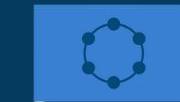


DataStax

# Developers

## NoSQL Data Modelling with Apache Cassandra™

Build API and Microservices for Apache Cassandra™



# Theme Colors



#3A3A42



FFFFFF



#1A1A30



#F8F9FA



#EB6C34



#4A9AB4



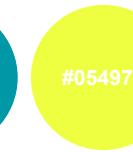
#998E5D



#3980D3



#055992



#054978



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



Indiana



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



E-commerce



Agility



Analytics



OPs



Backup recovery



Logic



AI-ML-1



AI-ML-2



Migrate-1



Migrate-2



Users/  
customers



Leader



Cluster

# Director of Developer Relations



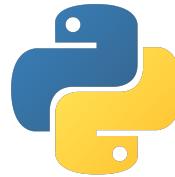
- Trainer
- Public Speaker
- Developers Support
- Developer Applications
- Developer Tooling
  
- Creator of ff4j ([ff4j.org](http://ff4j.org))
- Maintainer for 8 years+
  
- Happy developer for 14 years
- Spring Petclinic Reactive & Starters
- Implementing APIs for 8 years



Cédrick Lunven



# Developer Advocate



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



@stefano-lottini



@hemidactylus



@rsprrs

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



**Stefano Lottini**



## Developer Advocate Lead

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



@aleks-volochnev  
@HadesArchitect

- IT Exorcist
- Apache Cassandra™ certified
- Cloud Architect certified



Aleksandr Volochnev





Cedrick  
Lunven

Aleksandr  
Volochnev

Jack  
Fryer

Kirsten  
Hunter

Stefano  
Lottini

David  
Gilardi

Ryan  
Welford

Rags  
Srinivas

Sonia  
Siganporia

R

S



DataStax Developers Crew

# 01



**Bootcamp 2022**  
**Housekeeping**  
**Reminders**

# 02

**Microservices**  
**Why with Apache Cassandra ?**

# 03

**APIs**  
**Rest, GraphQL, gRPC**

# 04

**Todo Application**  
**TodoMVC, TodoBackend**

# 05

**Code Build Package**  
**Building efficient Data Model**

# 06

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



**Agenda**



## WEEK 1

January 5th - January 11th



## WEEK 3

January 19th-January 25th



## WEEK 2

January 12th - January 18th



## WEEK 4

January 26th-February 1st



**THIS SESSION CAN BE FOLLOWED  
WITHOUT ATTENDING PREVIOUS**



**Bootcamp 2022**

# todos

- ⌄ What needs to be done?
  - ✓ Week1, Learn about Apache Cassandra
  - ✓ Week2, Apache Cassandra Data Model
  - ✓ How to Connect to Cassandra

1 item left

All Active Completed

[Clear completed \(2\)](#)

Double-click to edit a todo

Written by [Addy Osmani](#), modified by [Pete Hodgson](#) to be integrated with a [TodoBackend API](#).

Part of [TodoMVC](#) & [TodoBackend](#)



**TodoApplication**



1

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

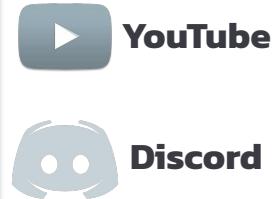
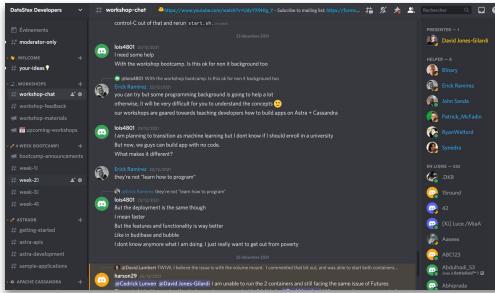


4-week bootcamp Housekeeping

**Livestream:** youtube.com/DataStaxDevs

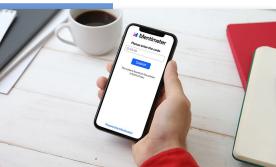
**Questions:** <https://dtsx.io/discord>

#### Agenda



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

How much experience do you have with the Spring Framework ?



Mentimeter

- 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 shows the code editor with "StargateDemoApplication.java" containing Spring Boot code. Below the code editor are logos for npm, node.js, and Maven. At the bottom is the Terminal tab with the command "gitpod /workspace/workspace-spring-stargate \$".

Gitpod

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 life 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 adjust 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 a GitHub repository named "DataStax-Examples / todo-astra-jamstack-netlify". The repository has 177 commits ahead of the master branch. The code editor shows a file named "functions/rfa-todos". The right sidebar displays the "astra.datastax.com/register" page, which includes sections for "Readme", "Releases", and "Contributors".

GitHub

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 "Cassandra Fundamentals" page under the "Learning Series Topics". It features a "GET STARTED" button and a list of 11 topics: 01 Introduction to Apache Cassandra™, 02 Cassandra Query Language, 03 Keyspaces and Data Replication Strategies, 04 Tables with Single-Row Partitions, 05 Tables with Multi-Row Partitions, 06 Queries, 07 Advanced Data Types, 08 Tunable Consistency and Consistency Levels, 09 Linearizable Consistency and Lightweight Transactions, 10 Readme, and 11 Releases.

The screenshot shows a GitHub repository named "DataStax-Examples / todo-astra-jamstack-netlify". It displays a list of branches (master, 177, 178), pull requests, actions, projects, wiki, security, insights, and settings. The master branch has 177 commits ahead of tjaiki/master. A green "Group" button is visible. The repository details show it's a Todo list JAMstack app example with React.js + DataStax Astra DB. It includes links to astria.datastax.com/register, netlify.toml, and building-sample-apps. There are no releases published.

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** Submit the homework (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".

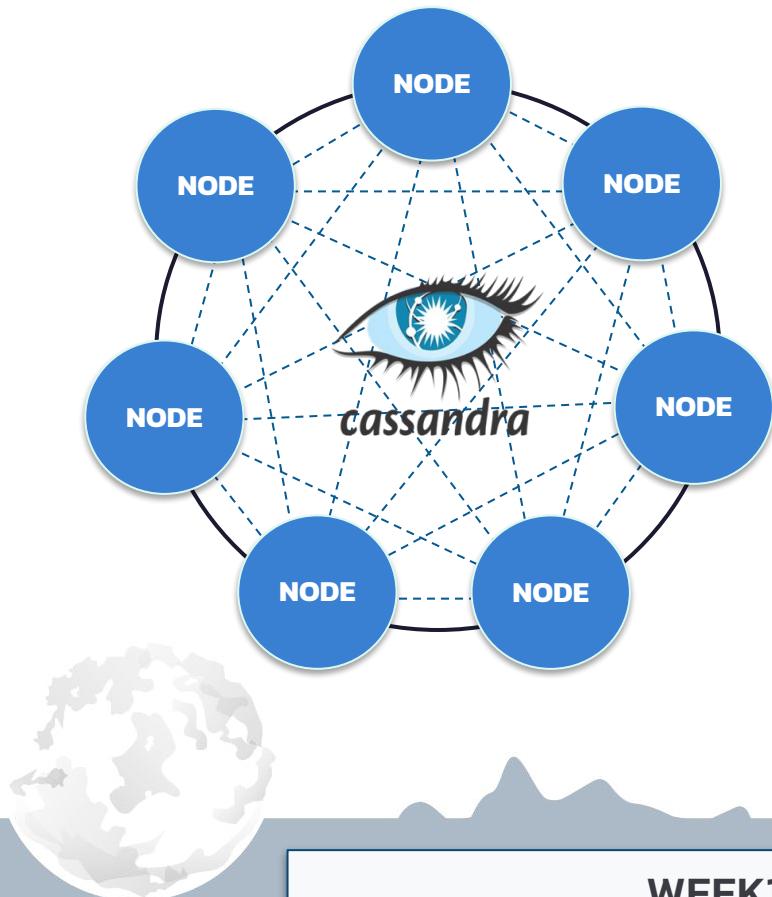
Go to [www.menti.com](http://www.menti.com) and use the code 3491 9972

## Leaderboard

User ID	User Name	Profile Icon
4821 p	spanda	Avatar of a green sphere
4820 p	Agent X9	Avatar of a black mask with red eyes
4775 p	Sam	Avatar of a cat-like face
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 emoji
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".

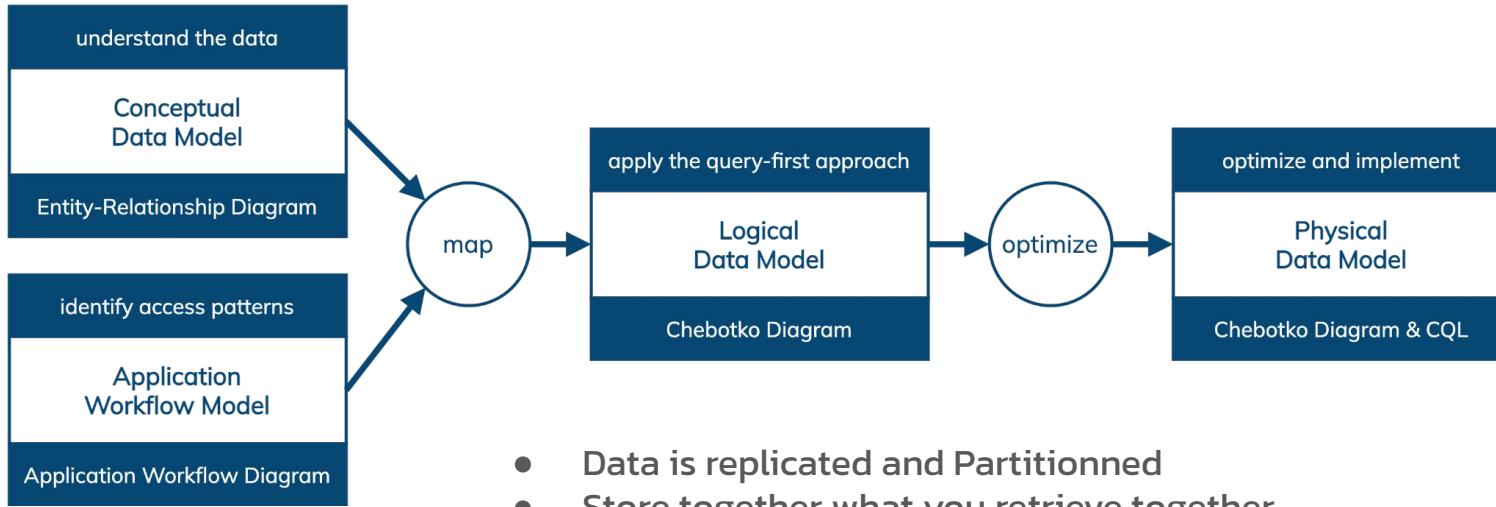
# Introduction to Apache Cassandra



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

WEEK1 in a Nutshell

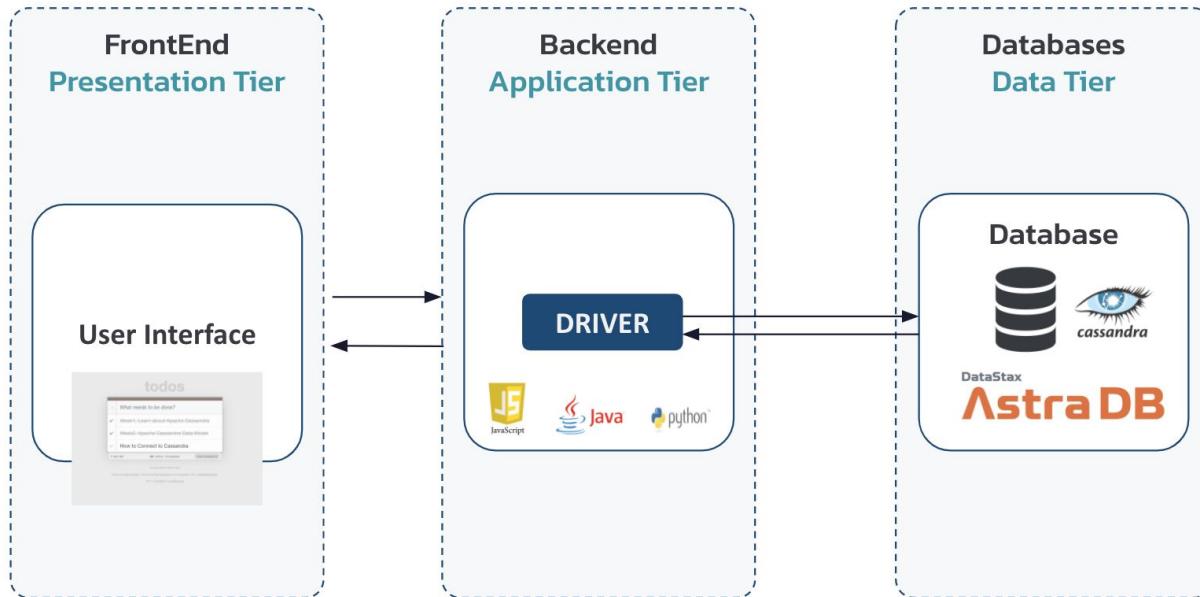
# Data Modeling with Apache Cassandra



- Data is replicated and Partitionned
- Store together what you retrieve together
- Avoid big or hot partitions

Week2 In a Nutshell

# Application Development with Apache Cassandra



Week3 in a nutshell





And Today for your pleasure

# 01



**Bootcamp 2022**  
**Housekeeping**  
**Reminders**

# 02

**Microservices**  
**Why with Apache Cassandra ?**

# 03

**APIs**  
**Rest, GraphQL, gRPC**

# 04

**Todo Application**  
**TodoMVC, TodoBackend**

# 05

**Code Build Package**  
**Building efficient Data Model**

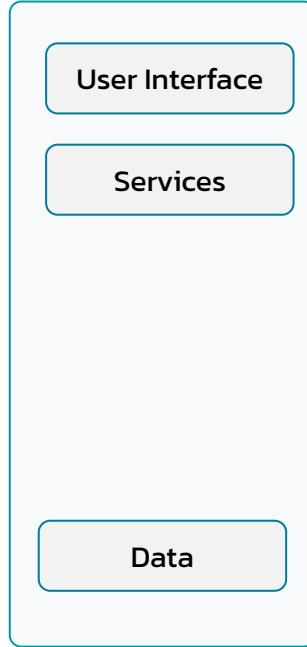
# 06

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

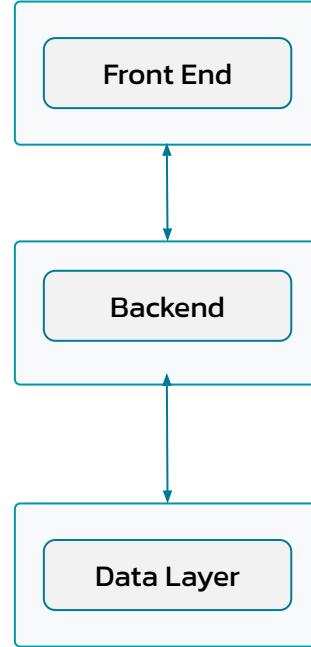


**Agenda**

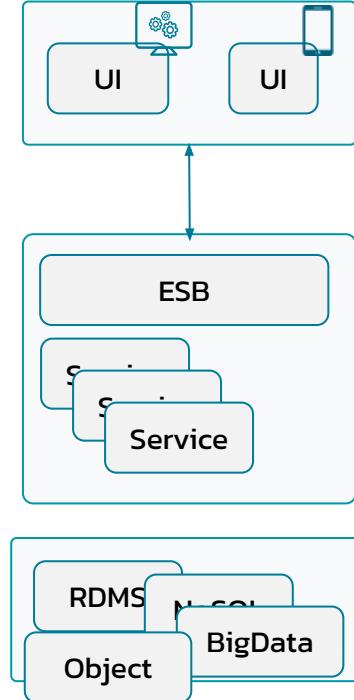
## Monolith



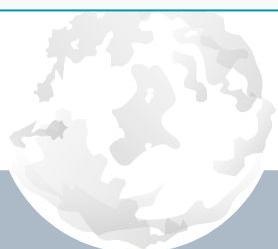
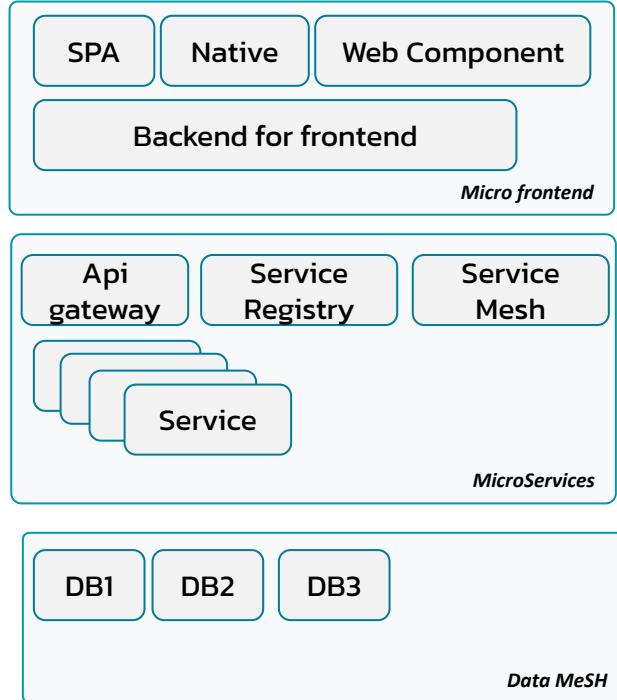
## Multi Tiers



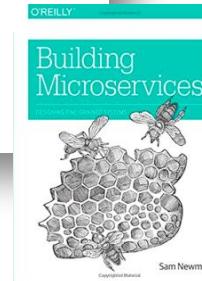
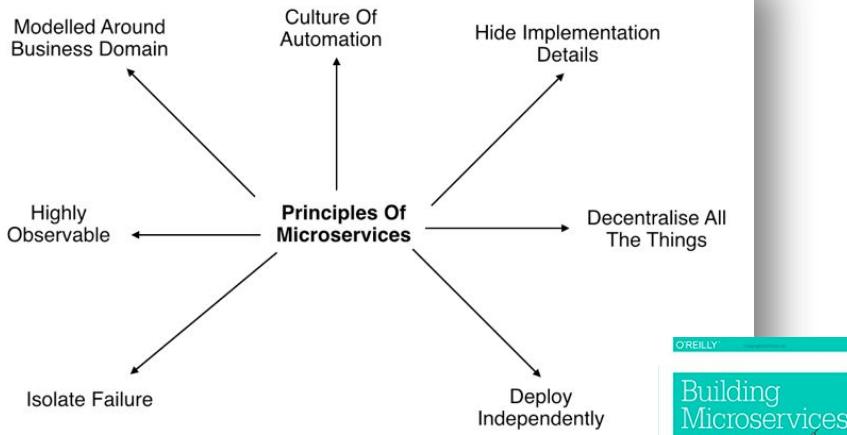
## SOA



## Microservices



To the Microservices Architecture



- ✓ Organized around Business Capabilities
- ✓ Products not Projects
- ✓ Smart endpoints and dumb pipes
- ✓ Decentralized Governance
- ✓ Decentralized Data Management
- ✓ Infrastructure Automation
- ✓ Design for failure
- ✓ Evolutionary Design

Martin Fowler

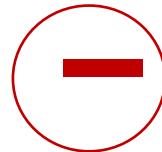


## ADVANTAGES



- Reduce Cost (Scaling, Design)
- Reduce Risk (resilience)
- Increase Release Speed
- Enable Visibility (security, monitoring)

## DISADVANTAGES

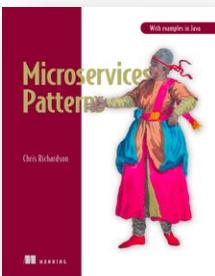
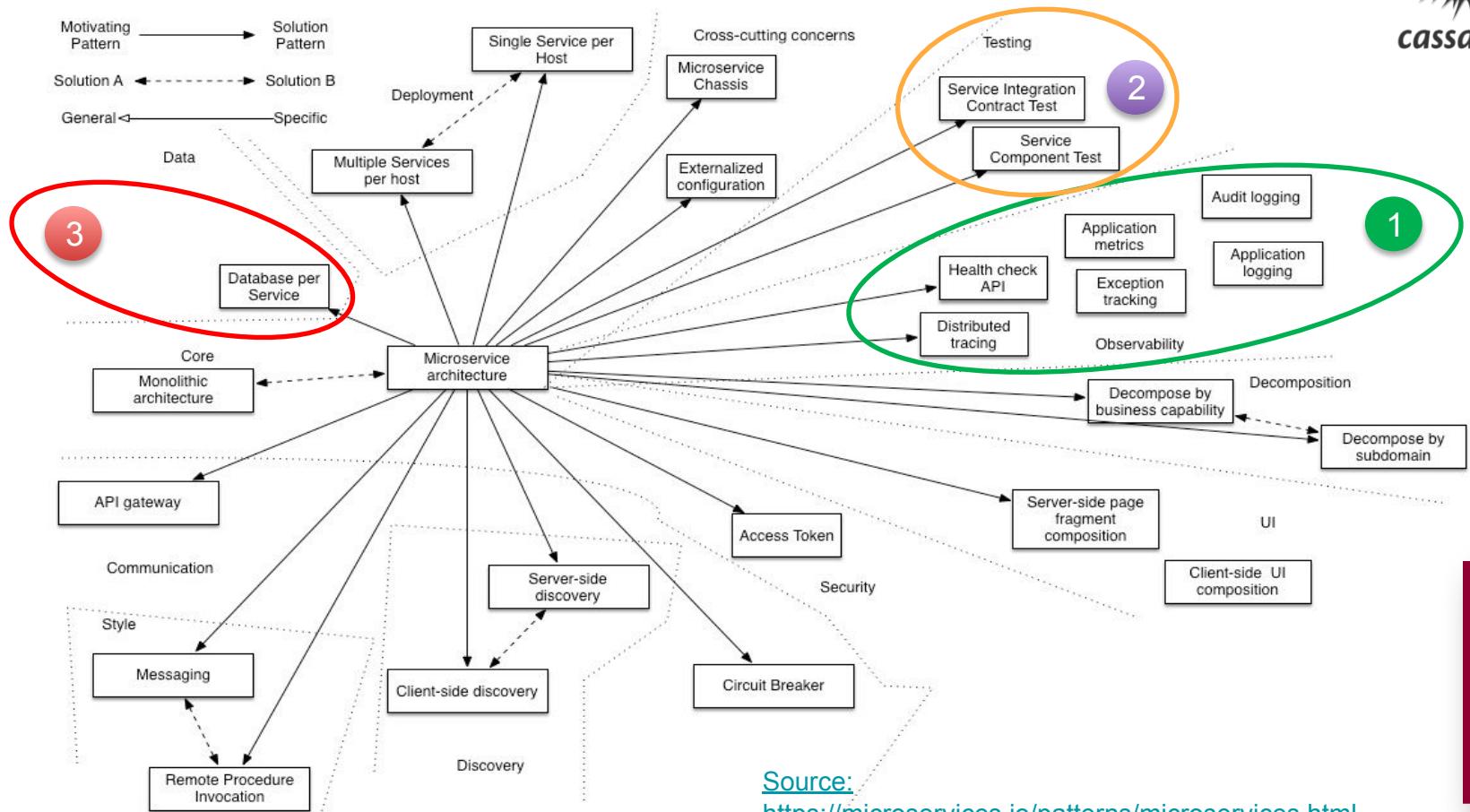


- Complexity (Security, Transaction, Orchestration)
- Cultural Changes
- Bigger RUN footprint



MicroServices

# Architectures Micro-Services : Patterns



**Microservices should not share anything**  
**You install a Database for each service**

**Do you ?**  
**Does anyone do that ?**

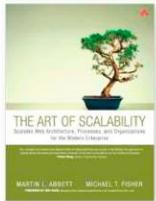
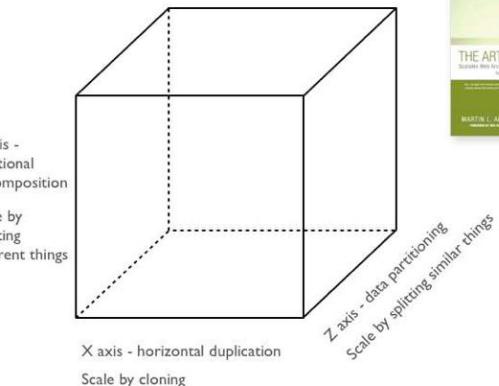
**No. So WHY ?**



# Database par service vs Database « partagée »

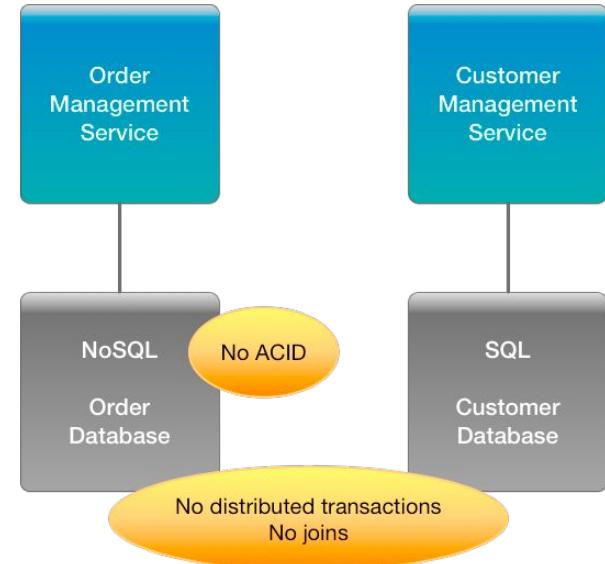
- Le couplage entre les services doit rester **lâche**
  - Chaque service doit être responsable de ses données
- Une base de donnée partagée introduirait une forme de couplage ?
- **Oui ! Mais...** : Base de données partagée  $\neq$  données partagées
    - Apache Cassandra™ permet un cloisonnement
      - Par keyspace (option de réplications)
      - Par table (1 table = 1 requête)
      - Par utilisateur (RBAC)
    - Apache Cassandra™ permet des modifications à chaud
      - L'arrêt d'un nœud n'entraîne pas d'arrêt du service

## 3 dimensions to scaling



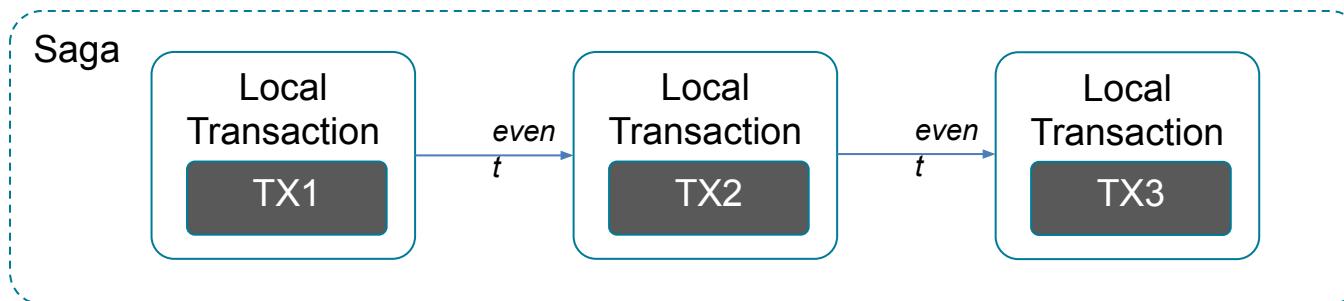
# Eventual Consistency ou passer de « ACID » vers « BASE »

- Pour des opérations cross-services:
  - Atomicity Consistency Isolation Durability (**ACID**) ne fonctionne plus.
  - Distributed transactions and 2 phases commit (**2PC**) ne fonctionnent plus.
- BASE (**B**asic **A**vailability, **S**oft-State, **E**ventual Consistency)
  - Privilégier la disponibilité sur la consistance
  - Event Sourcing: Sauvegarde des *messages* et non de l'état final
  - Idempotence: Il doit être possible de rejouer plusieurs fois chaque message
  - Api Composition: proxy permet de définir une vue unique



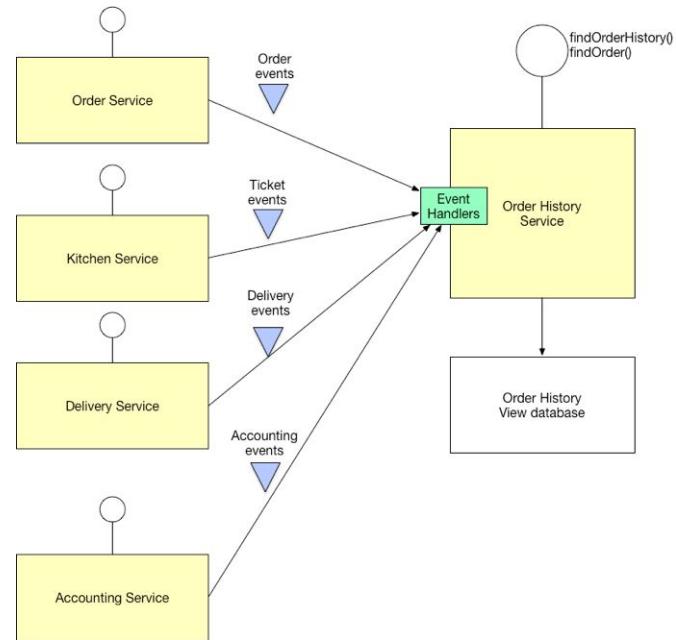
# Les « Sagas », pour assurer consistance de la donnée

- Définitions de transactions locales
- Envoi d'évènements de proche en proche
- Evènements de compensation en cas de problèmes
- 2 modes
  - Chorégraphie (Event Drivent Architecture)
  - Orchestration (BPM)



# Command Query Responsibility Segregation (CQRS)

- Séparer lecture et écriture de la donnée
- Les lectures sont réalisées sur des vues accessibles en *read-only* construites à la volée (Event Sourcing) sur la base des évènements.





- Architectures distribuées pensées pour la scalabilité et la résilience le requêtage en temps réel.
- Des modèles de données adaptés aux besoins : timeseries, requêtes CRUD sur les entités
- Un couplage lâche et une ségrégation by design , y compris sur le même cluster
  - *keyspace* = domaine
  - *table* = requêtes
- Une même philosophie pour l'implémentation des transactions et requêtes cross-services
  - Basic Availability, Soft-State Eventual Consistency (BASE)
  - Command Query Responsibility Segregation (CQRS)
  - Sagas

- **1 db per Microservices is a “lie” and does not mean a DB INSTALLATION.**
- 
- **REALTIME REQUESTS & SCALABILITY AT CORE**
- 
- **DISTRIBUTED ARCHITECTURES**
- **From ACID to BASE (Basic Availability, Soft-State, Eventual Consistency)**
- **Implementations: CQRS, Event Sourcing**
- **Colocate service and Data**
- 
- **DECOUPLING BY DESIGN**
- **1 KEYSPACE = DOMAIN**
- **1 QUERY = 1 TABLE**
- 



Cassandra



Microservices





# Hands-on Setup your Database

**#1 Connect to your Astra instance  
(Create/UnHibernate your Astra  
instance as needed)**

# 01



**Bootcamp 2022**  
**Housekeeping**  
**Reminders**

# 02

**Microservices**  
**Why with Apache Cassandra ?**

# 03

**APIs**  
**Rest, GraphQL, gRPC**

# 04

**Todo Application**  
**TodoMVC, TodoBackend**

# 05

**Code Build Package**  
**Building efficient Data Model**

# 06

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



**Agenda**

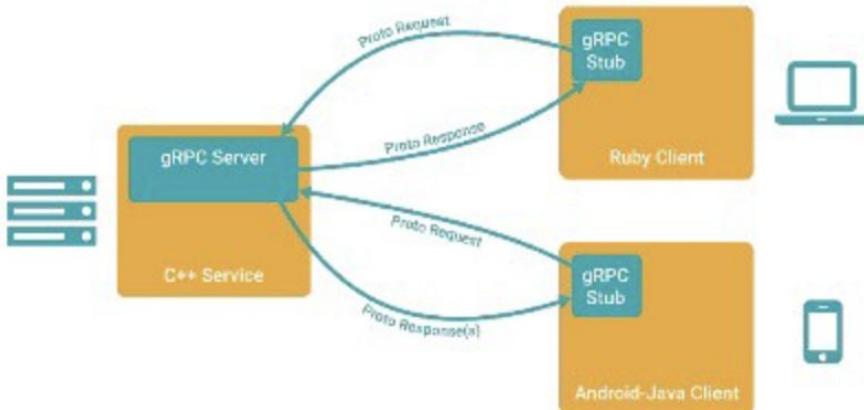
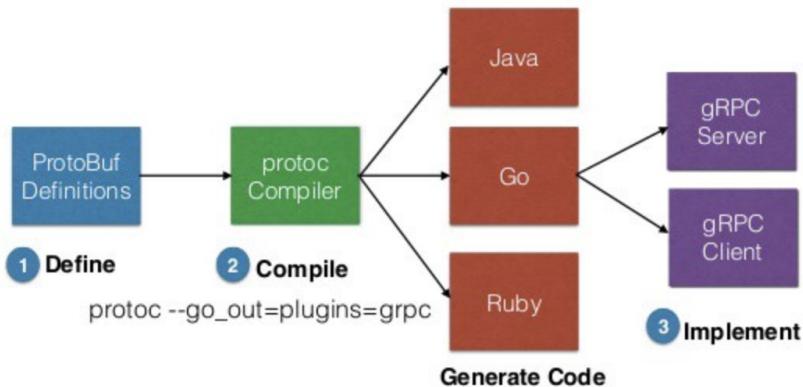


# { REST }

The screenshot shows the Swagger UI interface for a REST API. At the top, there's a navigation bar with the Swagger logo, the URL '/swagger.json', and a green 'Explore' button. Below the navigation is a section titled 'documents'. The main content area lists various API endpoints with their methods, URLs, and descriptions. The endpoints are color-coded by method: GET (blue), POST (green), PUT (orange), DELETE (red), and PATCH (light green). The descriptions provide details like 'List collections in namespace' or 'Create a new document'.

Method	URL	Description
GET	/v2/namespaces/{namespace-id}/collections	List collections in namespace
POST	/v2/namespaces/{namespace-id}/collections	Create a new empty collection in a namespace
GET	/v2/namespaces/{namespace-id}/collections/{collection-id}	Search documents in a collection
POST	/v2/namespaces/{namespace-id}/collections/{collection-id}	Create a new document
DELETE	/v2/namespaces/{namespace-id}/collections/{collection-id}	Delete a collection in a namespace
POST	/v2/namespaces/{namespace-id}/collections/{collection-id}/upgrade	Upgrade a collection in a namespace
POST	/v2/namespaces/{namespace-id}/collections/{collection-id}/batch	Write multiple documents in one request
GET	/v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id}	Get a document
PUT	/v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id}	Create or update a document with the provided document-id
DELETE	/v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id}	Delete a document
PATCH	/v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id}	Update data at the root of a document
GET	/v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id}/{document-path}	Get a path in a document
PUT	/v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id}/{document-path}	Replace data at a path in a document
DELETE	/v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id}/{document-path}	Delete a path in a document
PATCH	/v2/namespaces/{namespace-id}/collections/{collection-id}/{document-id}/{document-path}	Update data at a path in a document
GET	/v2/namespaces/{namespace-id}/collections/{collection-id}/json-schema	Get a JSON schema from a collection

# gRPC



@hostirosti @grpcio



It is a web protocol (HTTP) that specifies the way we build and query remote APIs using a Tree/JSON Syntax.

Based on a **strongly typed** language  
named GraphQL SDL (Schema  
Definition Language)

```
incomingWorkshops {  
  title  
  abstract  
  speakers  
}
```

```
type Workshop {  
  title: String!  
  abstract: String  
  speakers: [Speaker]  
  releaseYear: int  
}
```



A protocol that allows the client to specify exactly **what** data it needs from a model

```
incomingWorkshops {  
    title  
}
```

It allows one to aggregate data from **multiple relations** in a single query

```
incomingWorkshops {  
    title: String!  
    abstract: String  
    speakers: {  
        name  
    }  
    releaseYear: int  
}
```

# { REST }



- ❖ Decoupling Client / Server (*Schema on read*)
- ❖ Api Lifecycle (*Versioning*)
- ❖ Tooling (*API Management, Serverless*)



- ❖ Verbose payloads (*json, xml*)
- ❖ No discoverability
- ❖ Not suitable for command-like (functions) API



- ❖ CRUD superstar
- ❖ Relevant for mutations (OLTP)
- ❖ Public and web APIs
- ❖ Limited Business Scope



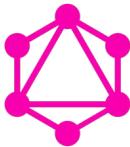
- ❖ High Performances (*http/2 – binary serialisation*)
- ❖ Multiple stubs : Sync, Async, Streaming
- ❖ Multi languages - Interoperability



- ❖ Strongly coupled (*schema with proto files*)
- ❖ No discoverability
- ❖ *Protobuf* serialization format



- ❖ Distributed network of services (no waits)
- ❖ High throughput & streaming use cases
- ❖ Command-like (eg: *slack*)



# GraphQL



- ❖ Discoverability, documentation
- ❖ Custom payloads
- ❖ Match standards (Json | Http)



- ❖ Single endpoint (*versioning, monitoring, security*)
- ❖ Complex implementation (*tooling, still young*)
- ❖ Nice for customers nasty for DB (*N+1 select*)



- ❖ Backend for frontend (JS)
- ❖ Service aggregation | composition (*joins*)
- ❖ When volume matters (*mobile phones*)

# 01



**Bootcamp 2022**  
**Housekeeping**  
**Reminders**

# 02

**Microservices**  
**Why with Apache Cassandra ?**

# 03

**APIs**  
**Rest, GraphQL, gRPC**

# 04

**Todo Application**  
**TodoMVC, TodoBackend**

# 05

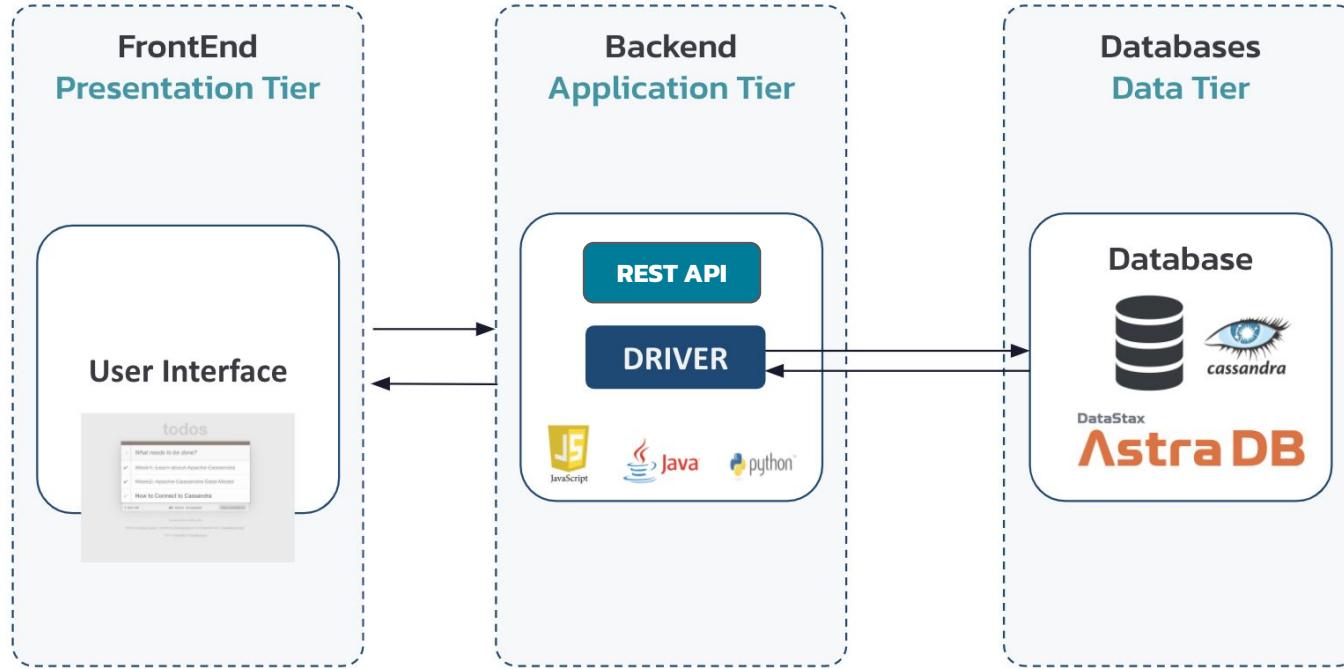
**Code Build Package**  
**Building efficient Data Model**

# 06

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



**Agenda**



Architecture of our Microservice



Helping you **select** an MV\* framework

[Download](#) [View on GitHub](#) [Blog](#)



Todo-Backend

Implementations Contribute

## Introduction

Developers these days are spoiled with choice w  
it comes to **selecting** an **MV\*** **framework** for  
structuring and organizing their JavaScript web ap  
Backbone, Ember, AngularJS... the list of new ar  
stable solutions continues to grow, but just how c  
you decide on which to use in a sea of so many  
options?

To help solve this problem, we created [TodoMVC](#)  
project which offers the same Todo application  
implemented using MV\* concepts in most of the  
popular JavaScript MV\* frameworks of today.

## Todo-Backend

*a shared example to showcase backend tech stacks*

The Todo-Backend project defines a simple web API spec - for managing a todo list. Contributors implement that spec using various tech stacks. Those implementations are cataloged below. A spec runner verifies that each contribution implements the exact same API, by running an automated test suite which defines the API.

The Todo-Backend project was inspired by the [TodoMVC project](#), and some code (specifically the todo client app) was borrowed directly from TodoMVC.

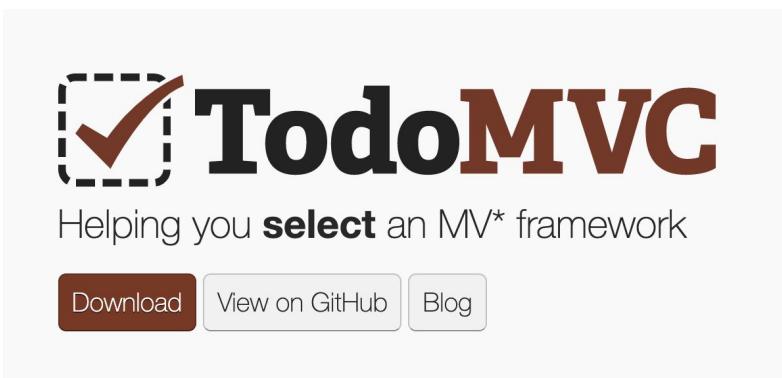
Created and curated by Pete Hodgson.

Featuring HTTP APIs built with:



# The Todo Application

<http://todomvc.com/>



The screenshot shows the TodoMVC homepage. It features a large logo with a checkmark inside a dashed box, followed by the text "TodoMVC" in a large, bold, brown font. Below the logo, the text "Helping you **select** an MV\* framework" is displayed. At the bottom, there are three buttons: "Download" (dark brown), "View on GitHub" (light gray), and "Blog" (light gray).



The screenshot shows the AngularJS example of the TodoMVC application. It features a red header with a white letter 'A'. The main interface displays a list of todo items with radio buttons next to them. The items are:

- Explain the use case
- Create the Data model
- Define the queries to perform
- Create the DDL
- Connect to Cassandra
- Create the CRUD repository
- Run the API

At the bottom, it shows "7 items left" and filters for "All", "Active", and "Completed".

<http://todomvc.com/examples/angularjs/>

# Todo-Backend

<https://www.todobackend.com/>

# Todo-Backend

*a shared example to showcase backend tech stacks*

The Todo-Backend project defines a simple web API spec - for managing a todo list. Contributors implement that spec using various tech stacks. Those implementations are cataloged below. A spec runner verifies that each contribution implements the exact same API, by running an automated test suite which defines the API.

The Todo-Backend project was inspired by the [TodoMVC](#) project, and some code (specifically the todo client app) was borrowed directly from TodoMVC.

Created and curated by [Pete Hodgson](#).

**featuring HTTP APIs built with:**



aiohttp



Akka



API Platform



Axon Framework



Azure Functions



CakePHP



Catalyst



Ceylon



Clojure



CoffeeScript



Compojure



CouchDB



Crystal



C#



django



.NET



Dropwizard



Elixir



ES6



express



Finatra

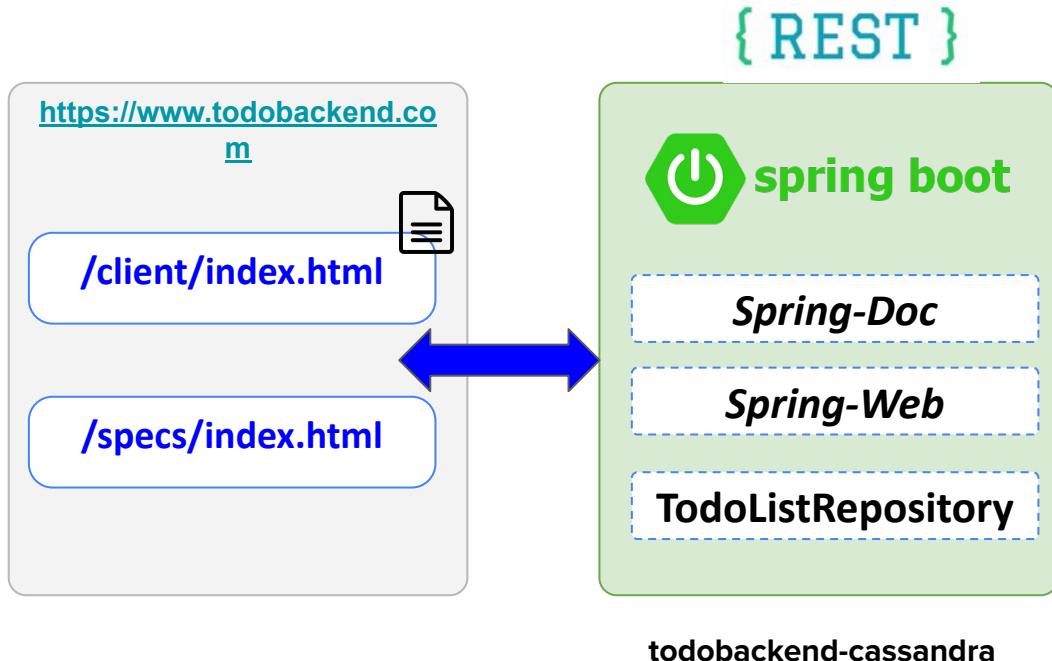


Finch

# Todo-Backend



# REST API = todobackend-cassandra



The screenshot shows the Swagger UI interface for the TodoBackend Rest API.

**Header:**

- Swagger UI
- Select a definition: Rest Controllers (Spring MVC)

**Section: DevWorkshop :: TodoBackend Rest API (1.0-SNAPSHOT, DAS)**

Implementation of TodoBackend application with Spring WebMVC and storage in Apache Cassandra

Terms of service  
Apache 2.0

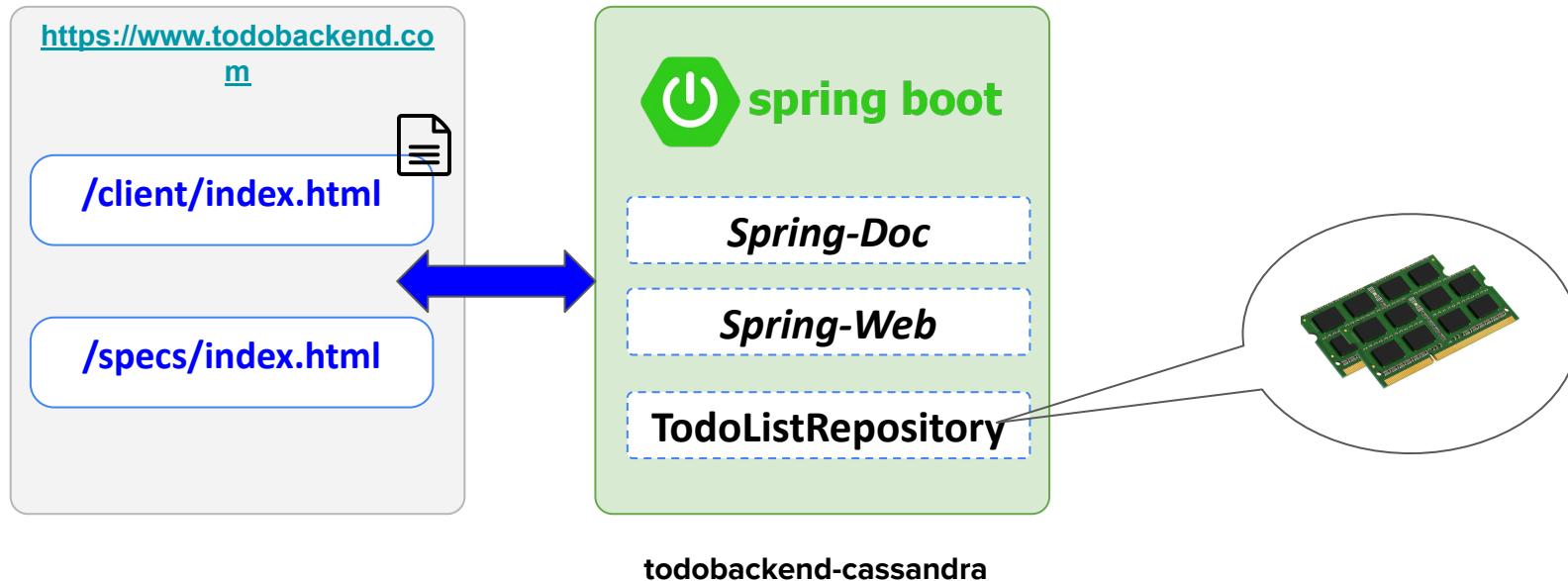
**Servers:**

http://localhost:8080 - Generated server url

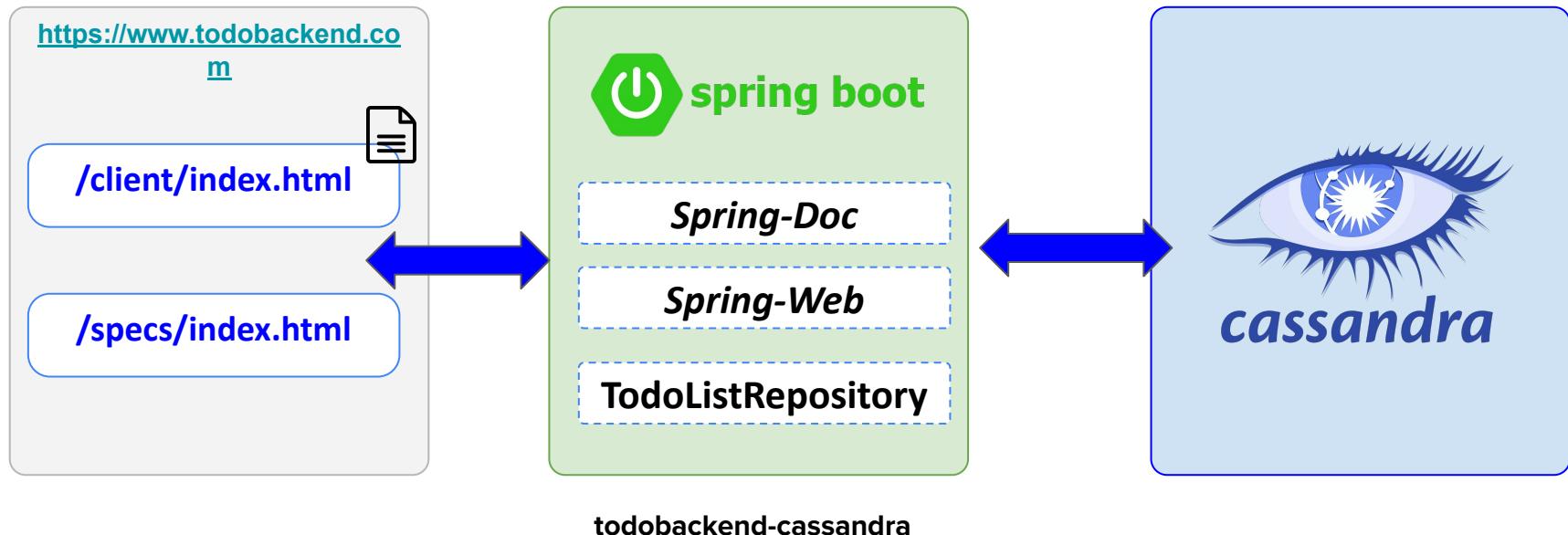
**Todos:** Implement CRUD operations for Todo Tasks

Method	Path	Description
GET	/api/v1/todos/{taskId}	Get details of a task if exists
DELETE	/api/v1/todos/{taskId}	Delete a task from its id if exists
PATCH	/api/v1/todos/{taskId}	Update an existing task
GET	/api/v1/todos	Retrieve the complete list of Tasks
POST	/api/v1/todos	Create a new task
DELETE	/api/v1/todos	Delete all tasks in one go

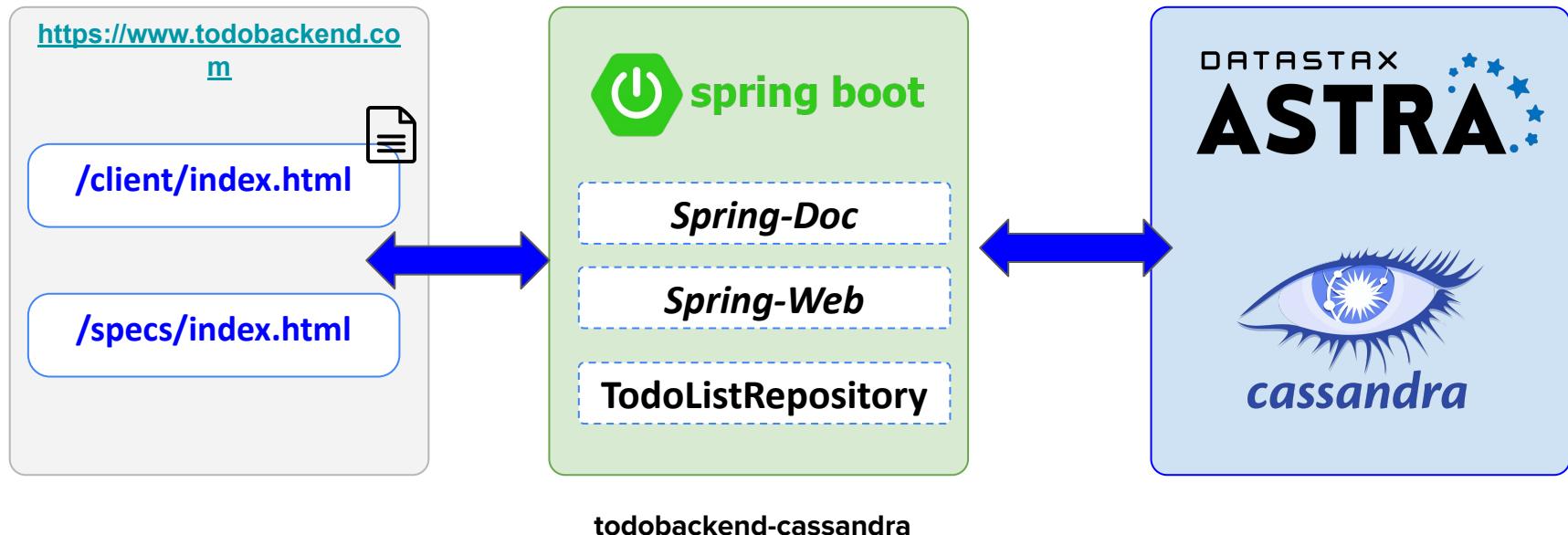
# REST API = todobackend-cassandra



# The Todo Application



# The Todo Application



## Todos

Implement CRUD operations for Todo Tasks

**GET** `/api/v1/todos/` Retrieve the complete list of Taskss

**POST** `/api/v1/todos/` Create a new task

**DELETE** `/api/v1/todos/` Delete all tasks in one go

**GET** `/api/v1/todos/{taskId}` Get details of a task if exists

**DELETE** `/api/v1/todos/{taskId}` Delete a task from its id if exists

**PATCH** `/api/v1/todos/{taskId}` Update an existing task



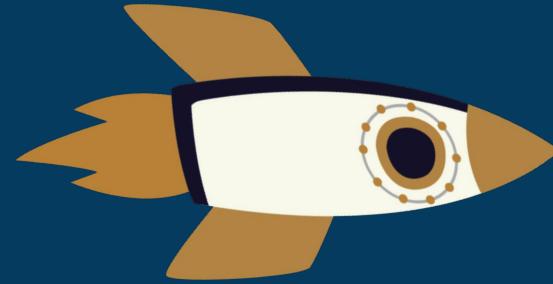
# todos

What needs to be done?

- ✓ Test the TodoApplication
- ✓ Create a REST API Backend
- ✓ Connect the backend to Cassandra
- ✓ Have fun
- ✓ Register to Youtube channel

4 items left    All Active Completed    Clear completed (1)





# Hands-on Setup your Application

- #2 Create your Token
- #3 Launch Gitpod
- #4 Know your Gitpod
- #5 Setup your Application
- #6 Connectivity



# 01



**Bootcamp 2022**  
**Housekeeping**  
**Reminders**

# 02

**Microservices**  
**Why with Apache Cassandra ?**

# 03

**APIs**  
**Rest, GraphQL, gRPC**

# 04

**Todo Application**  
**TodoMVC, TodoBackend**

# 05

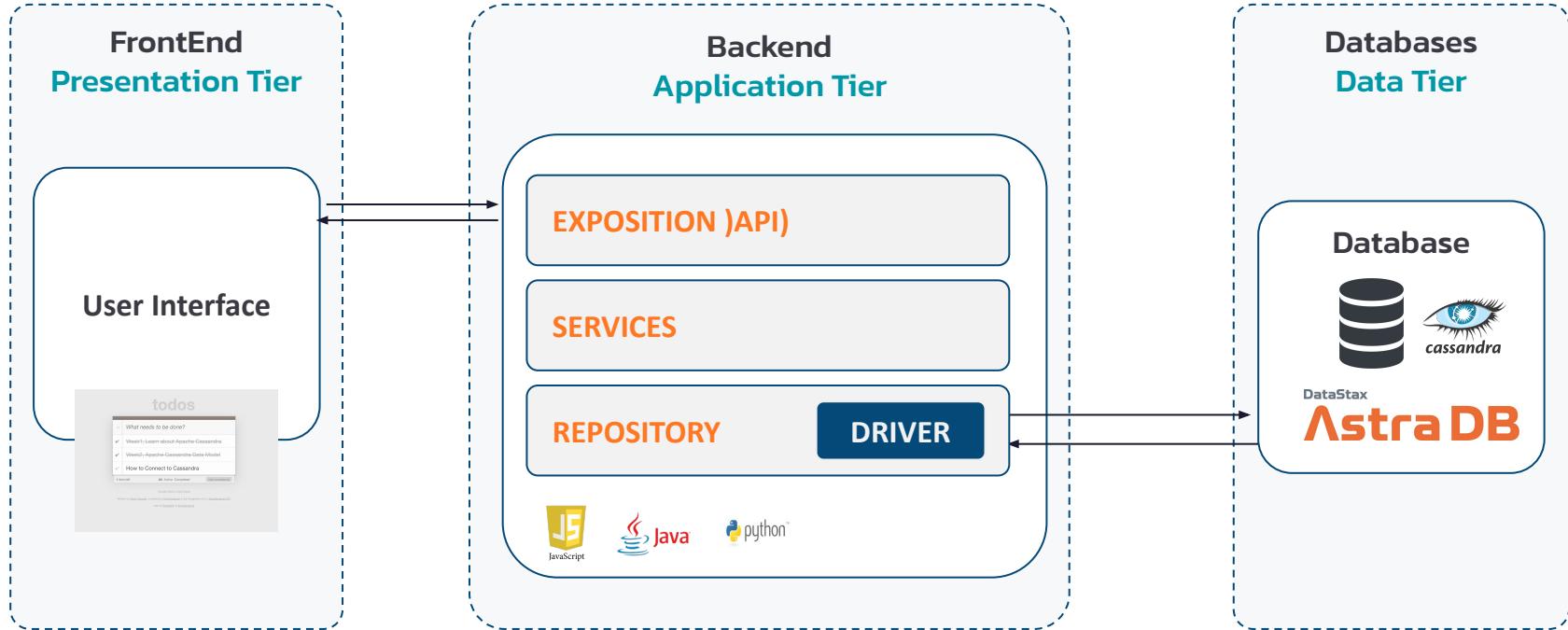
**Code Build Package**  
**Building efficient Data Model**

# 06

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



**Agenda**



DataStax Developers



# todos

- What needs to be done?
- ✓ Test the TodoApplication
- ✓ Create a REST API Backend
- ✓ Connect the backend to Cassandra
- ✓ Have fun
- ✓ Register to Youtube channel

4 items left    All Active Completed    Clear completed (1)

List all tasks

Create a new Task

Mark a task as  
completed/uncomplete

Delete a task



Service

## todoitems

user_id	TEXT	K
item_id	TIMEUUID	C↑
completed	BOOLEAN	
title	TEXT	
offset	INT	

```
CREATE TABLE todos.todoitems (
    user_id      text,
    item_id      timeuuid,
    completed    boolean,
    title        text,
    offset       int,
    PRIMARY KEY ((user_id),item_id)
);
```

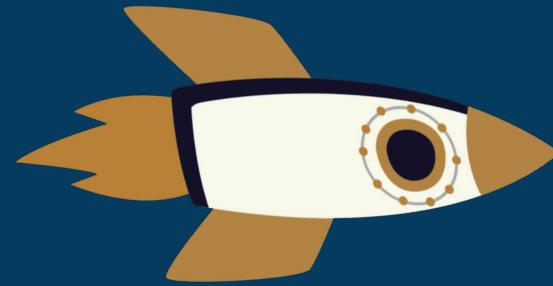


```
public interface TodoListRepository {  
  
    /** Constants for table todo_tasks to be used in statements */  
  
    String TABLE_TODO_TASKS      = "todo_tasks";  
    String TASK_COL_UID         = "uid";  
    String TASK_COL_TITLE       = "title";  
    String TASK_COL_COMPLETED   = "completed";  
    String TASK_COL_OFFSET      = "offset";  
  
    ...  
}
```

## Repository and Driver



Rest Api



# Hands-on Setup your Application

#7 Code Browsing  
#8 Run the application



# 01



**Bootcamp 2022**  
**Housekeeping**  
**Reminders**

# 02

**Microservices**  
**Why with Apache Cassandra ?**

# 03

**APIs**  
**Rest, GraphQL, gRPC**

# 04

**Todo Application**  
**TodoMVC, TodoBackend**

# 05

**Code Build Package**  
**Building efficient Data Model**

# 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 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".

# 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**



DataStax [Developers](#)



LIVE session

## Learn how to build an e-Commerce app!



LEVEL  
**UP**  
with the DataStax Developers

January 31

MON JAN 31 2022

Build an e-Commerce App with AstraDB

[Register Now](#)

LIVE

## Build a multiplayer real-time game

with WebSockets and Apache Pulsar messaging!



LEVEL  
**UP**  
with the DataStax Developers

February 02

WED FEB 02 2022

Build a multiplayer realtime game with WebSockets & Apache Pulsar messaging

[Register Now](#)



Next Week



DataStax Developers

# workshop-chat

<https://www.youtube.com/watch?v=MuwT5xxFVVI> - Subscribe to mailing list: [http...](http://)

Rechercher

PRESENTER — 1  
David Jones-Gilardi

HELPER — 7  
012345  
AaronP  
B1nary  
Chelsea Navo  
Jeremy Hanna  
John Sanda  
Patrick\_McFadin

EN LIGNE — 560  
-samu-  
6304-42JB  
Aahlya  
Abdurahim  
abhi3pathi  
Abhiis.s  
Abhineet  
Abirsh

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

ASTRA DB  
getting-started  
astra-apis  
astra-development  
sample-applications

APACHE CASSANDRA  
Cedrick Lun...

RIGGITYREKT Hier à 21:14  
I have a 5 node datacenter, 4 nodes are on dse version 5.1.20, one is on dse5.0.15. I am doing some mixed version testing for a class and the one node that is 5.0.15 is coming up as an analytics workload. I dont have /etc/default/dse, instead I am using /etc/init.d/dse-cassandra.  
how do I make that node start in cassandra workload, not in analytics?

RIGGITYREKT Hier à 23:39  
Okay I found out my issue, when i started DSE 5.0.15 it had endpointsnitch set to DseSimpleSnitch, the rest of my cluster is using PropertyFileSnitch, when i change it to PropertyFileSnitch, it still uses the simple snitch config. looking at the docs i see there is a way to go to GossipingPropertyFileSnitch, but i need the property file one. I can wipe this dbs, do anything with this node to get this done. how do i fix this?  
@here

Erick Ramirez Aujourd'hui à 02:19  
mixed versions isn't supported and you're guaranteed to run into weird issues that will cause further problems down the track

Cedrick Lunen Aujourd'hui à 09:01  
When you start a node you have parameters -k for analytics, -g for graph and -s for search. To remove analytics check and remove -k

Envoyer un message dans #workshop-chat

# !discord

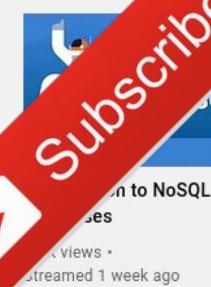
[dtsx.io/discord](https://dtsx.io/discord)



DataStax Developers Discord (18k+)



# Subscribe



# Subscribe



Astra Streaming Demo  
177 views • 2 weeks ago

Kubernetes Ingress Management with Traefik...  
496 views • Streamed 2 weeks ago

Build your own TikTok clone!  
1.9K views • Streamed 3 weeks ago

Build your own TikTok Clone!  
4K views • Streamed 3 weeks ago

How to use the Connect Driver in Astra DB  
113 views • 4 weeks ago

How to use the CQL Console in Astra DB  
39 views • 4 weeks ago



How to create an Authentication Token in...  
37 views • 4 weeks ago

How to use the Data Loader in Astra DB  
62 views • 4 weeks ago

Astra DB Sample App Gallery  
36 views • 4 weeks ago

How to use Secure Connect in Astra DB  
42 views • 4 weeks ago

Cassandra Day India: CL Room (Workshops)  
2.4K views • Streamed 4 weeks ago

Cassandra Day India: RF Room (Talks)  
1.3K views • Streamed 1 month ago

# Thank You!

