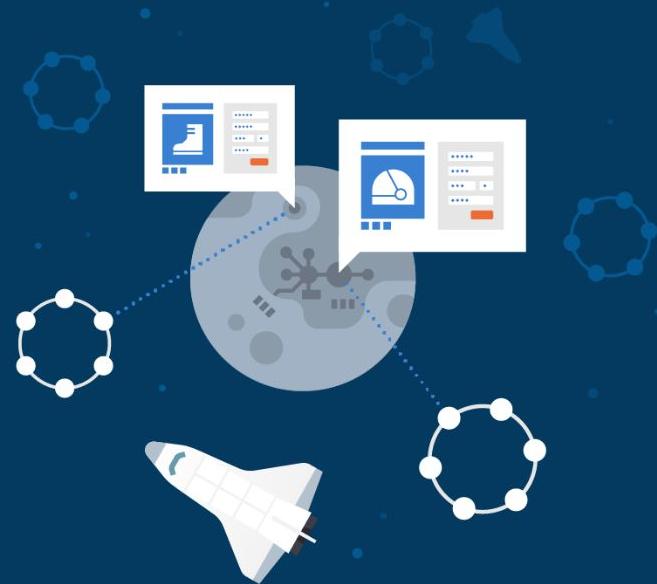


Potential Inconsistency

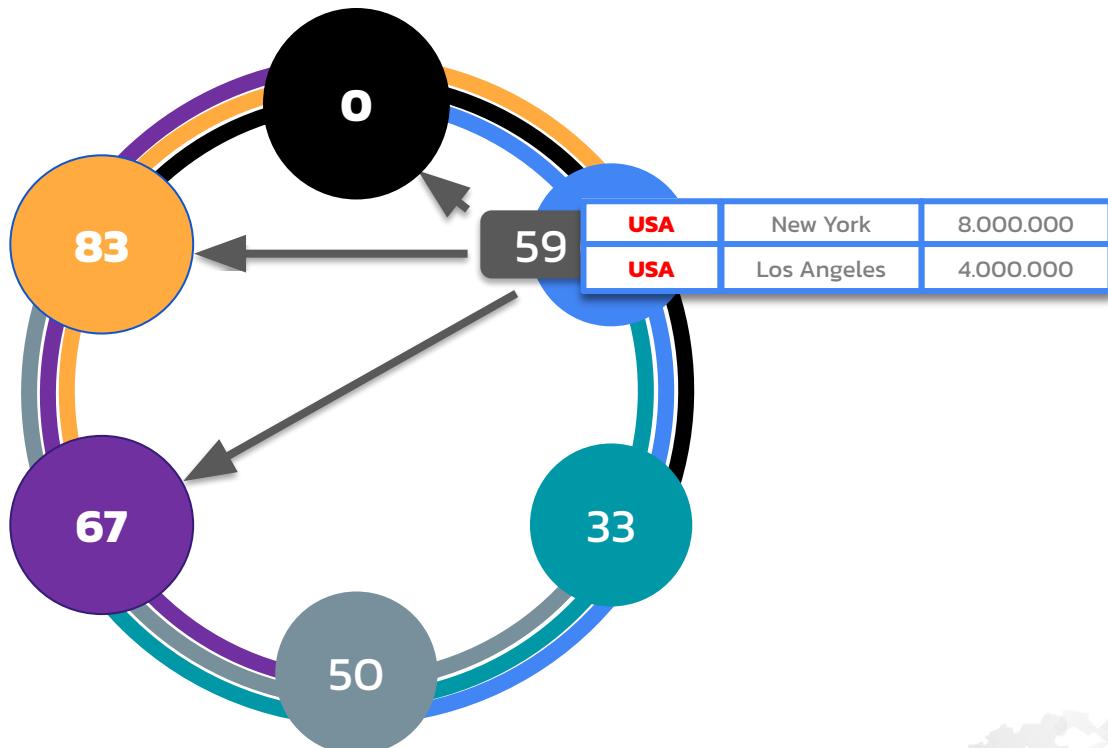


Potential Inconsistency

# Cassandra Layered Self-Defence

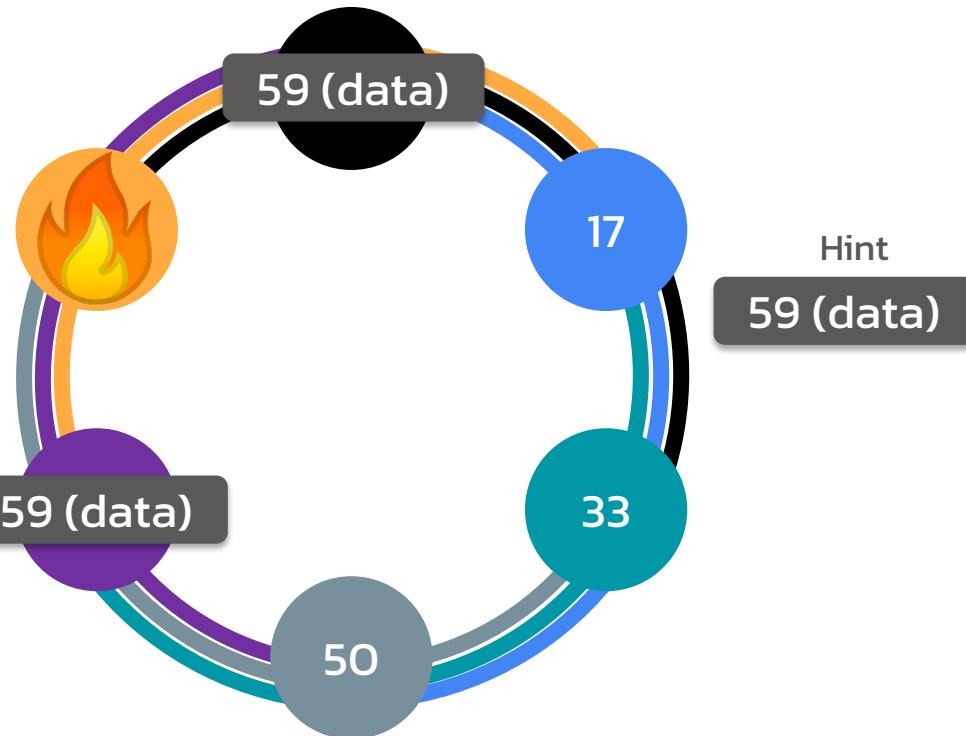


RF = 3



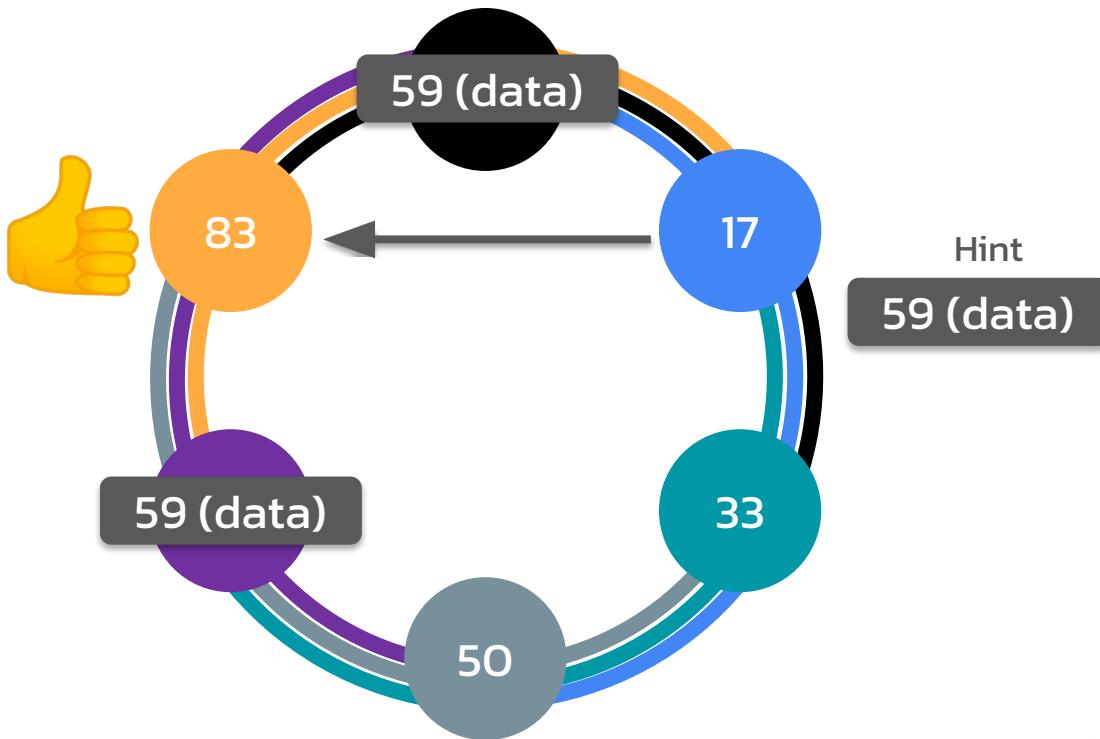
## Hinted Hand-offs

RF = 3



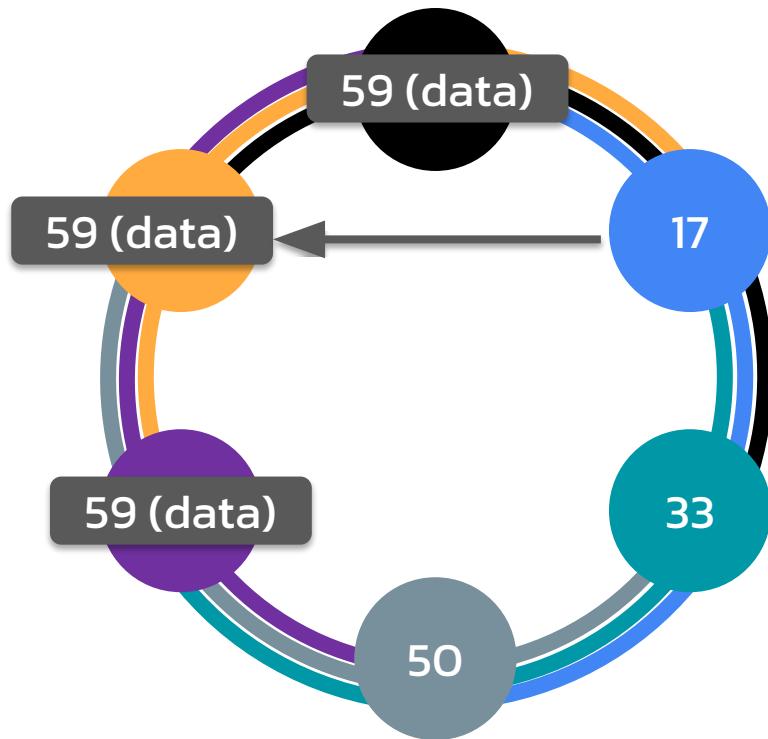
## Hinted Hand-offs

RF = 3



## Hinted Hand-offs

$RF = 3$



**Hinted Hand-offs**

# CAP Theorem



CAP Theorem operates three features:

1. Availability
2. Consistency
3. Partition Tolerance



## CAP Theorem Fundamentals

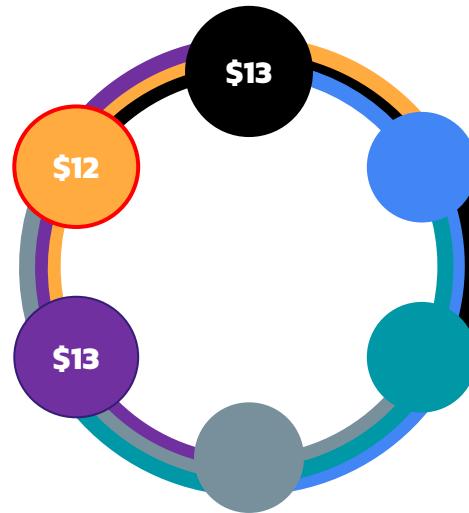


**Availability** basically means “Uptime”. You ask the question, you get the answer. If failure of a single or even multiple servers doesn’t lead to no response (no downtime), your system is **available**.



## CAP Theorem: Availability

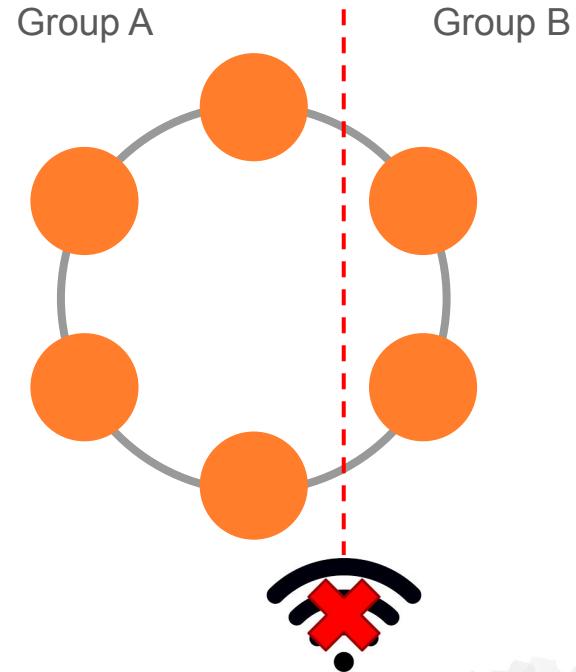
**Consistency** means “no stale data”. You ask for something, you get the most recent value. If one of your servers return outdated information, your system is **inconsistent**.



## CAP Theorem: Consistency

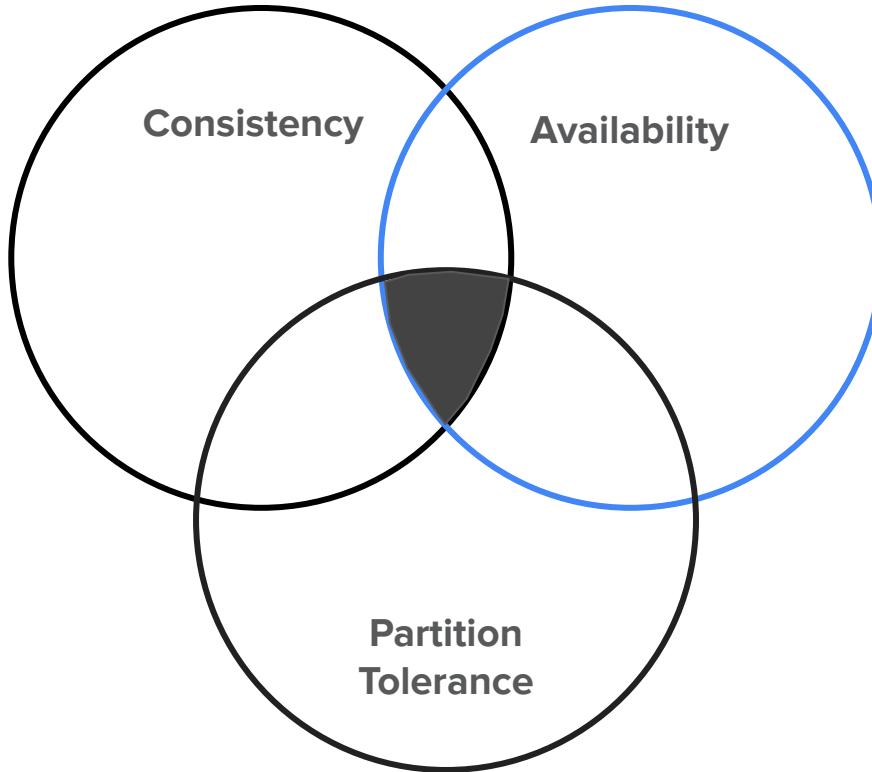


**Partition Tolerance** is the ability of a distributed system to survive “network partitioning”. Network partitioning means that the part of the servers can not reach the second part.



CAP Theorem: Partition Tolerance

In the distributed environment **in case of emergency** you can have only two guaranteed qualities out of three :(



## CAP Theorem

Cassandra is configurable consistent. In any moment of the time, for any particular query you can set the Consistency Level you require to have. It defines how many **CONFIRMATIONS** you'll wait before the response is dispatched;

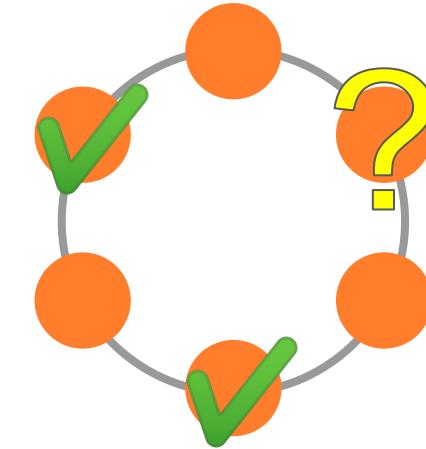
```
PreparedStatement pstmt = session.prepare(  
    "INSERT INTO product (sku, description) VALUES (?, ?)"  
);  
pstmt.setConsistencyLevel(ConsistencyLevel.ONE);
```

```
cqlsh> CONSISTENCY
```

```
Current consistency level is QUORUM.
```

```
cqlsh> CONSISTENCY ALL
```

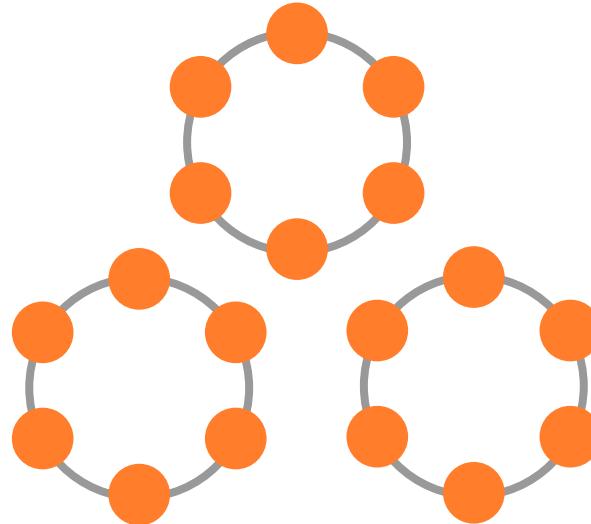
```
Consistency level set to ALL.
```



Is Cassandra AP or CP?



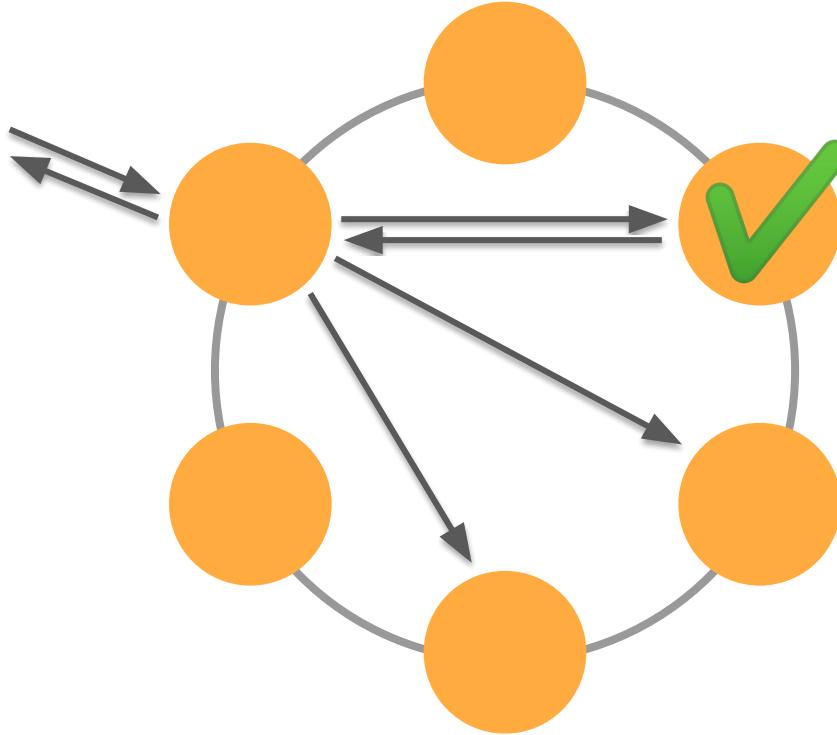
- ANY
- ONE
- LOCAL\_ONE
- TWO, THREE
- QUORUM
- LOCAL\_QUORUM
- EACH\_QUORUM
- ALL



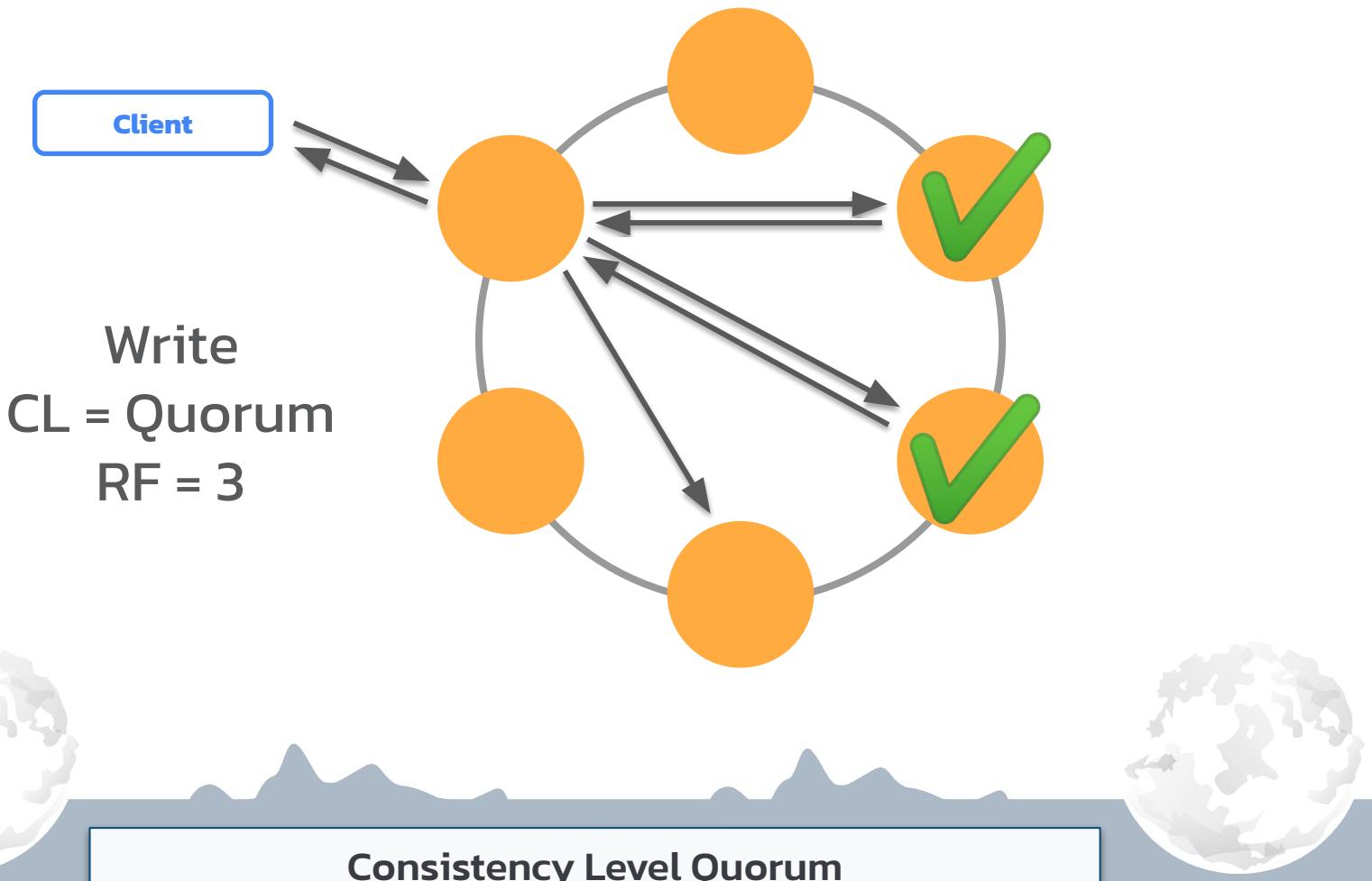
## Query Consistency Levels

**Client**

Write  
CL = ONE  
RF = 3

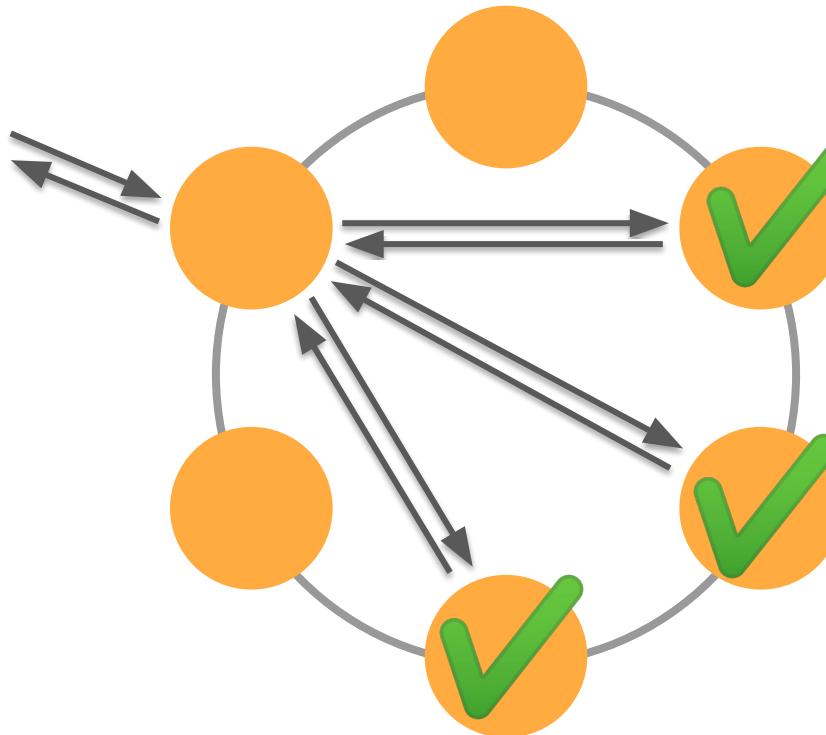


**Consistency Level One**



**Client**

**Write**  
**CL = ALL**  
**RF = 3**



**Consistency Level ALL**

Client

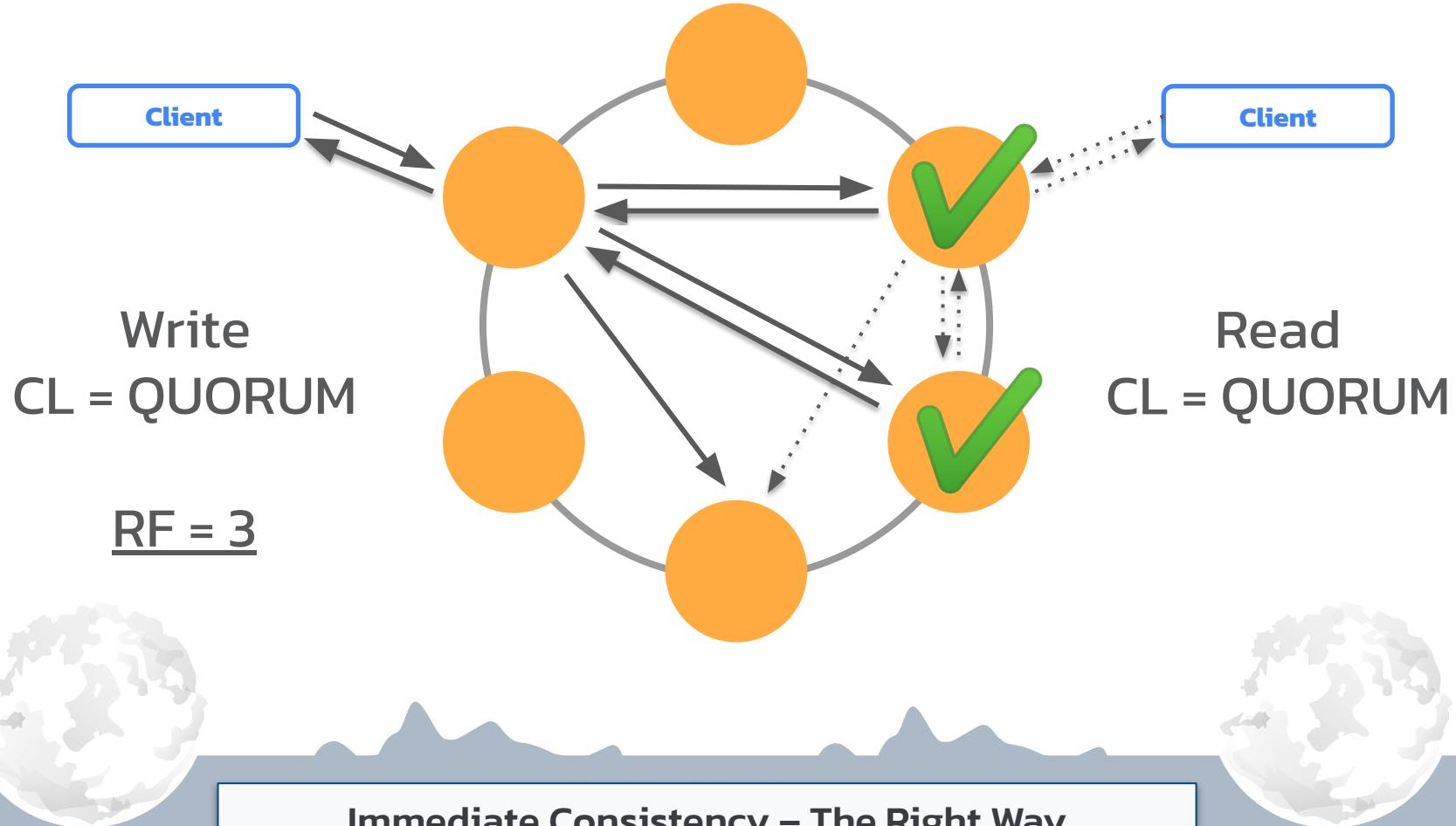
Writes  
CL = ALL  
RF = 3

IT'S A TRAP!

CAP THEOREM IS STILL HERE!

Consistency Level ALL

CL Write + CL Read > RF → Immediate Consistency



# 01



Intro to Bootcamp 2022  
**HouseKeeping**

# 02

Intro to NoSQL Databases  
**Power of Purpose-built DB**

# 03

Intro to Cassandra  
**Peer-to-Peer Database**

# 04

**CAP Theorem**  
**Law of Distributed Systems**

# 05

**Tables and Partitions**  
**How do you structure data**

# 06

**What's next?**  
**Quiz, Homework, Next week**



Agenda

# Data Structure: a Cell



An intersection of a row  
and a column, stores data.



Data Structure: a Cell

# Data Structure: a Row



A single, structured data item in a table.

1	John	Doe	Wizardry
---	------	-----	----------



Data Structure: a Row

# Data Structure: a Partition



A group of rows having the same partition token, a base unit of access in Cassandra.

IMPORTANT: stored together, all the rows are guaranteed to be neighbors.

ID	First Name	Last Name	Department
1	John	Doe	Wizardry
399	Marisha	Chavez	Wizardry
415	Maximus	Flavius	Wizardry



Data Structure: a Partition



# Data Structure: a Table



A group of columns and rows storing partitions.

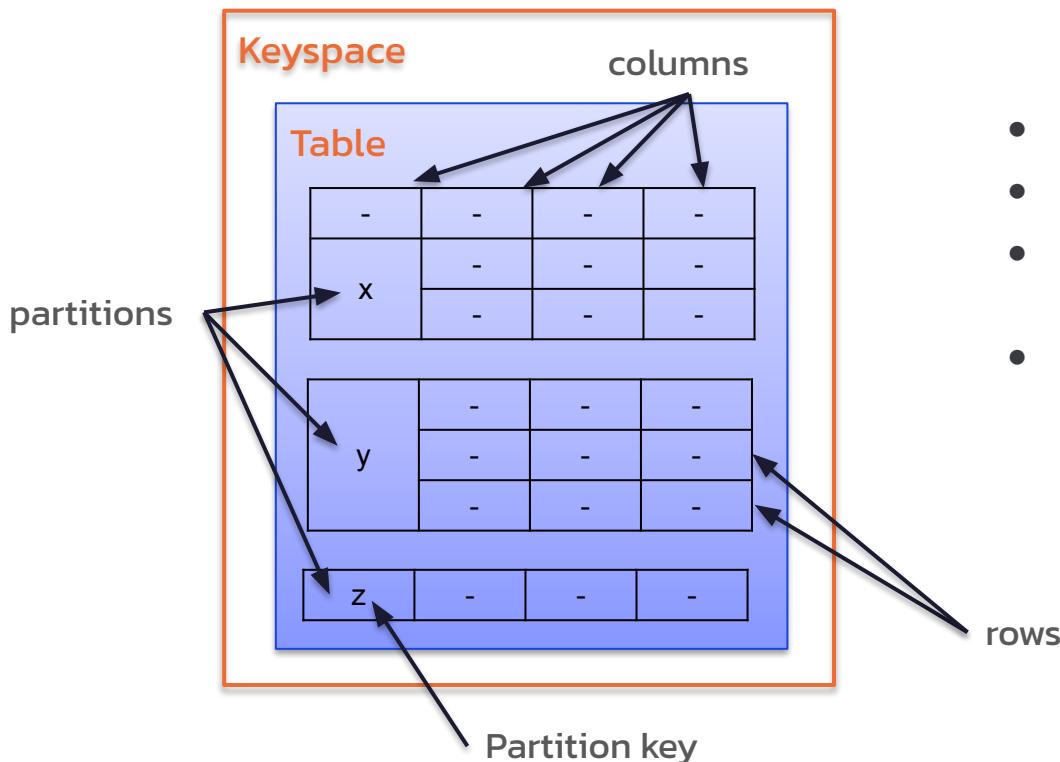
ID	First Name	Last Name	Department
1	John	Doe	Wizardry
2	Mary	Smith	Dark Magic
3	Patrick	McFadin	DevRel



Data Structure: a Table



# Data Structure: Overall

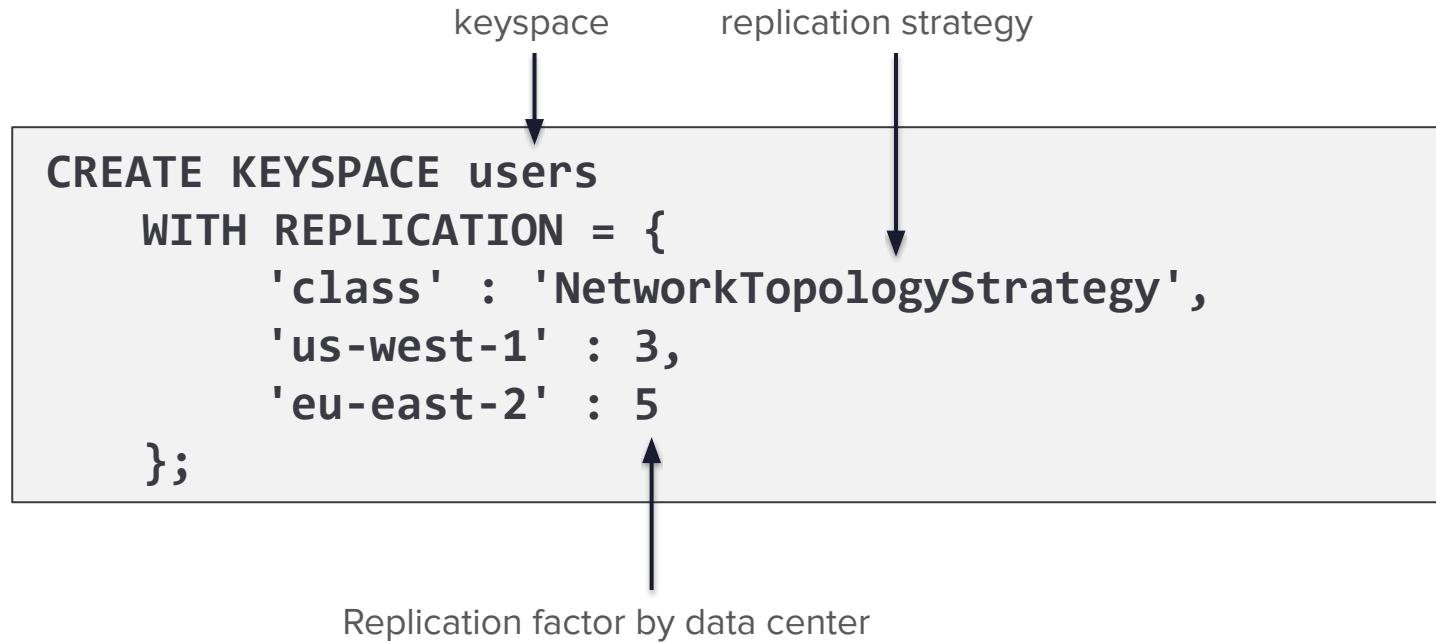


- Tabular data model, with one twist
- *Tables* are organized in *rows* and *columns*
- Groups of related rows called *partitions* are stored together on the same node (or nodes)
- Each row contains a *partition key*
  - One or more columns that are hashed to determine which node(s) store that data

# Creating a Table in CQL



# Data is Replicated



# Primary Key

An identifier for a row. Consists of at least one Partition Key and zero or more Clustering Columns.

**MUST ENSURE UNIQUENESS.**  
**MAY DEFINE SORTING.**

```
CREATE TABLE killrvideo.users_by_city (
    city text,
    last_name text,
    first_name text,
    address text,
    email text,
    PRIMARY KEY ((city), last_name, first_name, email));
```



Examples:

```
PRIMARY KEY ((city), last_name, first_name, email);
```

```
PRIMARY KEY (user_id);
```

# Partition Key

An identifier for a partition.

Consists of at least one column,  
may have more if needed

**PARTITIONS ROWS.**

```
CREATE TABLE killrvideo.users_by_city (
    city text,
    last_name text,
    first_name text,
    address text,
    email text,
    PRIMARY KEY ((city), last_name, first_name, email));
```



Examples:

```
PRIMARY KEY (user_id);
```

```
PRIMARY KEY ((video_id), comment_id);
```

# Clustering Column(s)

Used to ensure uniqueness and sorting order. Optional.

```
CREATE TABLE killrvideo.users_by_city (
    city text,
    last_name text,
    first_name text,
    address text,
    email text,
    PRIMARY KEY ((city), last_name, first_name, email));
```

Partition key

Clustering columns

PRIMARY KEY ((city), last\_name, first\_name);



**Not Unique**

PRIMARY KEY ((city), last\_name, first\_name, email);



PRIMARY KEY ((video\_id), comment\_id);



**Not Sorted**

PRIMARY KEY ((video\_id), created\_at, comment\_id);



# 01



**Intro to Bootcamp 2022**  
**HouseKeeping**

# 02

**Intro to NoSQL Databases**  
**Power of Purpose-built DB**

# 03

**Intro to Cassandra**  
**Peer-to-Peer Database**

# 04

**CAP Theorem**  
**Law of Distributed Systems**

# 05

**Tables and Partitions**  
**How do you structure data**

# 06

**What's next?**  
**Quiz, Homework, Next week**



**Agenda**

# menti.com



Go to [www.menti.com](http://www.menti.com) and use the code 3491 9972

## Inequality predicates are allowed on ...

A bar chart titled "Inequality predicates are allowed on ...". The y-axis represents the count of inequality predicates, ranging from 1 to 15. The x-axis categories are "All table columns", "Partition key columns", "clustering key columns", and "No inequality predicates are allowed".

Column Type	Count
All table columns	4
Partition key columns	3
clustering key columns	15
No inequality predicates are allowed	1

Below the chart, there is a video player interface showing a video of a person speaking. The video player includes controls like play/pause, volume, and a progress bar indicating 2:10:19 / 2:26:05. The title of the video is "Big paycheck".

Go to [www.menti.com](http://www.menti.com) and use the code 3491 9972

## Leaderboard

User ID	User Name	Profile Icon
4821 p	spanda	Avatar of a green sphere
4820 p	Agent X9	Avatar of a black mask
4775 p	Sam	Avatar of a cat
4711 p	CCedrickThePresenter	Avatar of a green leaf
4468 p	shubham	Avatar of a yellow bird
4371 p	aaa	Avatar of a yellow cat
3895 p	vignesh	Avatar of a red crown
3877 p	adry	Avatar of a smiling face
3861 p	Millie	Avatar of a brown bear
3812 p	Puggie	Avatar of a brown dog

Below the leaderboard, there is a video player interface showing a video of a person speaking. The video player includes controls like play/pause, volume, and a progress bar indicating 2:11:07 / 2:26:05. The title of the video is "Big paycheck".

# SWAG WINNERS



Congratulations to 1st, 2nd and 3rd place on the Menti quiz!

To claim your prize, please send an email to:

[gary.harvey@datastax.com](mailto:gary.harvey@datastax.com)

**\*\* Include a screenshot of your Menti screen**



Swag Winners!

# Homework

!homework

🎓🔥 Intro to Cassandra for Developers  
using DataStax Astra DB 🔥🎓



The course card has a dark blue header with a white eye icon and the text "DS201". The main body is white with dark blue text. It reads "DS201: Foundations of Apache Cassandra™ and DataStax" and "Online Course · 4h". At the bottom is a blue button with the word "Resume" and a small white icon.



DataStax Developers



# Next Week

**Design a Data Model for a Full-Stack Application**

12 January, 2022

#NoSQL, #Apache Cassandra, #APIs, #DataModeling #Certification

Python Java JS Apache Cassandra AWS Lambda

WED JAN 12 2022

Bootcamp - Design a Data Model for a Full-Stack Application

Register Now

# Join our 17k Discord Community

## DataStax Developers



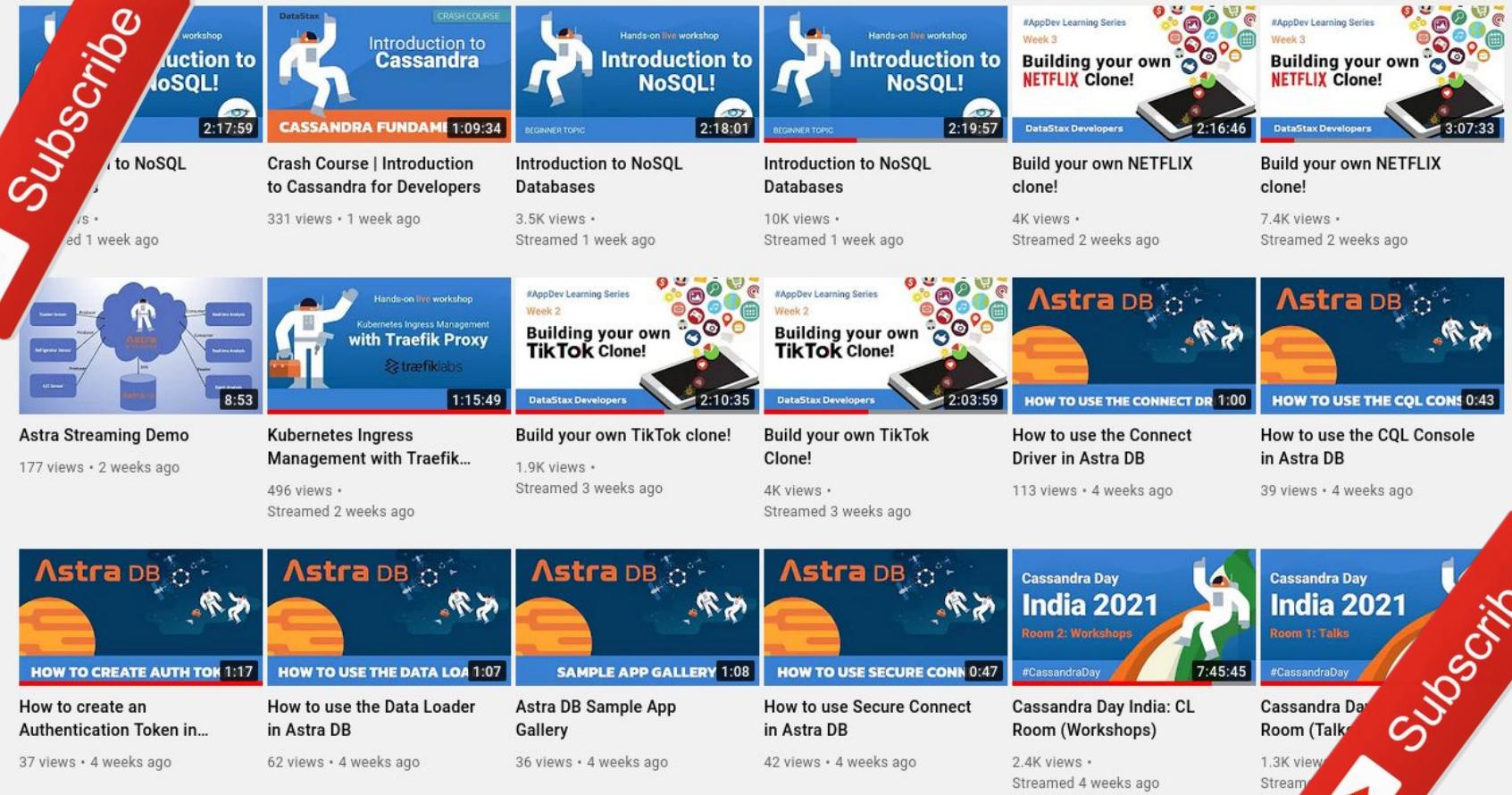
!discord

[dtsx.io/discord](https://dtsx.io/discord)

The screenshot shows the DataStax Developers Discord server interface. The left sidebar lists various channels: Événements, # moderator-only, # WELCOME, start-here, code-of-conduct, # introductions, upcoming-events, useful-resources, # memes, # your-ideas, @ the-stage, # WORKSHOPS, # workshop-chat, # workshop-feedback, # workshop-materials, # upcoming-workshops, # ASTRADB, # getting-started, # astra-apis, # astra-development, # sample-applications, and # APACHE CASSANDRA. The # workshop-chat channel is currently active, displaying a message from user RIGGITYREKT about mixed DSE versions and another from Cedrick Lunven about removing analytics parameters. The right sidebar shows a list of presenters and helpers, and a list of users currently online.

Discord Community

# Subscribe



# Subscribe

DataStax Developers

# Thank You!

