DataStax

Developers

SKY TRAINING

Introduction to Cassandra

Building efficient applications with Apache Cassandra

DataStax Sky account team

James Kenney – Account Manager
Amarjit Basra – Data Architect
Matt Chalmers – PS manager
Astyn Gould – Astra Success Team

Sky Slack Channel : #datastax-sky-shared-channel (95 members)

Face-to-Face meetings at Osterley Site

Director of Developer Relations

clunven

clunven

clun

- Trainer
- Public Speaker
- Developers Support
- Developer Applications
- Developer Tooling

- Creator of ff4j (ff4j.org)
- Maintainer for 8 years+

- Happy developer for 14 years
- Spring Petclinic Reactive & Starters
- Implementing APIs for 8 years

DataStax

sky

Cédrick Lunven

Solution Architect

- Joined DataStax in January 2022
- 28 years in Enterprise Software
- Wide industry experience
- Support Developers and Enterprises in creating and optimising Cassandra & DSE Solutions
- Back-end technology “junkie”

DataStax

Phil Miesle

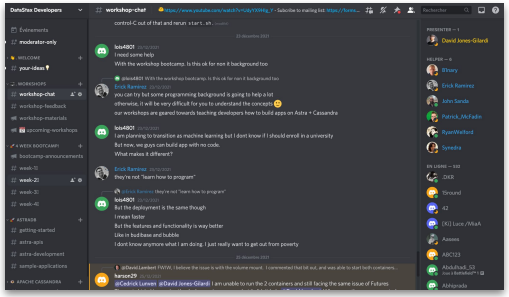
sky

DataStax Developers

Games and quizzes: menti.com

Mentimeter

Questions: https://dtsx.io/discord



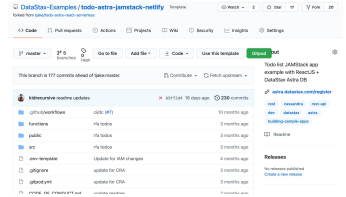
Discord
(#workshop-chat)

DataStax

sky

Interactive Workshops

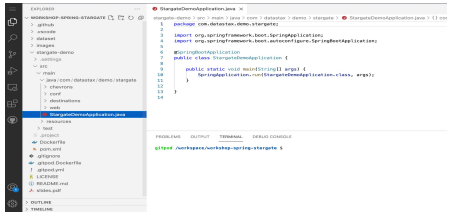
Source code + exercises + slides



GitHub

!github

IDE



Gitpod

!gitpod

Nothing to install !

Database + Api + Streaming

DataStax Astra DB

!astra

Hands-On Housekeeping

572 and counting

Intro to Cassandra Workshop

Awarded to **YOUR NAME** • attendee@workshop.com

Issued on **Jun 16, 2021**

Offered By
DataStax Developers

Upgrade Complete! This badge is to certify successful completion of the DataStax Cassandra Workshop: "Fundamentals of Apache Cassandra".

Verified

Last verified by Badgr on **Jun 17, 2021**

Re-verify Badge

EARNING CRITERIA

Recipients must complete the earning criteria to earn this Badge

To earn this badge, individuals must complete the follow
during the **Intro to Cassandra Workshop**:

- Attend the lecture
- Complete the practical steps by doing all requi

Achievement Unlocked !

01

Why NoSQL ?

02

Introduction to
Apache Cassandra™

03

Distributed Architecture
The CAP Theorem

04

Tables
and Partitions

05

The Art of
Data Modelling

06

What's next?

Quiz, Homework, Next week

DataStax

sky

Agenda

01

Why NoSQL ?

02

Introduction to
Apache Cassandra™

03

Distributed Architecture
The CAP Theorem

04

Tables
and Partitions

05

The Art of
Data Modelling

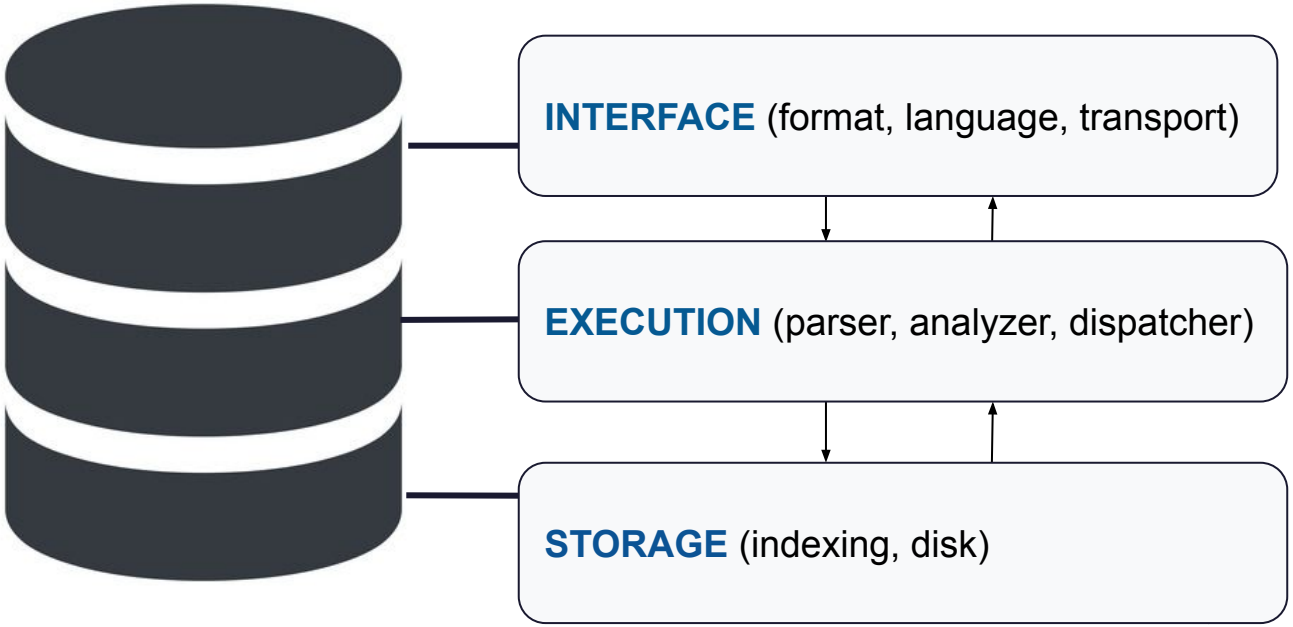
06

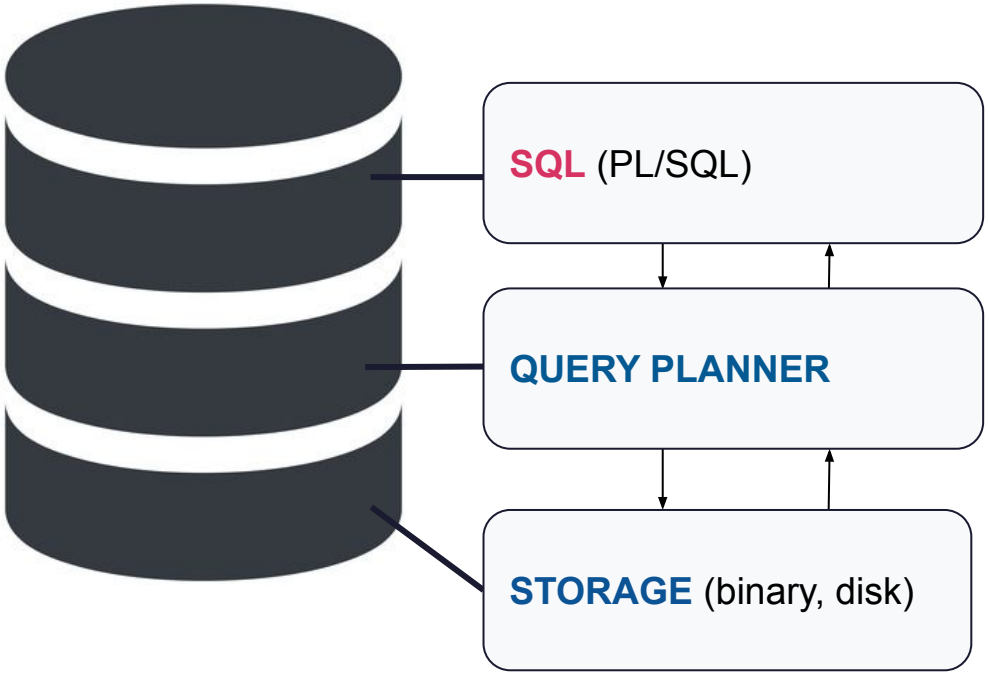
What's next?
Quiz, Homework, Next week

DataStax

sky

Agenda


INTERFACE (format, language, transport)
EXECUTION (parser, analyzer, dispatcher)
STORAGE (indexing, disk)

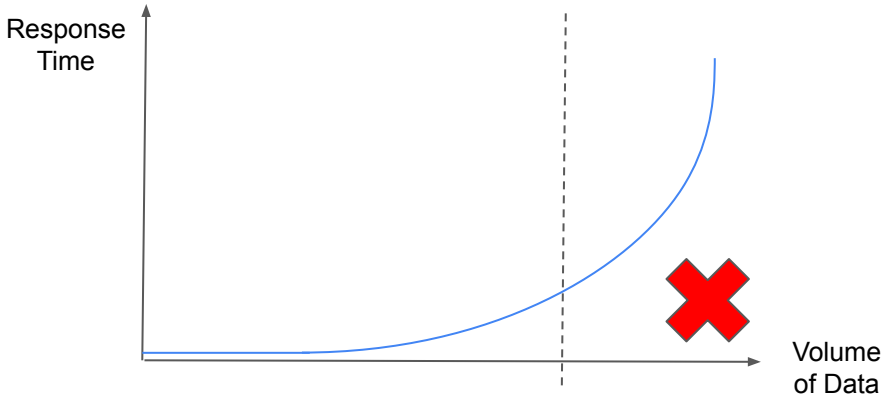

SQL (PL/SQL)
QUERY PLANNER
STORAGE (binary, disk)

OLTP: Online Transaction Processing

Fast Processing
Transactional
High number of transactions
Hot / Live Data
“Row oriented”
Normalized Data (3NF)

OLAP: Online Analytical Processing

Slow Queries
Historical
High volume of data
Cold Data
“Column oriented”
Denormalized Data



DataStax

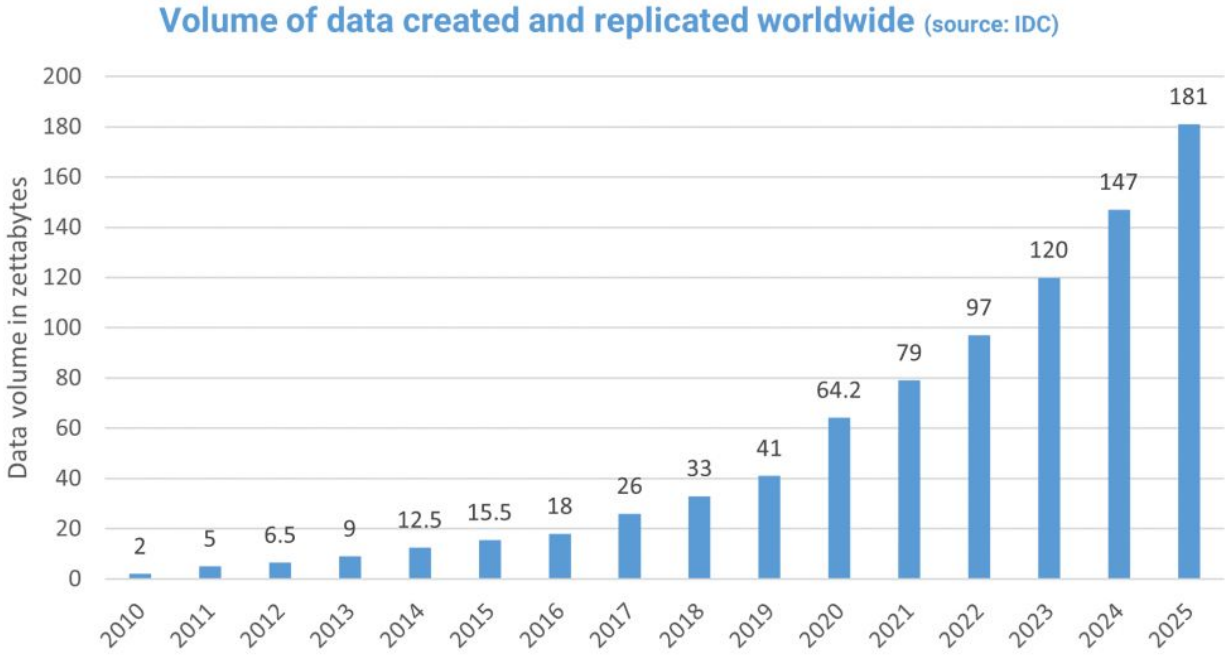
sky

Relational Databases are versatile

- They have been designed to run on a single machine

- New requirements
  - V: VOLUME
  - V: VELOCITY
  - V: VARIETY

WHAT'S A ZETTABYTE?	
1 kilobyte	1,000
1 megabyte	1,000,000
1 gigabyte	1,000,000,000
1 terabyte	1,000,000,000,000
1 petabyte	1,000,000,000,000,000
1 exabyte	1,000,000,000,000,000,000
1 zettabyte	1,000,000,000,000,000,000,000



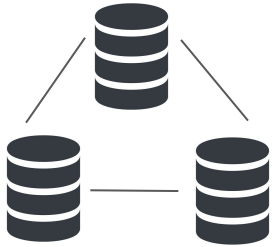
DataStax

sky

Relation Databases have limited scalability

- Meetup names on June 11, 2009 in San Francisco

- Web Giants to needed more scalable systems
  - Distributed by design (horizontal scalability)
  - Data is replicated
  - Cope with the variety of data

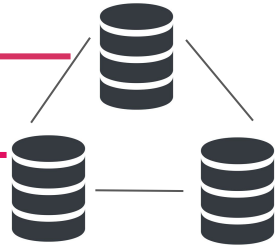


NoSQL

Not Only SQL

AP vs CP (CA = not distributed)

Server vs **VM** vs **Container** vs **Cloud**



SQL or **JSON** or **CQL** or **N1QL** or **Cypher...**

Query Parser and execution engine

Text, Binary, Graph

Ledger Databases *awsqldb, ksql_db*	Time-Series Databases *Influx, OpenTSDB, Prometheus*	Tabular Databases ***Cassandra**, Hbase, Bigtable*	Object Databases *S3, Minio, Ceph*
Document Databases *MongoDB, Elastic, DocDb*	Graph Databases *Neo4j, Datastax Graph, Titan*	Key/Value Databases *Redis, Dynamo, Memcache*	

DataStax

sky

NoSQL Databases

01

Why NoSQL ?

02

Introduction to
Apache Cassandra™

03

Distributed Architecture
The CAP Theorem

04

Tables
and Partitions

05

The Art of
Data Modelling

06

What's next?
Quiz, Homework, Next week

DataStax

sky

Agenda

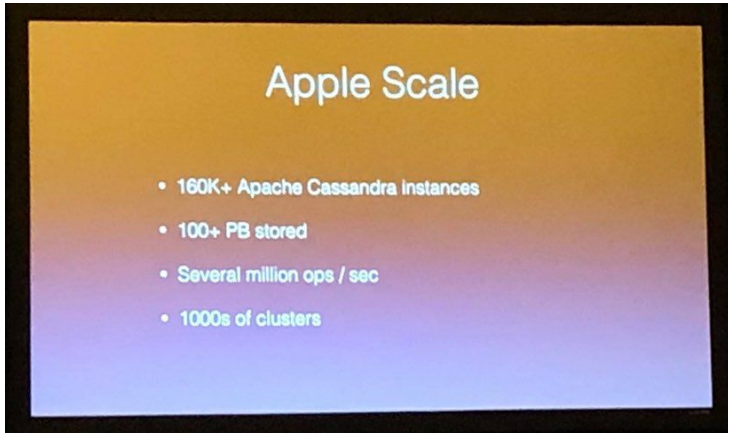
Apache Cassandra @ Netflix

- 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



And many others...

DataStax

DataStax Developers

Cassandra Biggest Users (and Developers)

Apache Cassandra

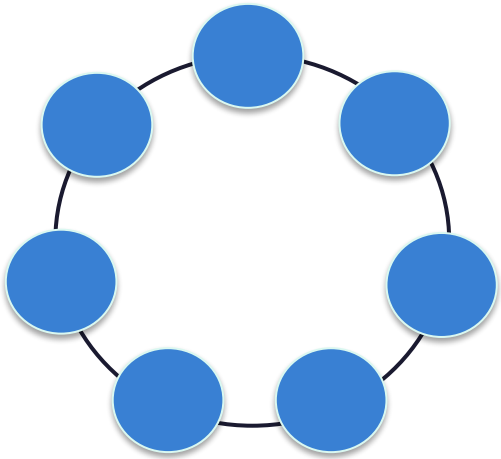
NoSQL Distributed Decentralised Database Management System

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

DataStax

sky

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



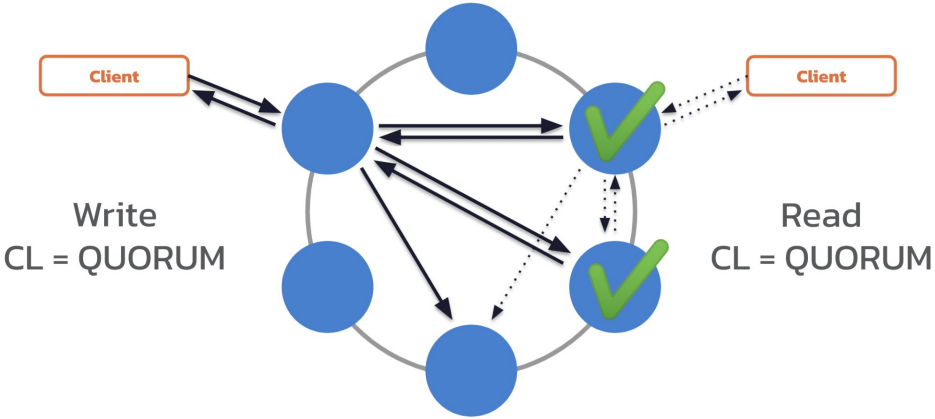
DataStax

sky

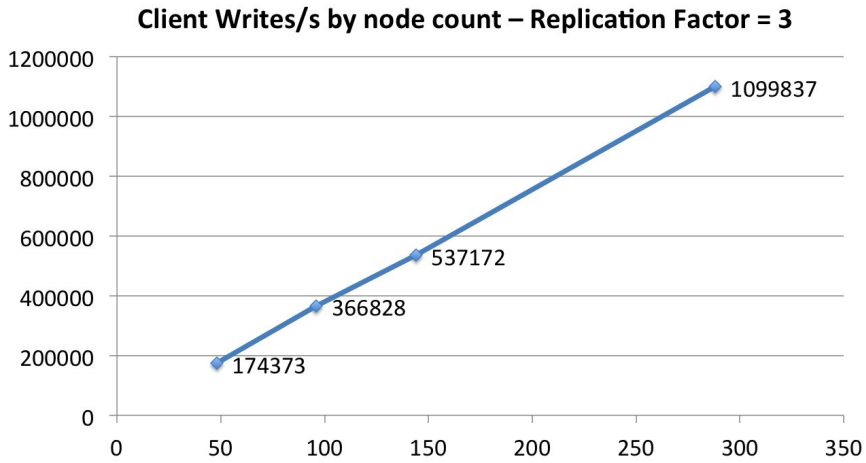
Big Data Ready

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.



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



NETFLIX

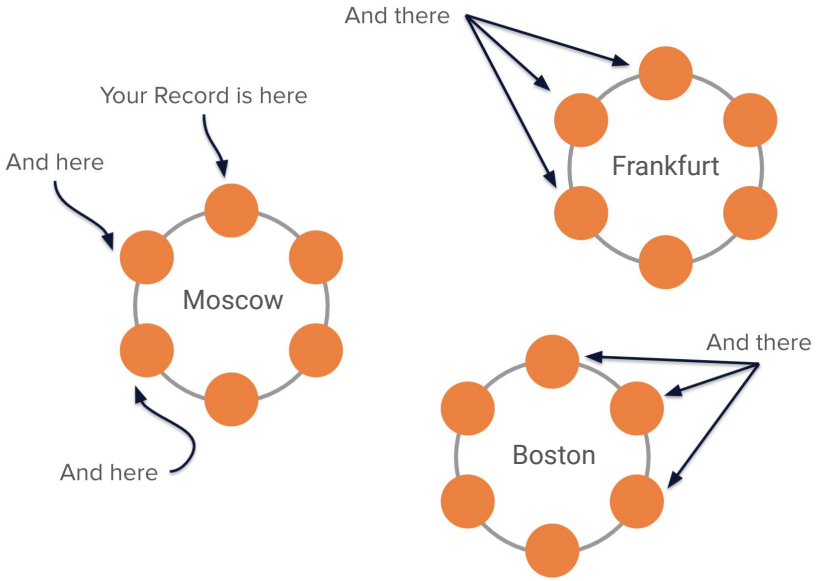
DataStax

sky

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



DataStax

DataStax Developers

Highest Availability

sky

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

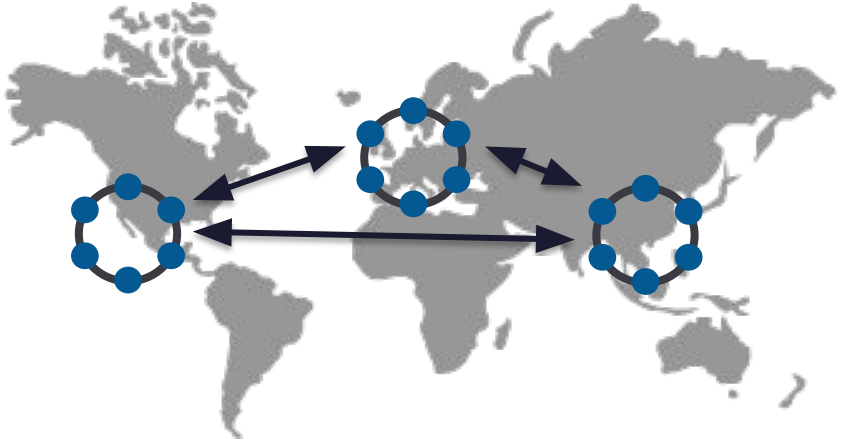
DataStax

sky

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)!



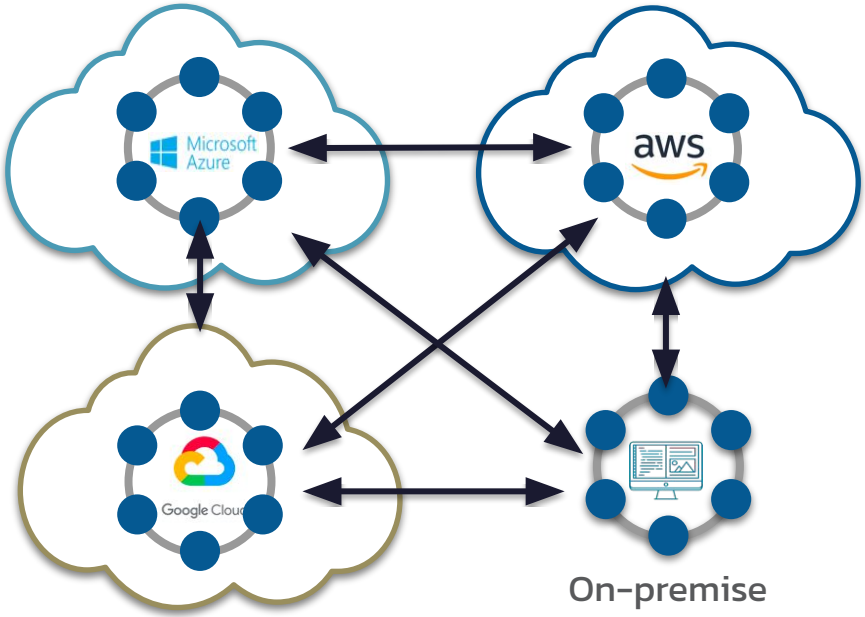
DataStax

DataStax Developers

sky

Geographical Distribution

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



DataStax

sky

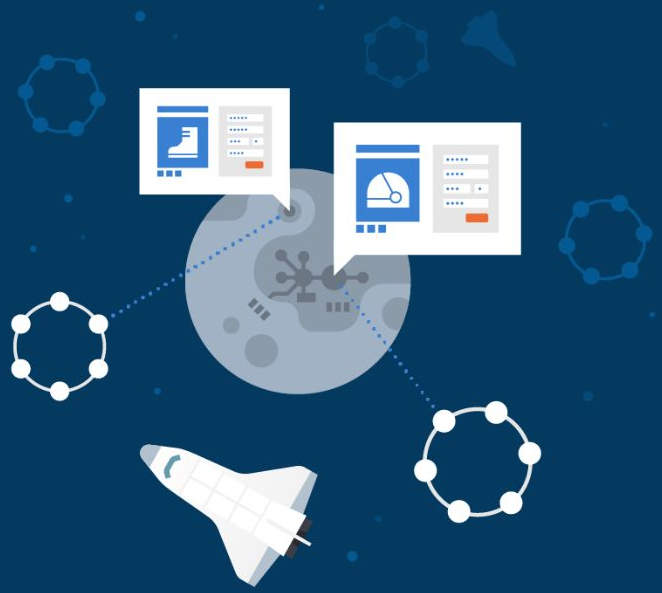
Platform Agnostic

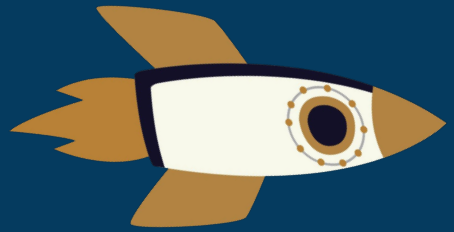
Cassandra doesn't belong to any of commercial vendors but controlled by a non-profit Open Source **Apache Software Foundation**, already familiar to you by *Hadoop*, *Spark*, *Kafka*, *Zookeeper, Maven* and many other projects.

DataStax

sky

Vendor Independent





Let’s GO (!github)

1. Objectives
2. Frequently asked questions
3. Materials (Slides)

Free to Use

Up to 80GB storage and/or 20 million operations monthly.

Serverless

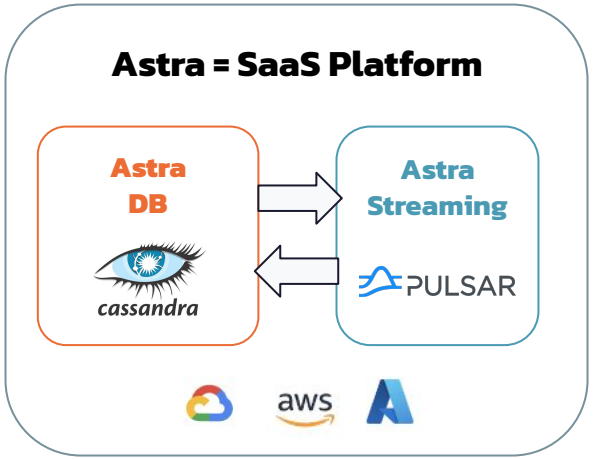
Lower your costs by running Cassandra clusters only when needed.

No Operations

Eliminate the overhead to install, operate, and scale Cassandra.

Data APIs

Work natively with Document (JSON), REST, GraphQL and gRPC APIs.



Global Scale

Put your data where you need it without compromising performance, availability or accessibility.

End-to-End Security

Secure connect with VPC peering and Private Link. Bring your own encryption key management. SAML SSO secure account access.

Zero Lock-in

Deploy on AWS, GCP or Azure and keep compatibility with open-source Cassandra.

Relational Indexes

Storage Attached Indexing (SAI) lets you query tables using any columns.

DataStax

sky

Astra = Cassandra As a Service++

Lab 1

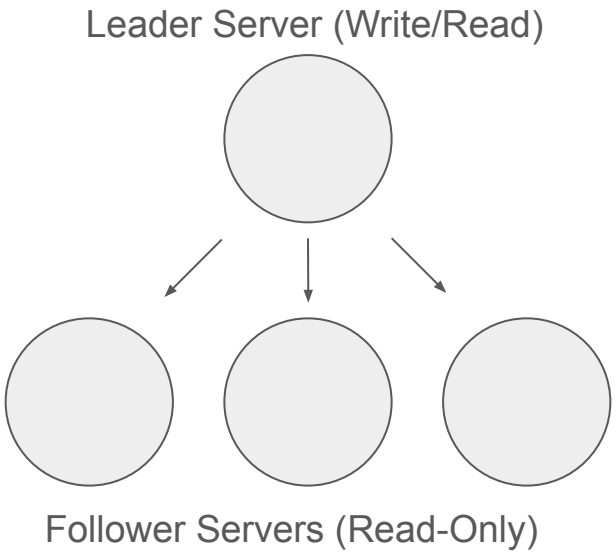
Database Setup

http://github.com/datastaxdevs/workshop-cassandra-fundamentals

- ✅ Create a new database
- ✅ Wake up an existing hibernated database

All servers are created equal

Cassandra Holy Book, P.I Paragraph I



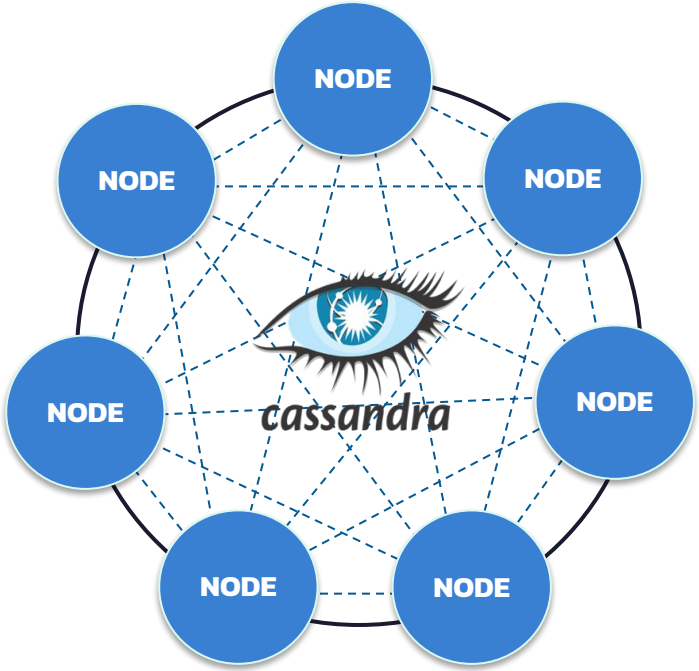
1. Single Point of Failure
2. Hard to scale for writes
3. Application needs to know where to write

DataStax

DataStax Developers

Traditional Architecture

sky



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)

DataStax

DataStax Developers

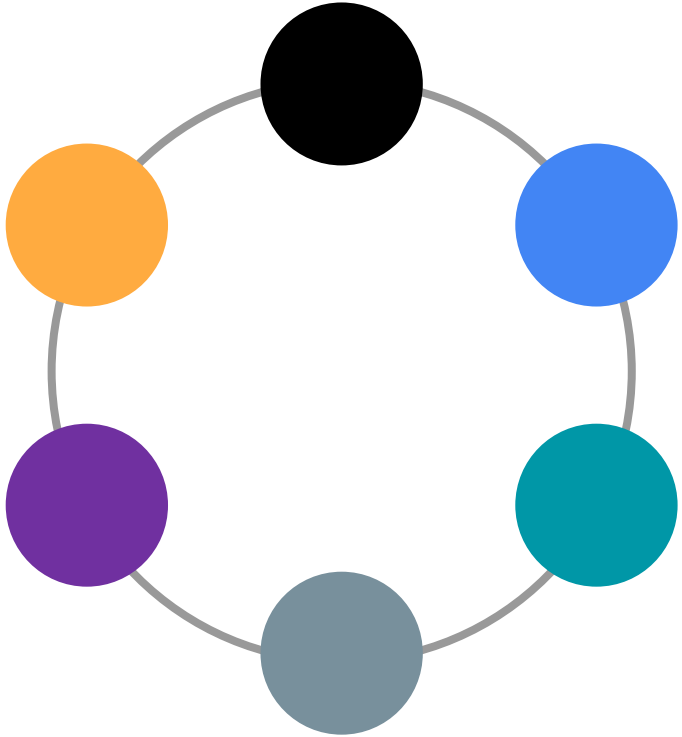
Master-less (Peer-to-Peer) Architecture

sky

Data is Replicated

RF = ?

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

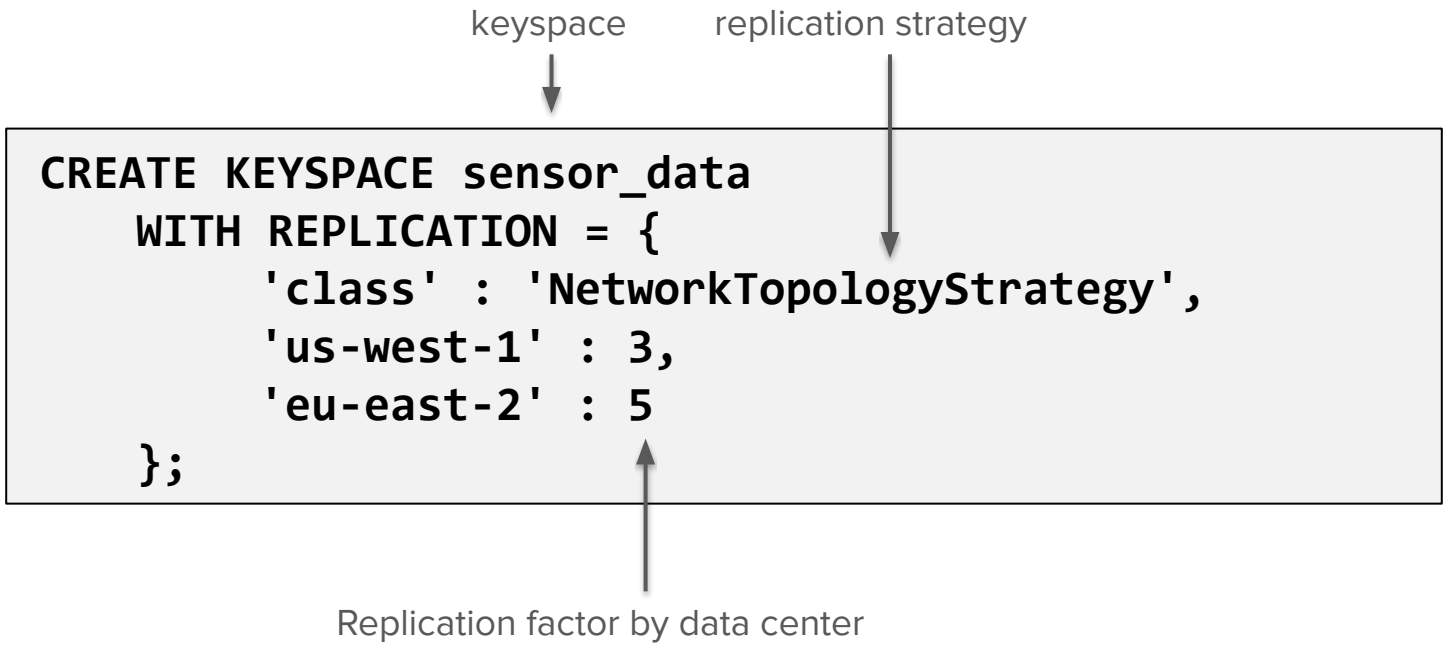


DataStax

sky

DataStax Developers

Replication Factor



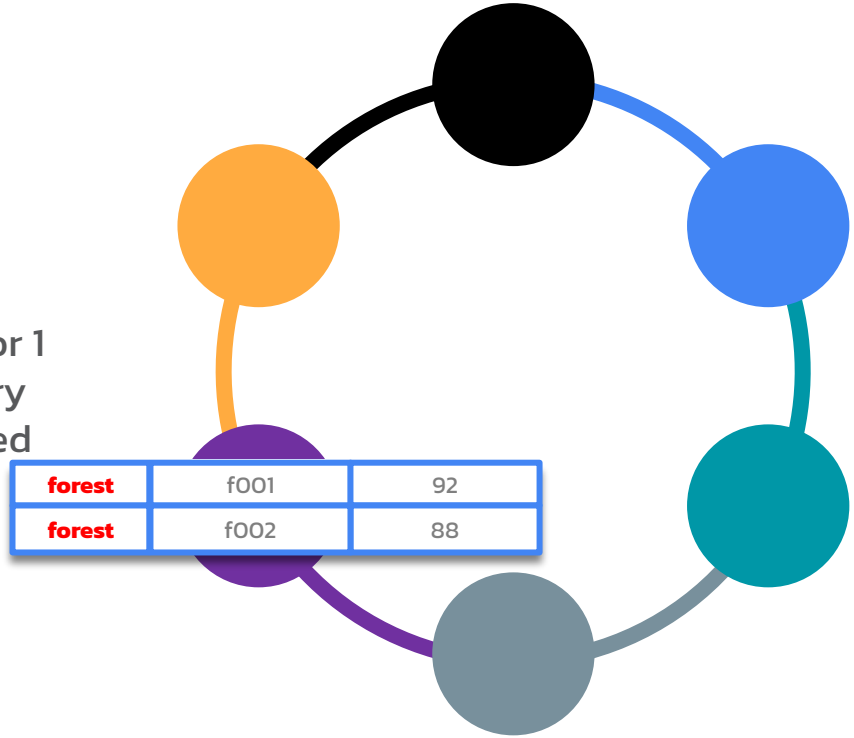
DataStax

DataStax Developers

Data is Replicated

RF = 1

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



DataStax

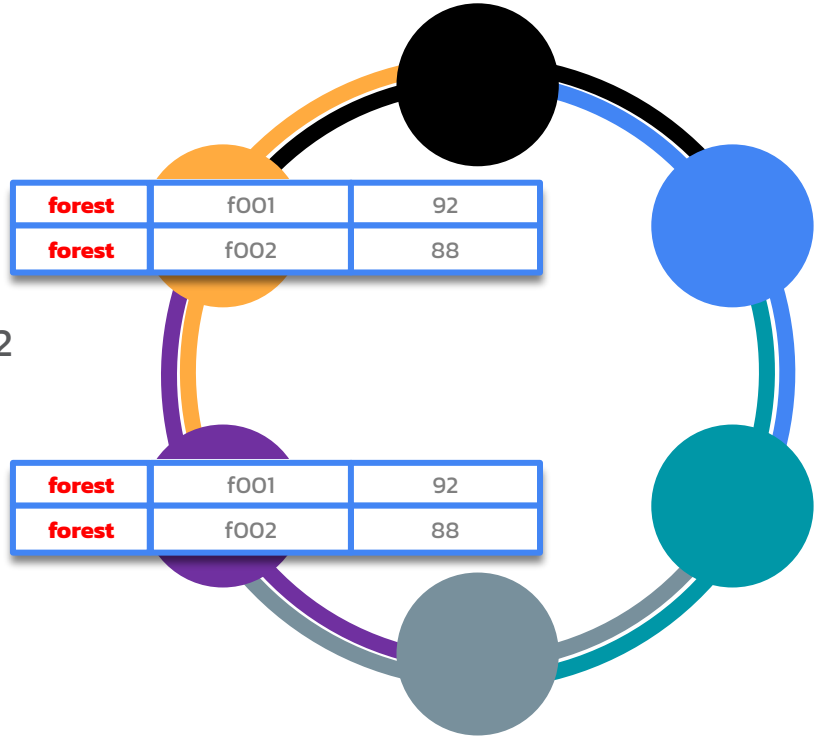
sky

DataStax Developers

Replication Factor

RF = 2

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



DataStax

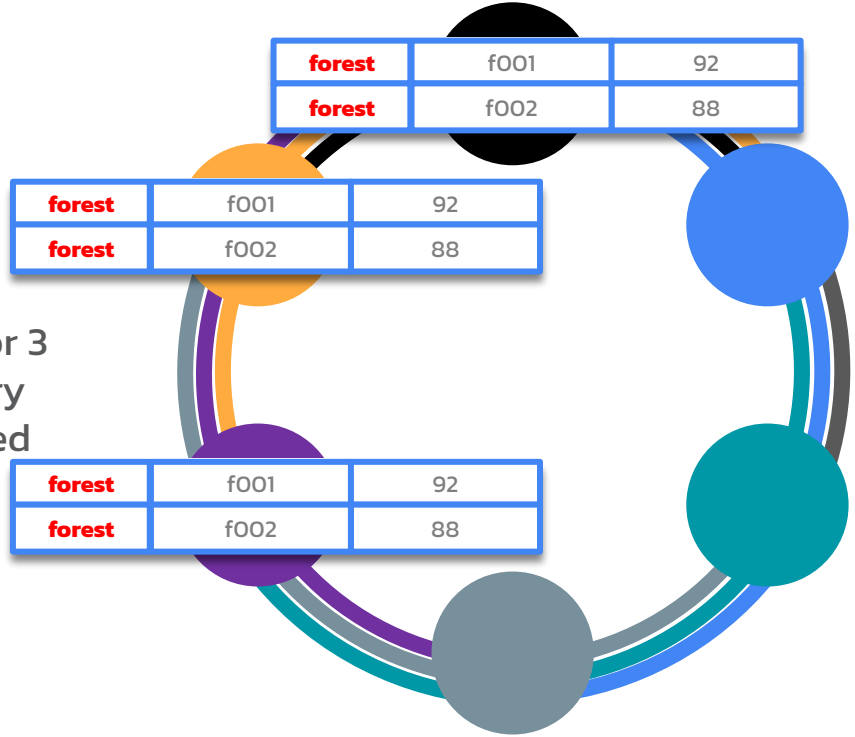
sky

DataStax Developers

Replication Factor

RF = 3

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



DataStax

sky

DataStax Developers

Replication Factor

Data is Distributed

network	sensor	temperature
forest	f001	92
forest	f002	88
volcano	v001	210
sea	s001	45
sea	s002	50
home	h001	72
car	c001	69
car	c002	70
dog	d001	40
road	r001	105
road	r002	110
ice	i001	35

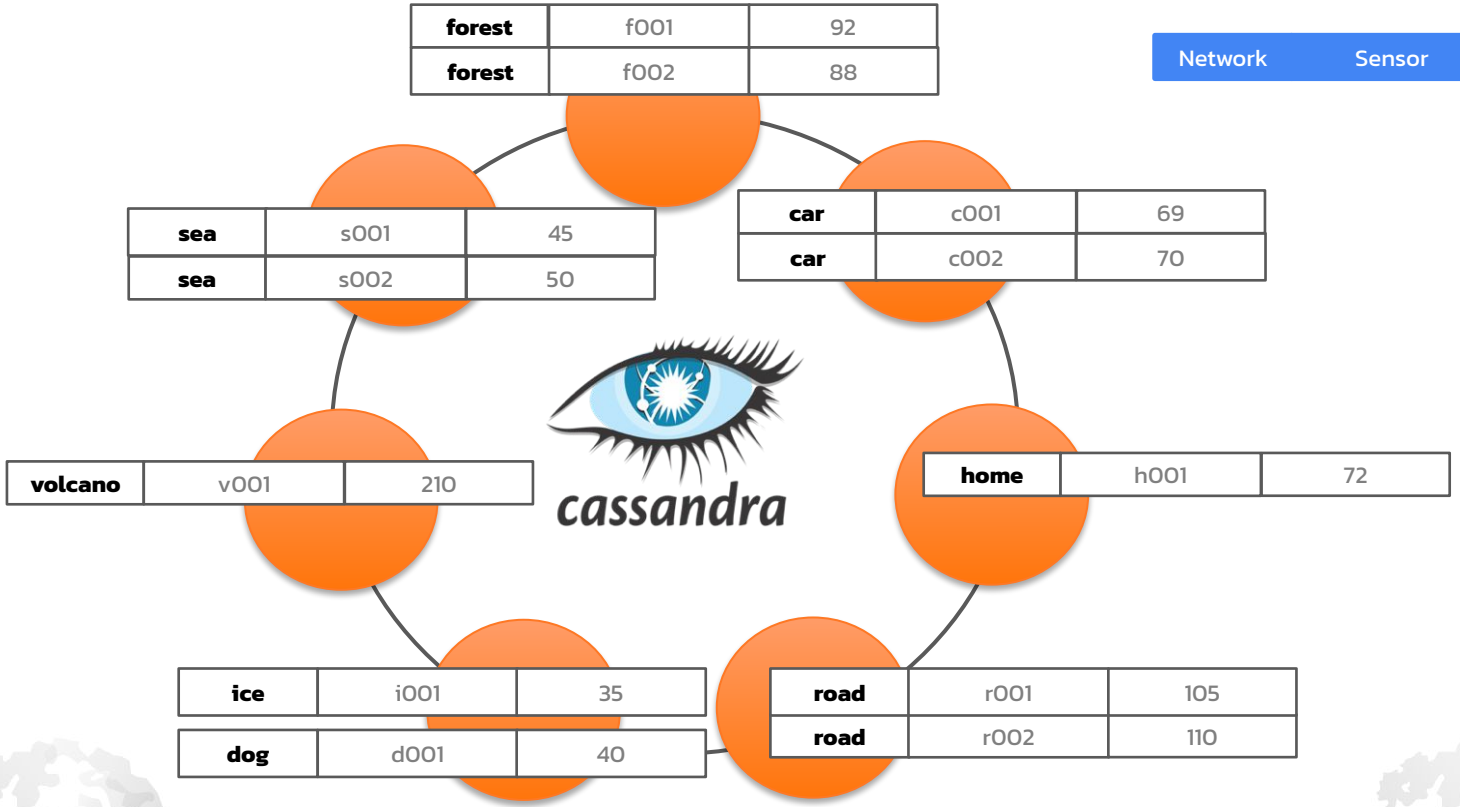
Partition Key

DataStax

DataStax Developers

sky

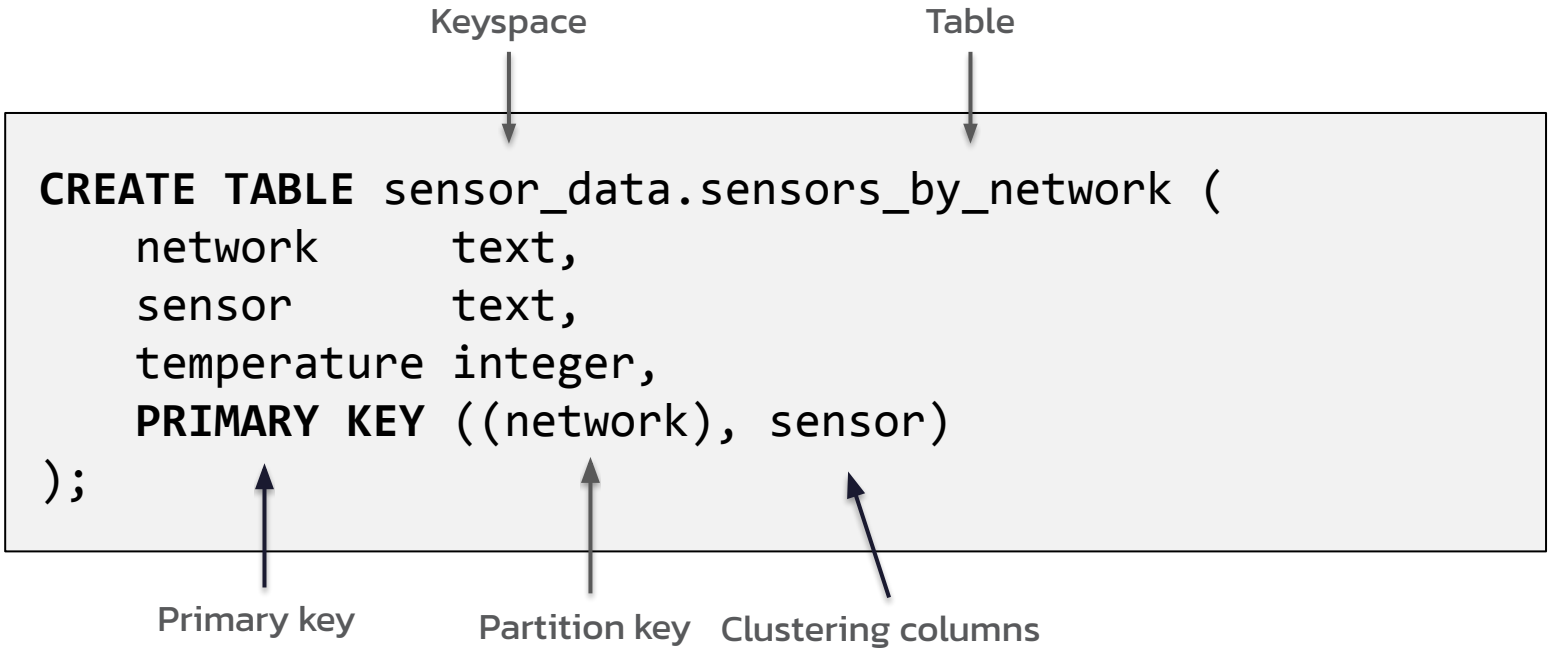
Data is Distributed



DataStax

sky

Data is Distributed



DataStax

DataStax Developers

Key-based Partitioning

network	sensor	Value
forest	f001	92
forest	f002	88
sea	s001	45
sea	s002	50
road	r001	105
road	r002	110

Partition Keys

Partitioner

Murmur3 Hashing

Network	Sensor	Value
59	f001	92
59	f002	88
12	s001	45
12	s002	50
45	r001	105
45	r002	110

Tokens

Cassandra Nodes

DataStax

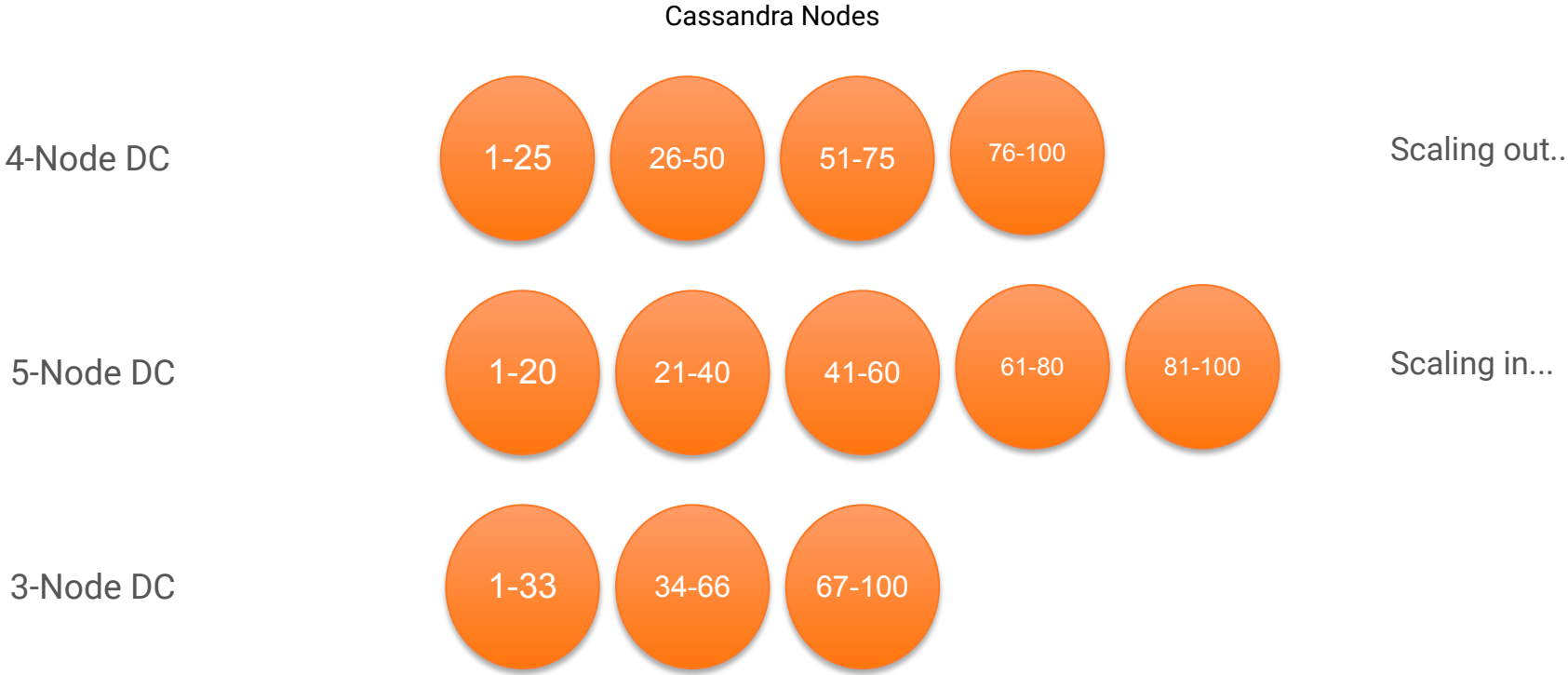
sky

DataStax Developers

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.



NOTICE: No downtime! All scaling operations are live!

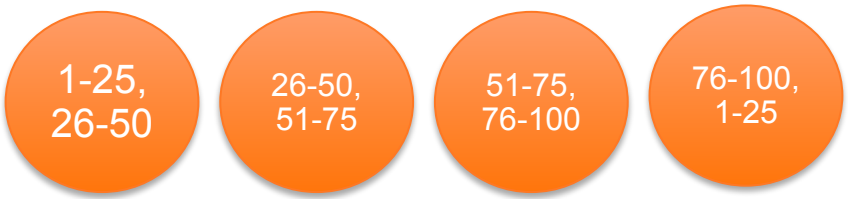
DataStax

sky

Token Ranges Recalculation

Replication Factor = 1

Replication Factor = 2

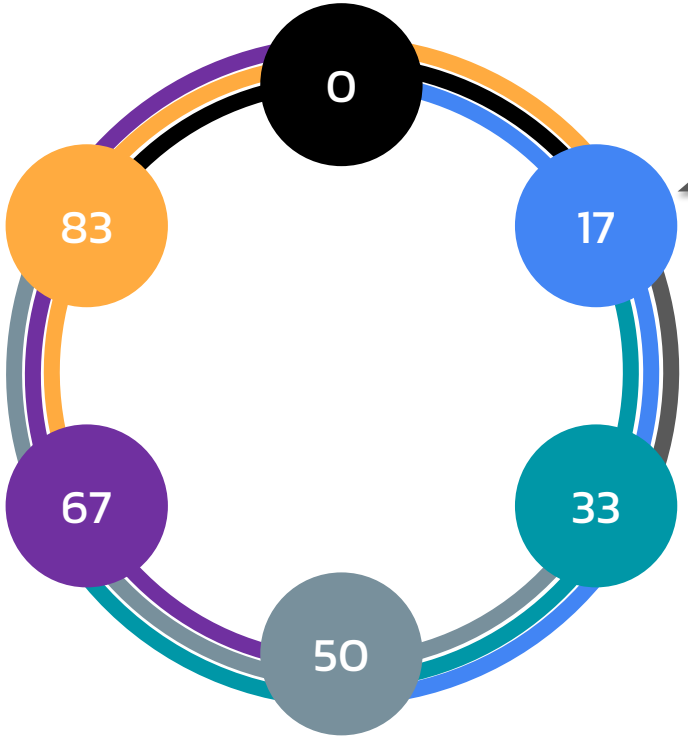


And so on…

forest	f001	92
forest	f002	88

59 (data)

RF = 3



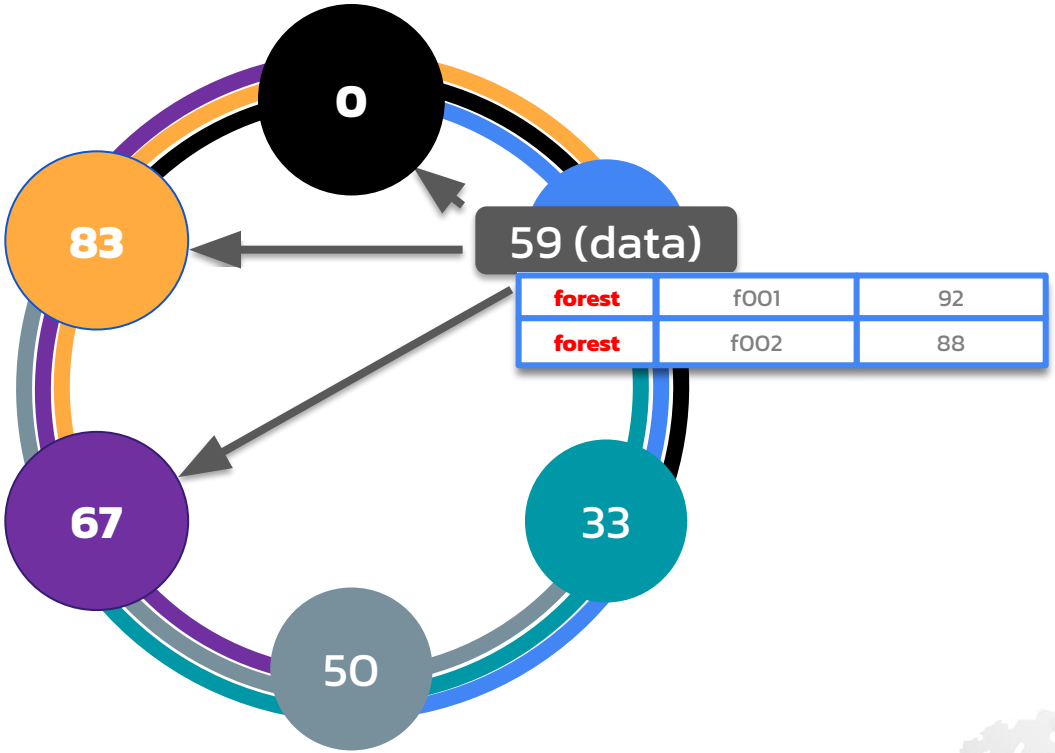
DataStax

sky

DataStax Developers

Partitioning + Replication

RF = 3



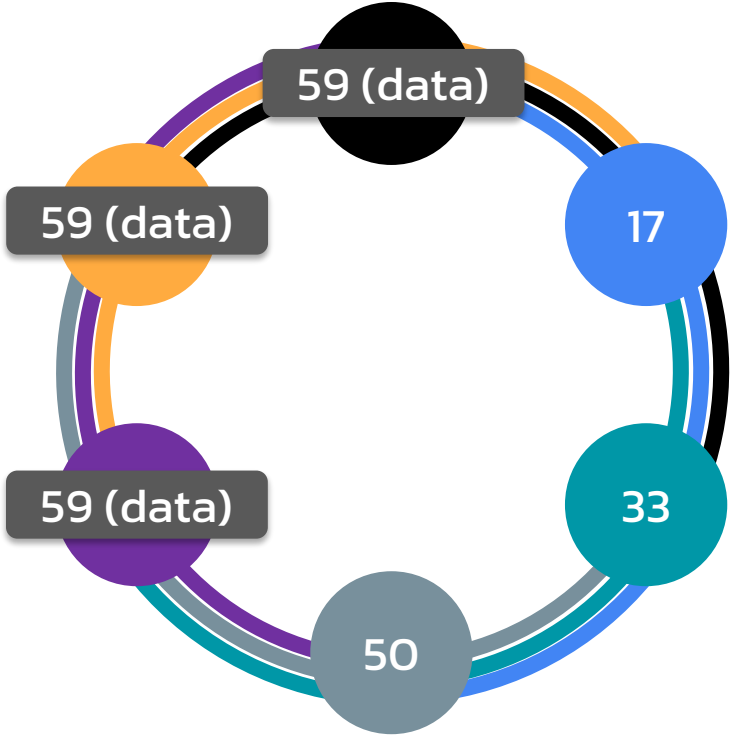
DataStax

sky

DataStax Developers

Partitioning + Replication

RF = 3

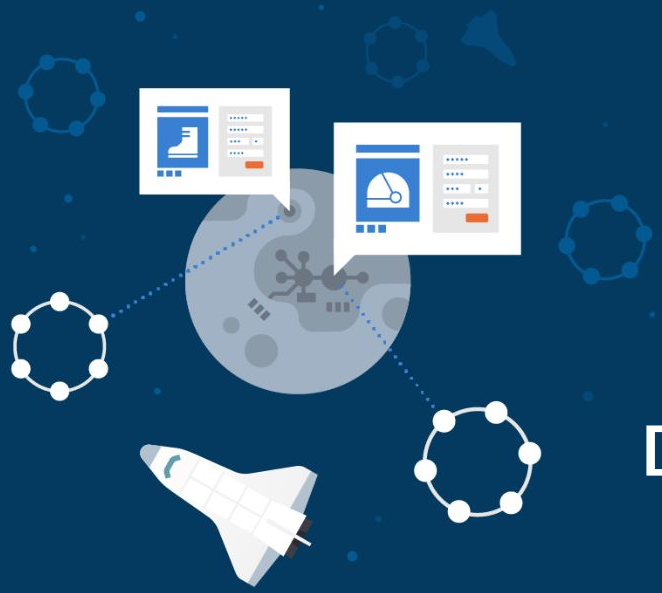


DataStax

sky

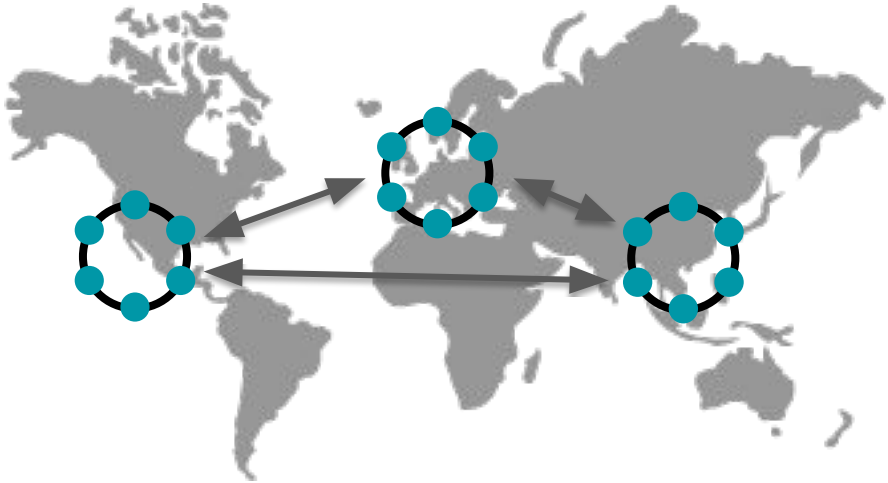
IMPORTANT

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.

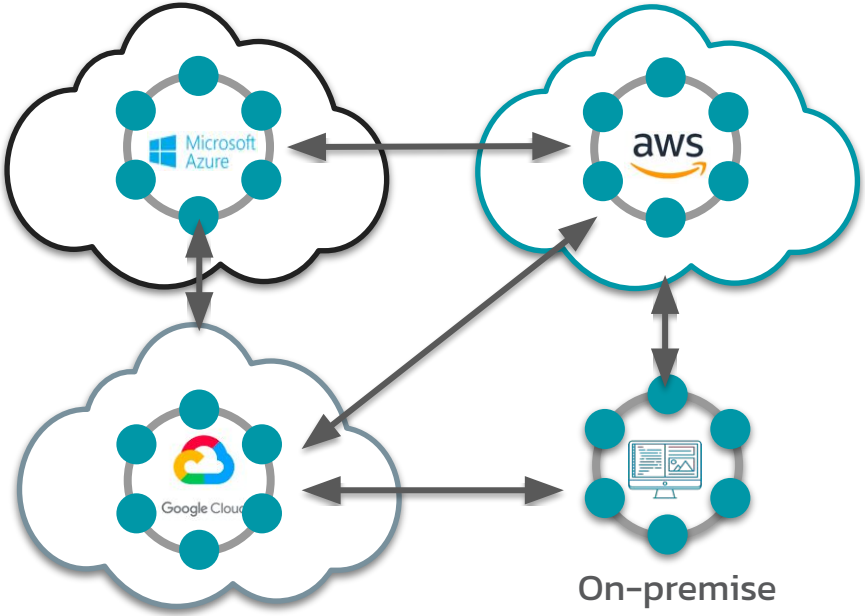


Data is GLOBALLY distributed

Geographical Distribution



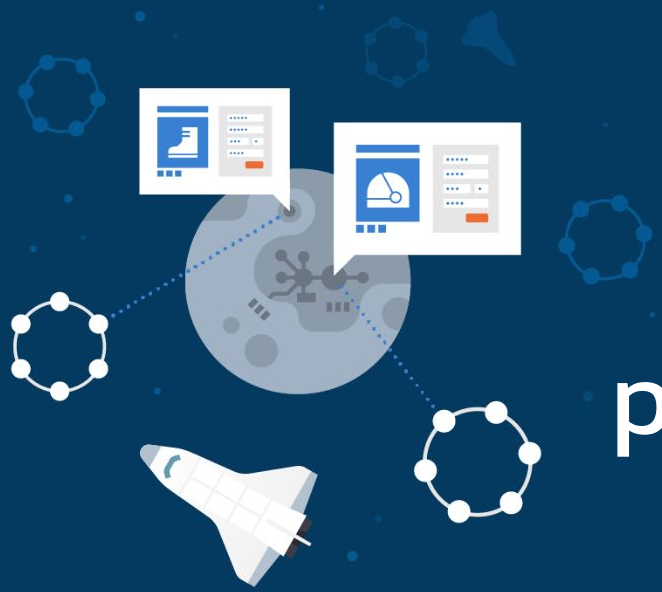
Hybrid-Cloud and Multi-Cloud



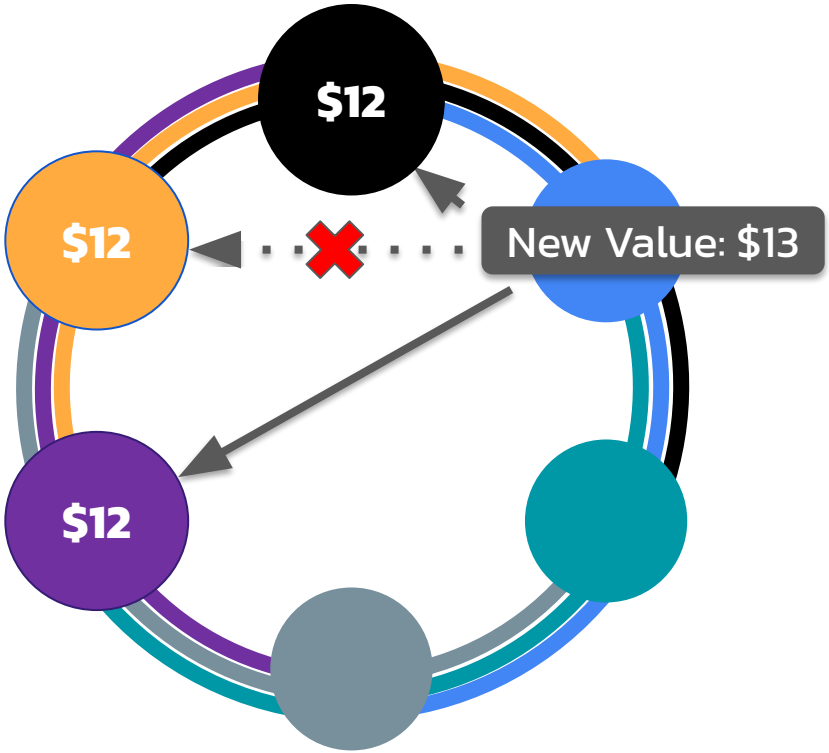
DataStax

sky

Data is globally distributed



What is the biggest problem of replication?

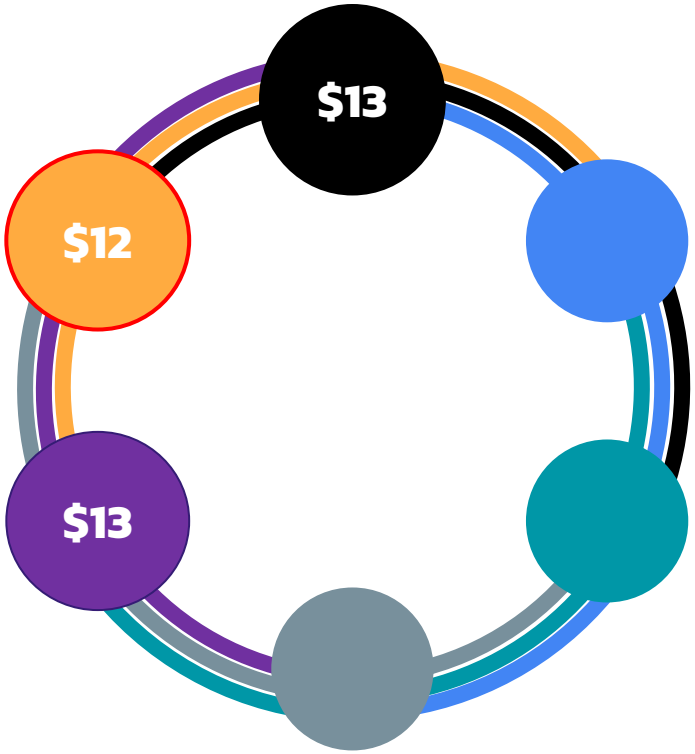


DataStax

sky

DataStax Developers

Potential Inconsistency

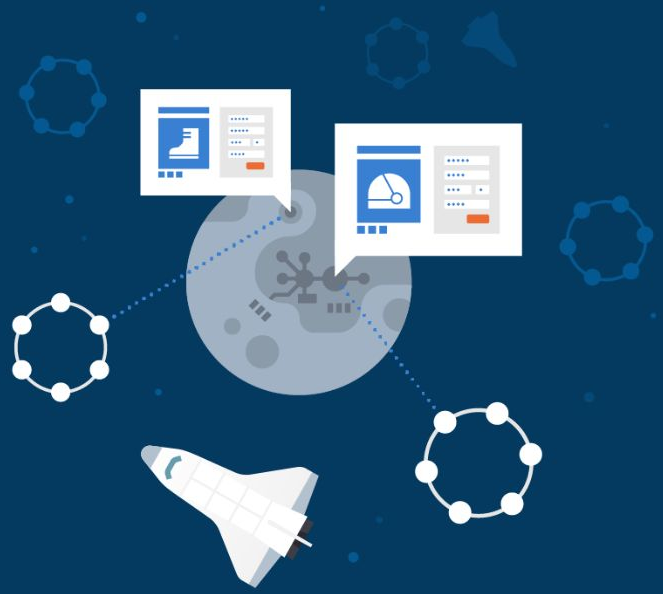


DataStax

sky

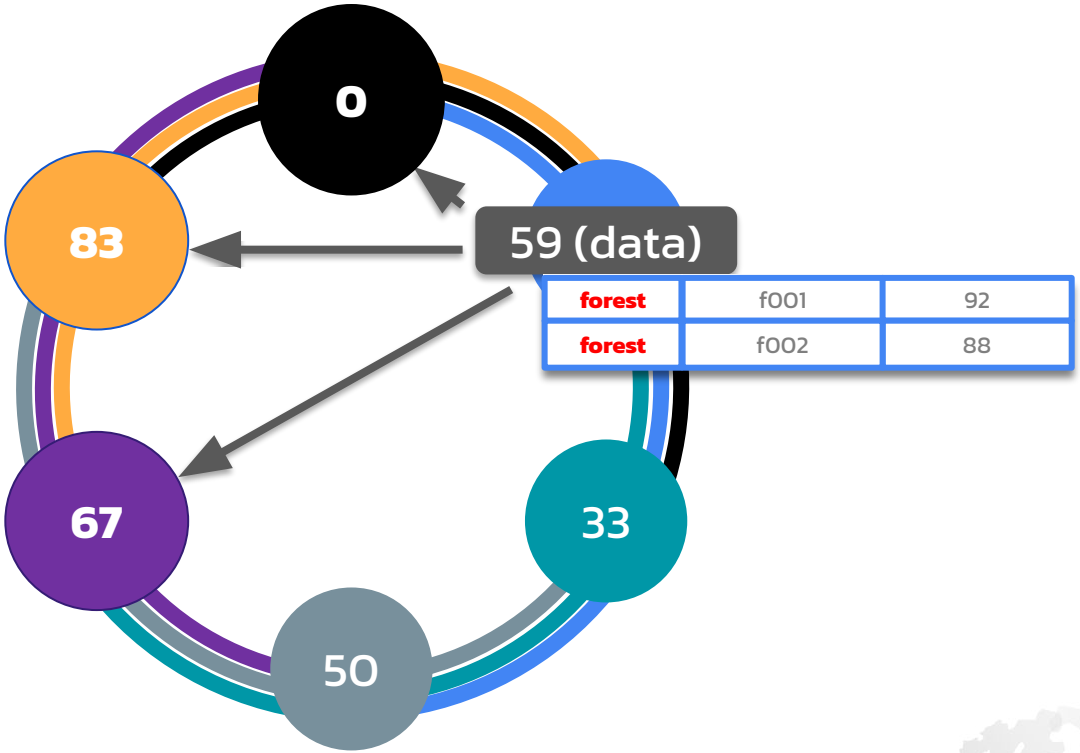
DataStax Developers

Potential Inconsistency



Cassandra Layered Self-Defence

RF = 3



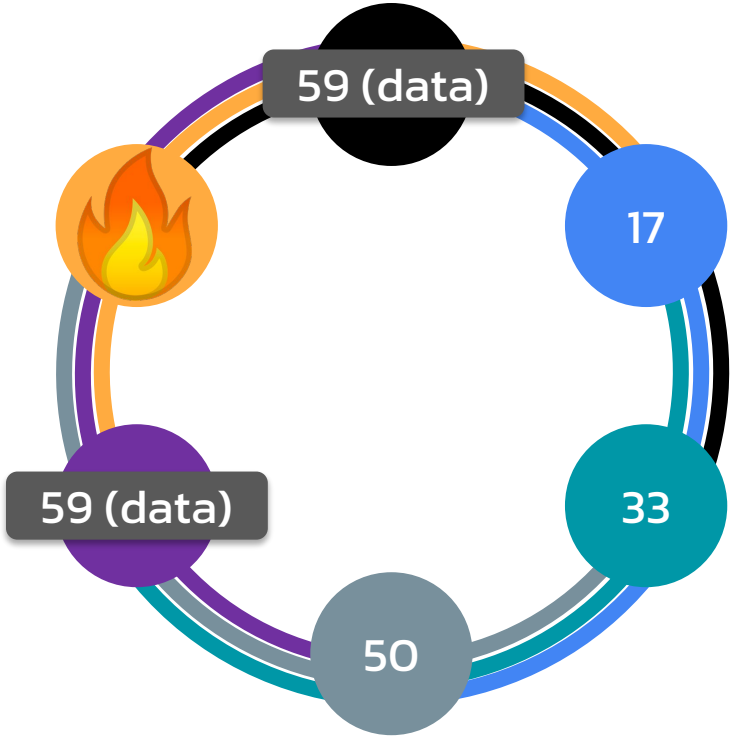
DataStax

sky

DataStax Developers

Hinted Hand-offs

RF = 3



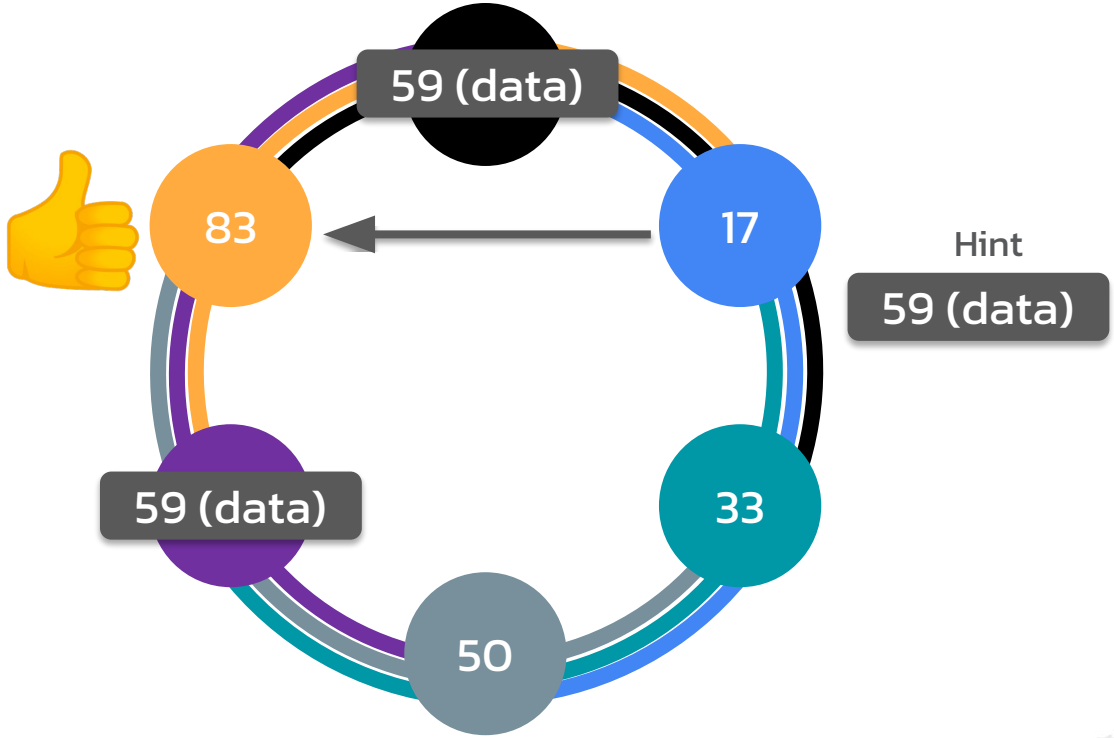
Hint

59 (data)

DataStax

sky

Hinted Hand-offs



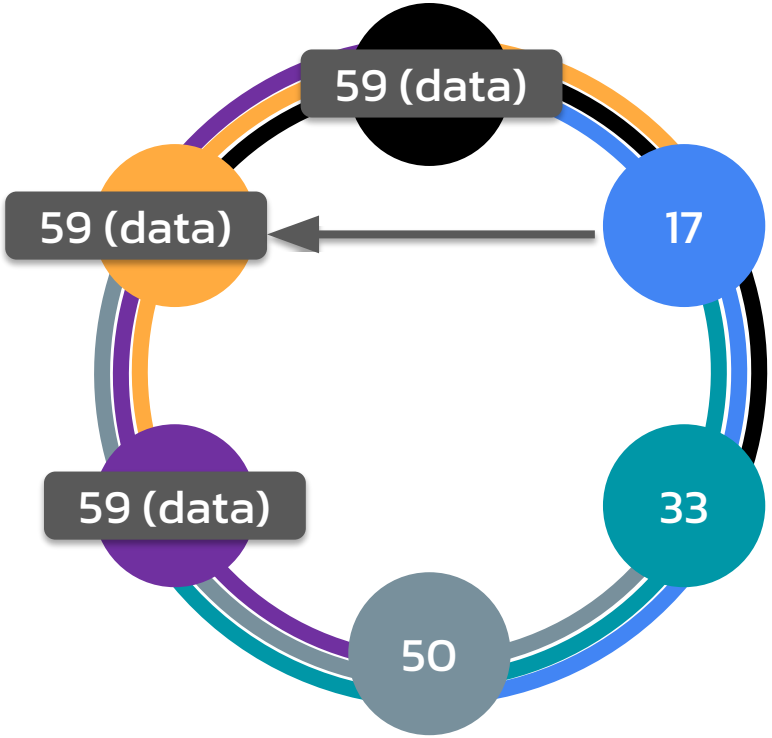
DataStax

sky

DataStax Developers

Hinted Hand-offs

RF = 3



DataStax

sky

DataStax Developers

Hinted Hand-offs

01

Why NoSQL ?

02

Introduction to
Apache Cassandra™

03

Distributed Architecture
The CAP Theorem

04

Tables
and Partitions

05

The Art of
Data Modelling

06

What's next?
Quiz, Homework, Next week

DataStax

sky

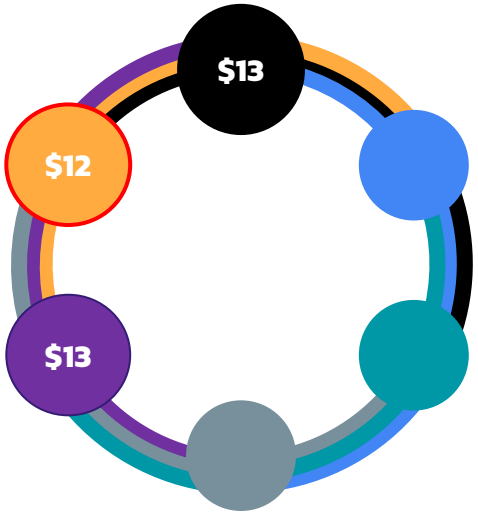
Agenda

CAP Theorem operates three features:

1. Availability
2. Consistency
3. Partition Tolerance

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**.

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**.



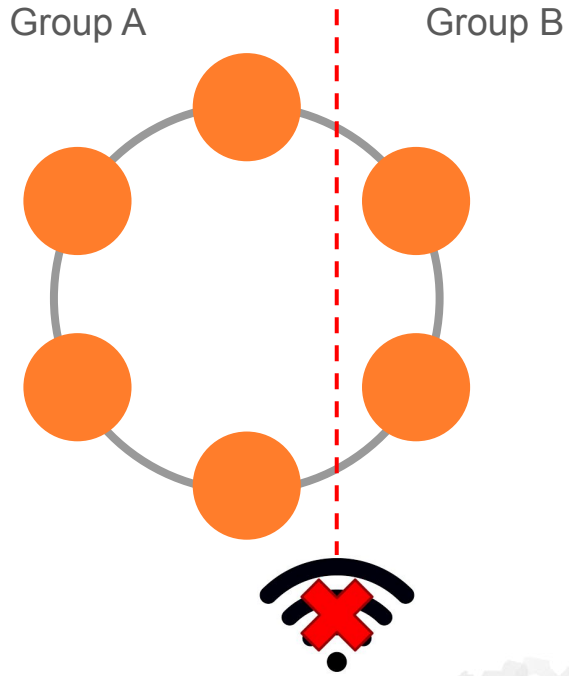
DataStax

sky

DataStax Developers

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.

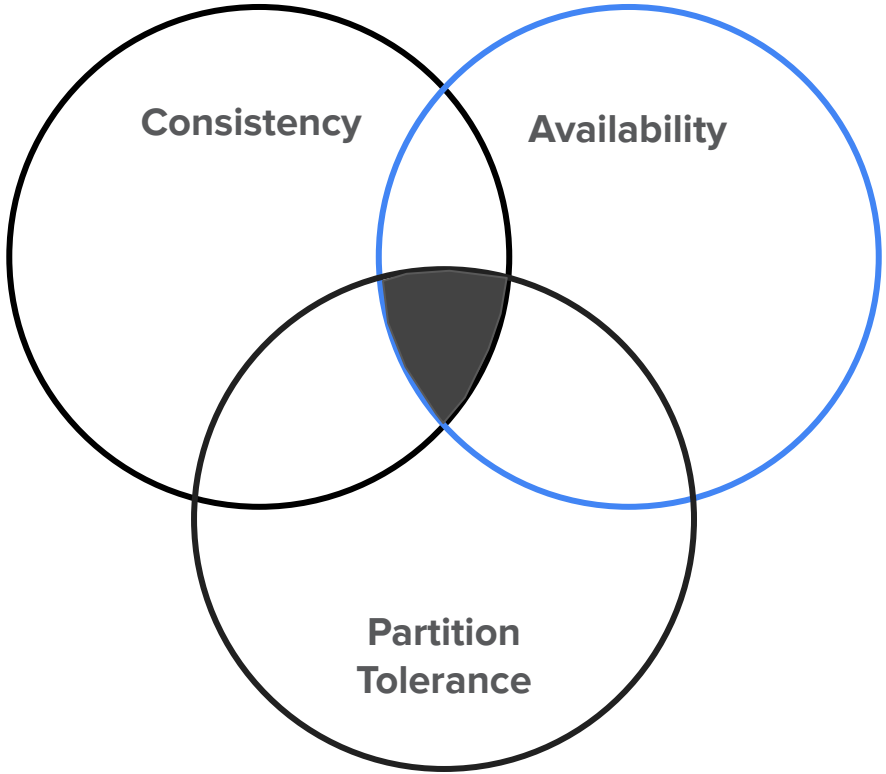


DataStax

sky

CAP Theorem: Partition Tolerance

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



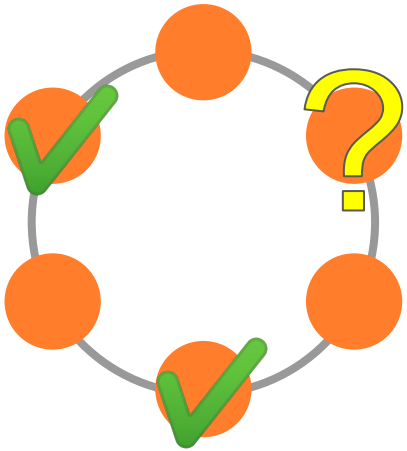
DataStax

sky

CAP Theorem

Cassandra is [configurably 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);
```



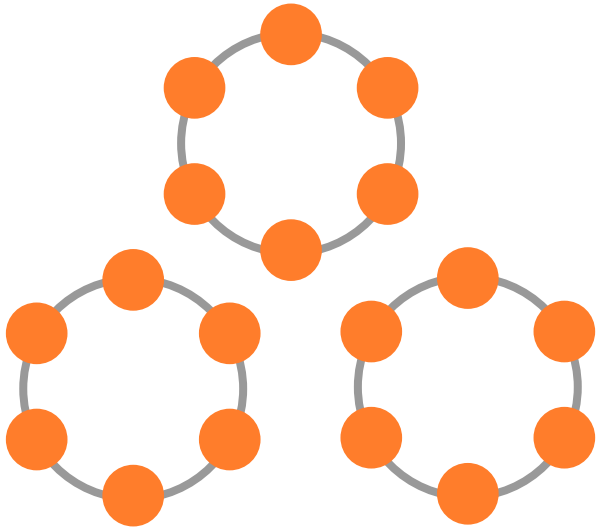
```
cqlsh> CONSISTENCY

Current consistency level is QUORUM.

cqlsh> CONSISTENCY ALL

Consistency level set to ALL.
```

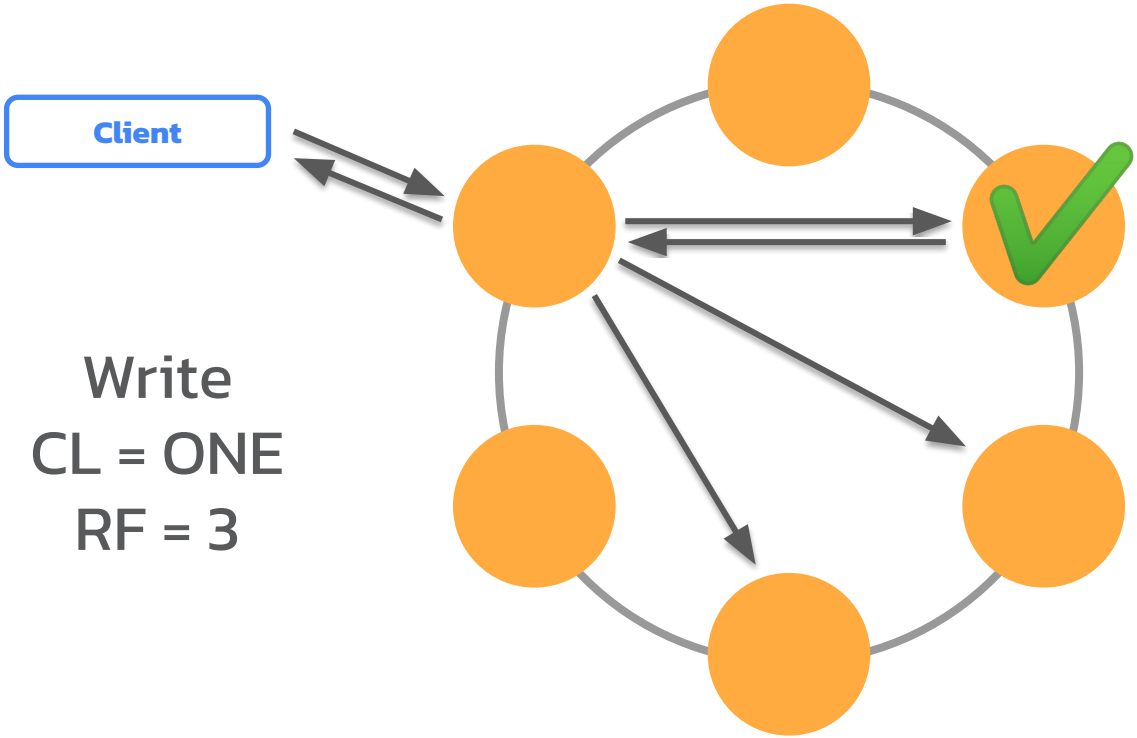
DataStax

sky

Is Cassandra AP or CP?

- ANY
- ONE
- LOCAL_ONE
- TWO, THREE
- QUORUM
- LOCAL_QUORUM
- EACH_QUORUM
- ALL

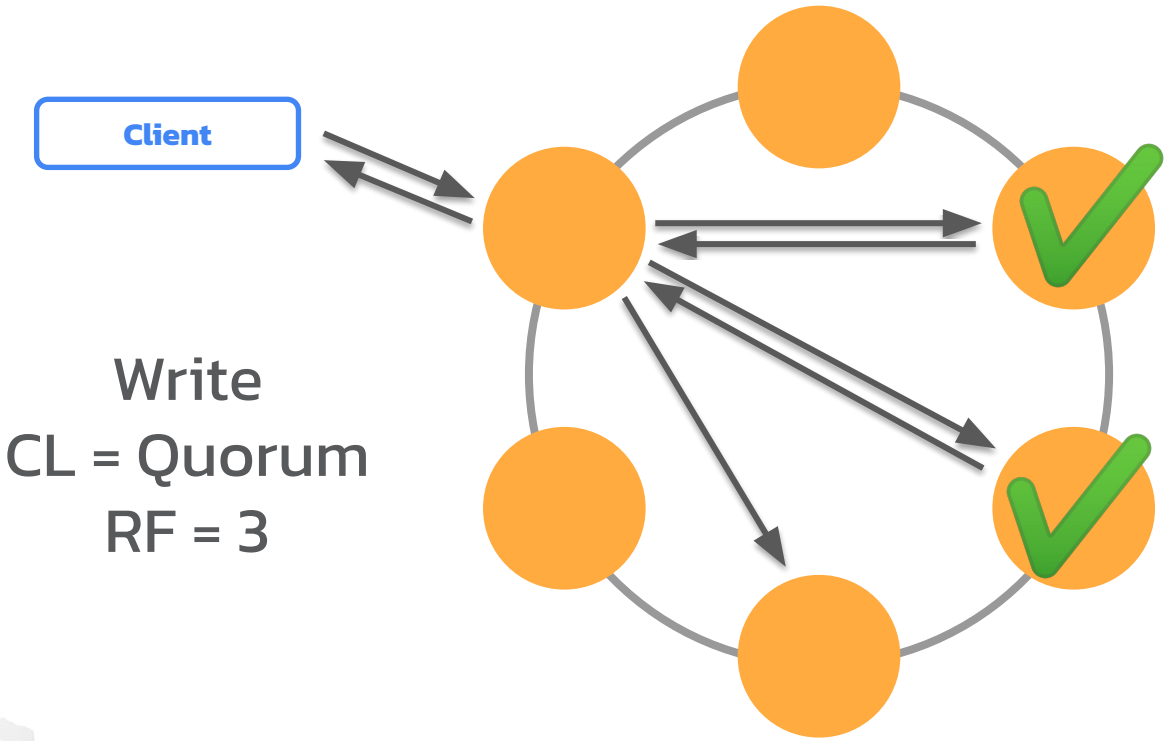




Write
CL = ONE
RF = 3

DataStax

Consistency Level One

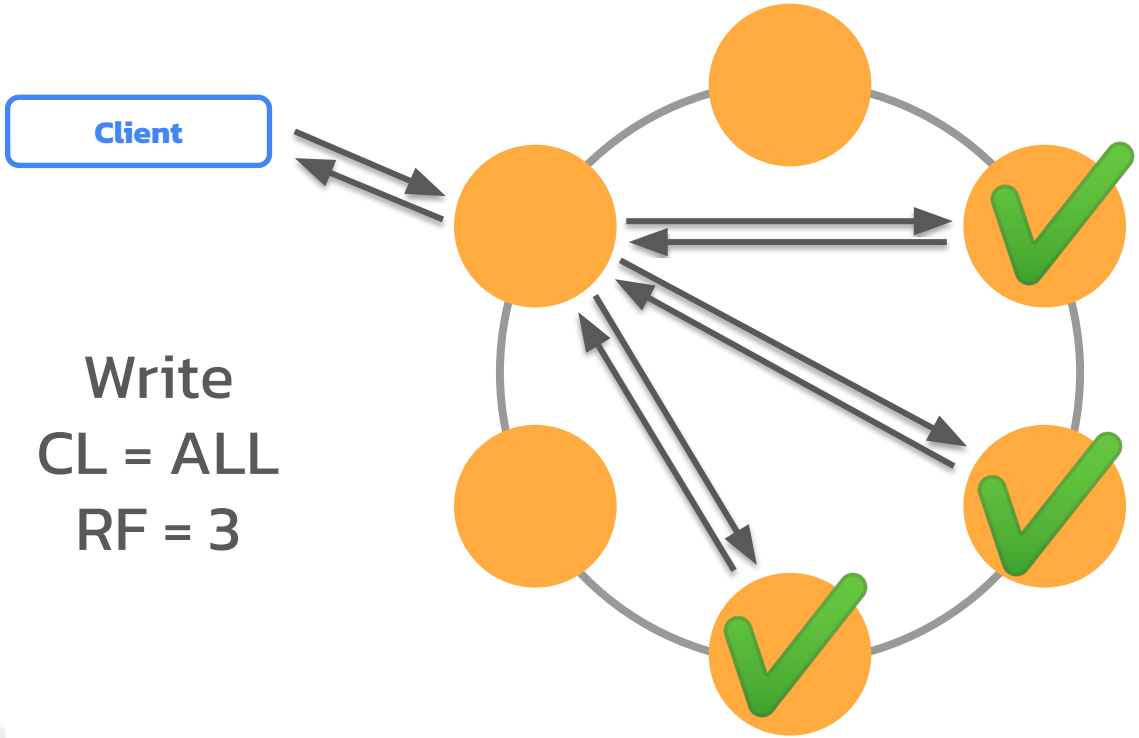


DataStax

sky

DataStax Developers

Consistency Level Quorum



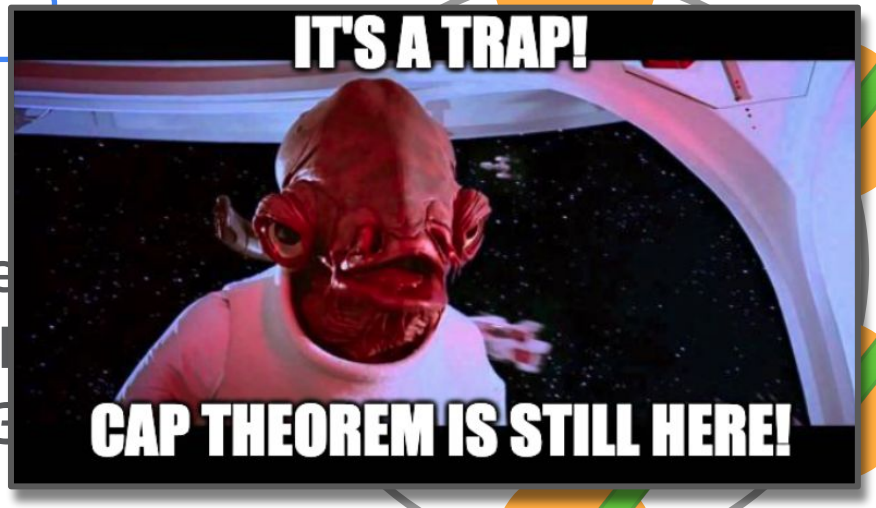
DataStax

sky

Consistency Level ALL

Client

Write
CL = A
RF = 3

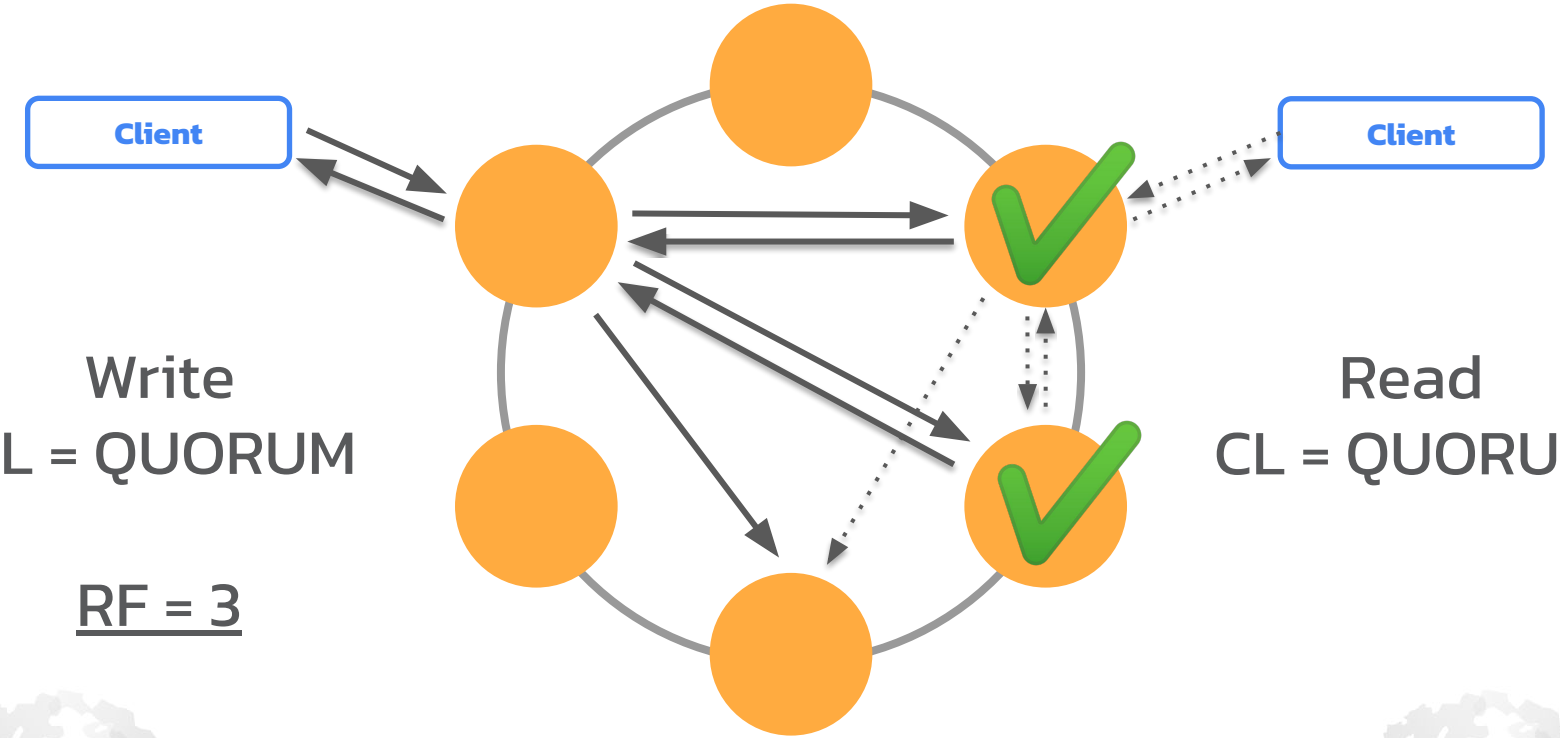


DataStax

sky

Consistency Level ALL

CL Write + CL Read > RF → Immediate Consistency



DataStax

sky

Immediate Consistency – The Right Way

01

Why NoSQL ?

02

Introduction to
Apache Cassandra™

03

Distributed Architecture
The CAP Theorem

04

Tables
and Partitions

05

The Art of
Data Modelling

06

What's next?
Quiz, Homework, Next week

DataStax

sky

Agenda

Data Structure: a Cell

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

f001

Data Structure: a Row

A single, structured data item in a table.

forest	**f001**	**92**

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.

forest	**f001**	**92**
forest	**f002**	**88**
forest	**f003**	**90**
sea	**s001**	**45**

Data Structure: a Table

A group of columns and rows storing partitions.

network	sensor	temperature
forest	f001	92
forest	f002	88
volcano	v001	210
sea	s001	45
sea	s002	50
home	h001	72
car	c001	69
car	c002	70
dog	d001	40
road	r001	105
road	r002	110
ice	i001	35

DataStax

Data Structure: a Table

Lab 2

Create Tables

http://github.com/datastaxdevs/workshop-cassandra-fundamentals

01

Why NoSQL ?

02

Introduction to
Apache Cassandra™

03

Distributed Architecture
The CAP Theorem

04

Tables
and Partitions

05

The Art of
Data Modelling

06

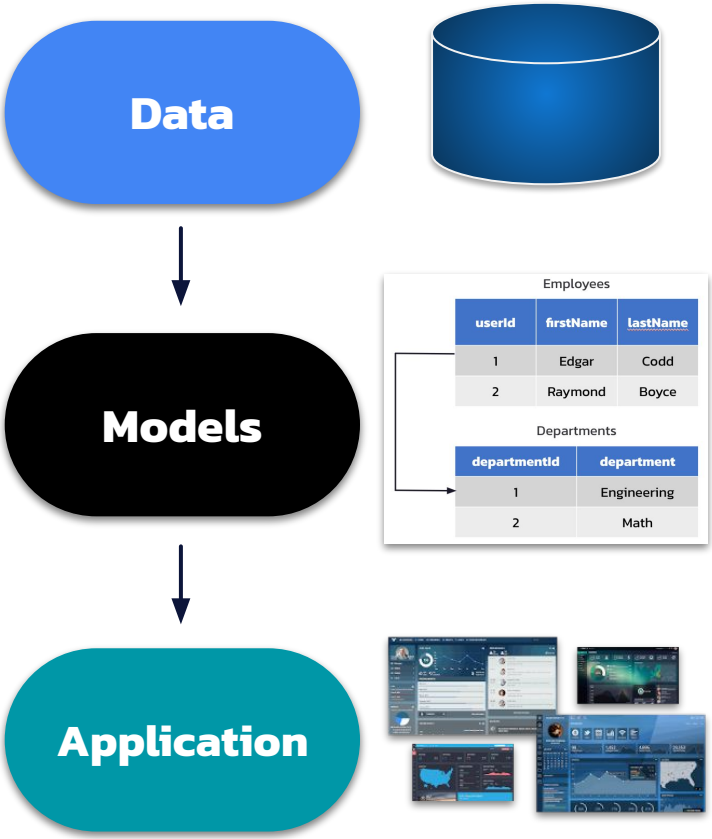
What's next?
Quiz, Homework, Next week

DataStax

sky

Agenda

1. Analyze raw data
2. Identify entities, their properties and relations
3. Design tables, using **normalization** and foreign keys.
4. Use JOIN when doing queries to join normalized data from multiple tables

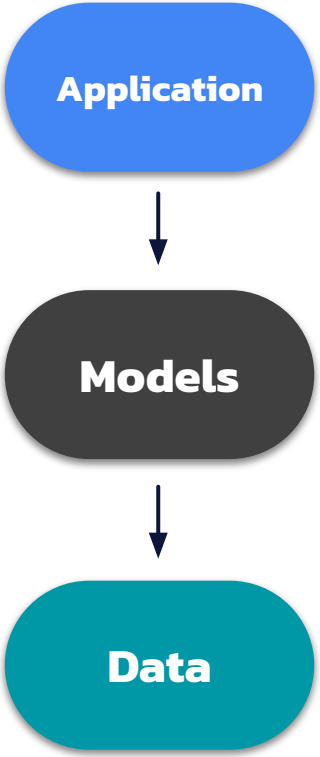


DataStax

sky

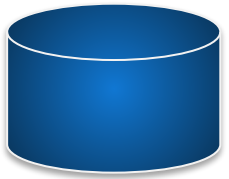
Relational Data Modelling

1. Analyze user behaviour (customer first!)
2. Identify workflows, their dependencies and needs
3. Define Queries to fulfill these workflows
4. Knowing the queries, design tables, using **denormalization**.
5. Use BATCH when inserting or updating denormalized data of multiple tables



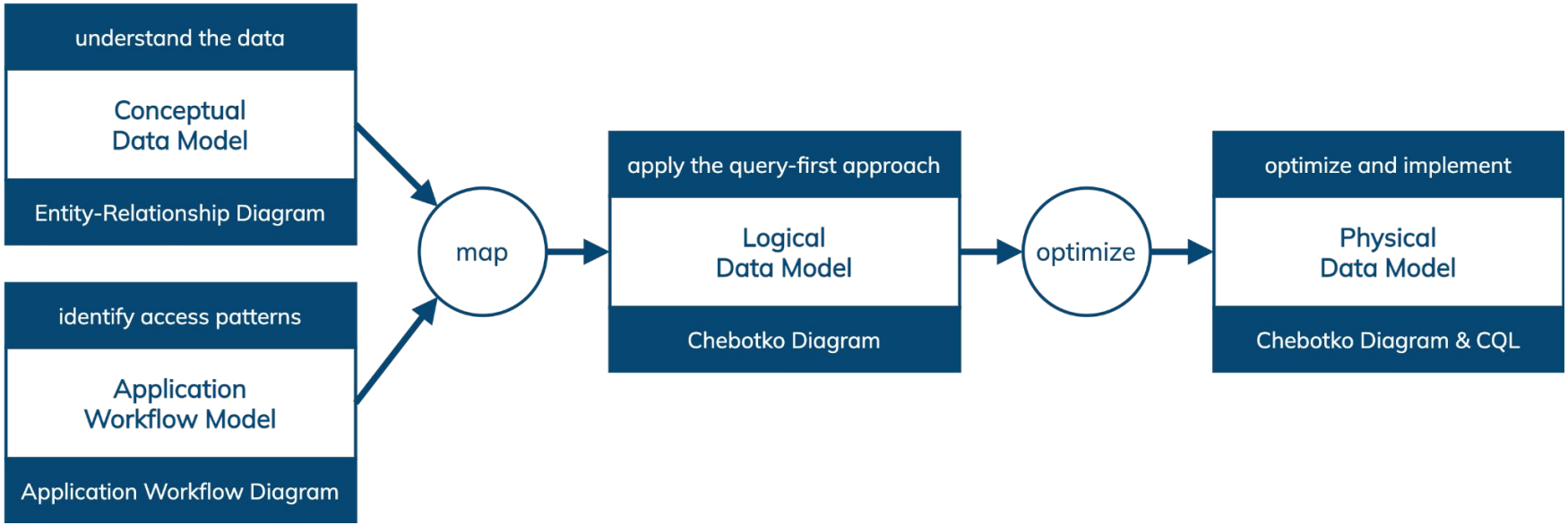
Employees

userId	firstName	lastName	department
1	Edgar	Codd	Engineering
2	Raymond	Boyce	Math
3	Sage	Lahja	Math
4	Juniper	Jones	Botany



DataStax

sky

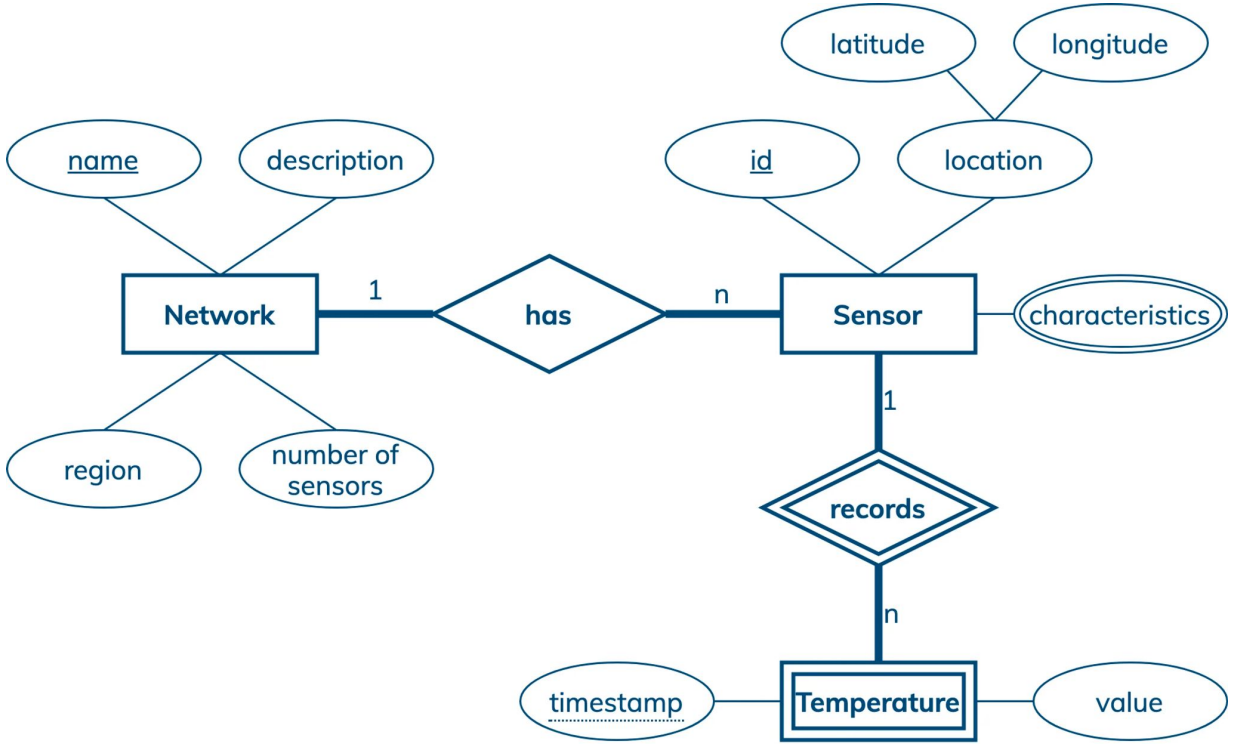


very often, 1 query = 1 table

DataStax

sky

Data Modelling Methodology

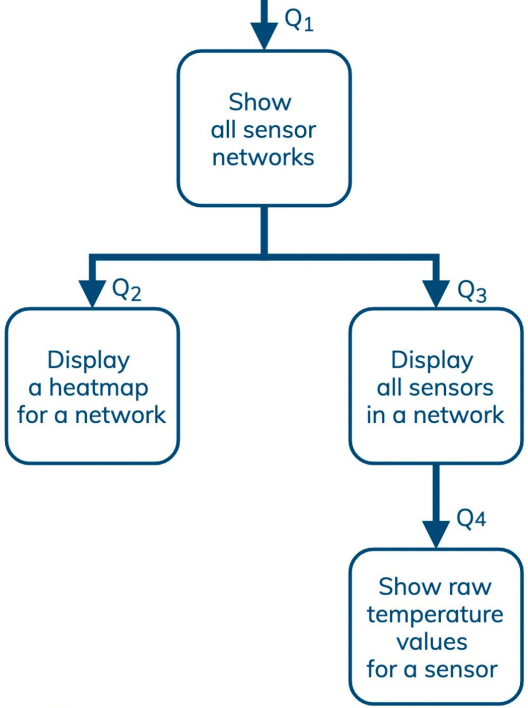


Entity-Relationship Diagram

DataStax

sky

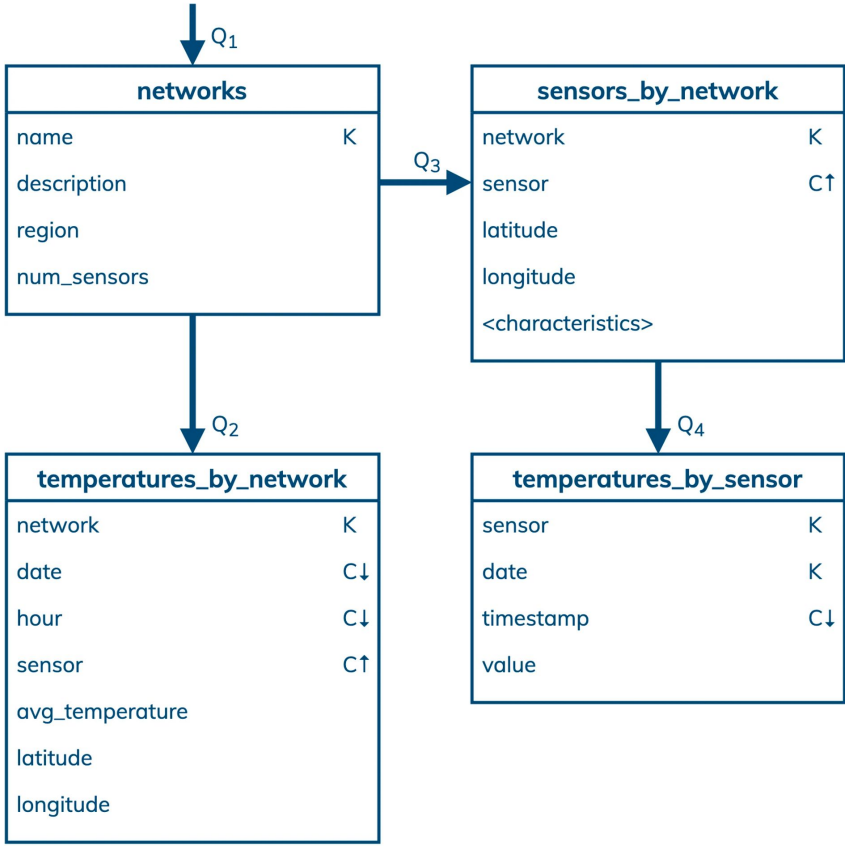
Conceptual Data Model



Data access patterns

Q_1: Find information about all networks; order by name (asc)
Q_2: Find hourly average temperatures for every sensor in a specified network for a given date range; order by date (desc) and hour (desc)
Q_3: Find information about all sensors in a specified network
Q_4: Find raw measurements for a particular sensor on a specified date; order by timestamp (desc)

Application Workflow

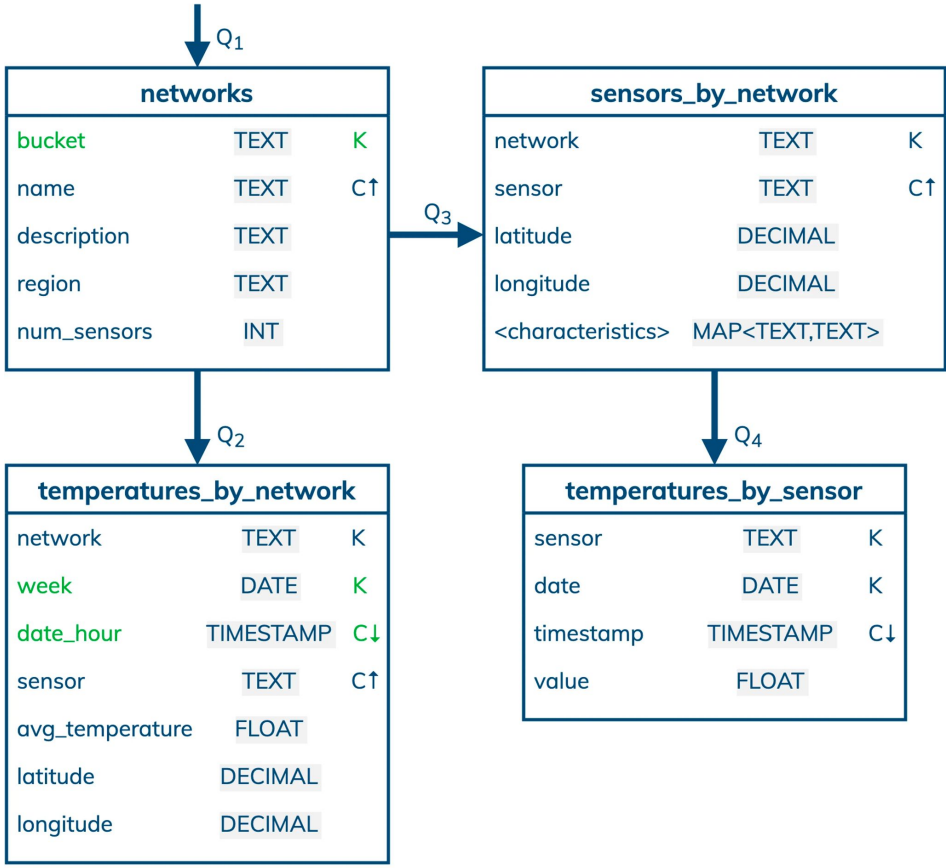


Chebotko Diagram

DataStax

sky

Mapping (1 / 2)



DataStax

Chebotko Diagram

Optimize

DataStax Developers

sky

```
CREATE TABLE networks (
  bucket TEXT,
  name TEXT,
  description TEXT,
  region TEXT,
  num_sensors INT,
  PRIMARY KEY ((bucket),name)
);
```

Q3

```
CREATE TABLE sensors_by_network (
  network TEXT,
  sensor TEXT,
  latitude DECIMAL,
  longitude DECIMAL,
  characteristics MAP<TEXT,TEXT>,
  PRIMARY KEY ((network),sensor)
);
```

Q2

```
CREATE TABLE temperatures_by_network (
  network TEXT,
  week DATE,
  date_hour TIMESTAMP,
  sensor TEXT,
  avg_temperature FLOAT,
  latitude DECIMAL,
  longitude DECIMAL,
  PRIMARY KEY ((network,week),date_hour,sensor)
) WITH CLUSTERING ORDER BY (date_hour DESC, sensor ASC);
```

Q4

```
CREATE TABLE temperatures_by_sensor (
  sensor TEXT,
  date DATE,
  timestamp TIMESTAMP,
  value FLOAT,
  PRIMARY KEY ((sensor,date),timestamp)
) WITH CLUSTERING ORDER BY (timestamp DESC);
```

Lab 3

CRUD Operations

http://github.com/datastaxdevs/workshop-cassandra-fundamentals

01

Why NoSQL ?

02

Introduction to
Apache Cassandra™

03

Distributed Architecture
The CAP Theorem

04

Tables
and Partitions

05

The Art of
Data Modelling

06

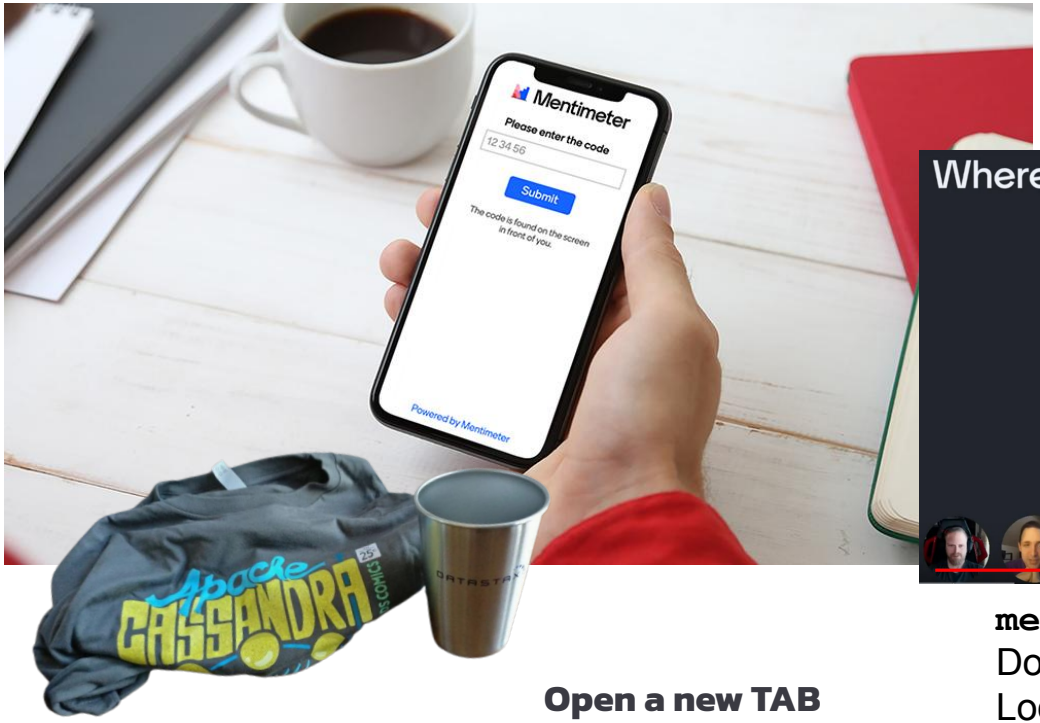
What's next?

Quiz, Homework, Next week

DataStax

sky

Agenda



Open a new TAB

CODE= XXXXX

`menti.com` ⇒ enter code
Don't answer in YT chat
Look at phone (not at YT)
Keep it open for later

DataStax

sky

"Menti" for survey and quiz !

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

** Include a screenshot of your Menti screen

DataStax

sky

Homework

!homework

Intro to Cassandra Homework

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

- Workshop materials: https://github.com/datastaxdevs/workshop-intro-to-cassandra
- Discord chat: https://dtsx.io/discord

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.

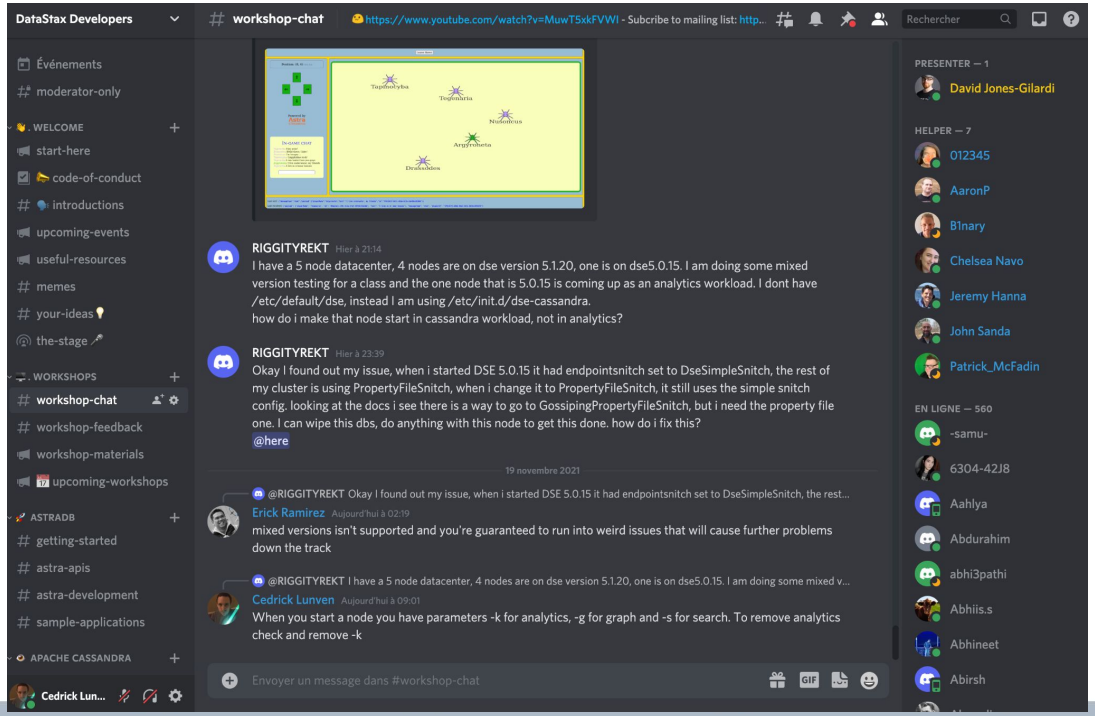
* Required

DataStax

sky

Join our 17k Discord Community

DataStax Developers



!discord

dtsx.io/discord

DataStax

DataStax Developers

Discord Community

Thank You!